Nostalgia Ti Frega

A photographic topography of memory, identity, and place.

An exegesis submitted in fulfilment of

the requirements for the degree of

Doctorate of Creative Arts

Carla Drago

B.A. (Communications)
Master of Interactive Multimedia

Faculty of Arts and Social Sciences
University of Technology, Sydney
December 2013

CERTIFICATE OF ORIGINAL AUTHORSHIP

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Student:

Production Note:
Signature removed prior to publication.

Date: 12 November 2013

ACKNOWLEDGEMENTS

The completion of this doctoral project would not have been possible without the support, participation, contribution, and counsel of many people, most significant of whom are the members of the extended community of *Salitani* who agreed to be interviewed and photographed. *Grazie a tutti per aver partecipato*! I hope you like the photos.

I'd like to acknowledge the significant contribution of Dr Meredith Jones, my supervisor, without whom the project would never have been conceived let alone completed. Also Dr Elaine Lally and Dr Paula Hamilton, who both acted as co-supervisors at different times. Thanks to Dr Linda Leung who allowed me to use previously published material we have co-authored in this exegesis.

Thanks to my creative and technical collaborators, Dino Karlis, Stefanie Posavec, Tim Schober, and Derya Irkdas Dogu; my translators Valentina Meglioli and Daniele Vecchiato; my app checker Beatrice Drago; and my friends-with-great-advice Adriaan Stellingwerff, Mei Tsering, Justin McMurray, Conor Delahunty, Antoine Tremoulet, and Anna Burles.

To Dr Marsha Rosengarten, Cath Le Couter, Lara Esden, Richard Payten, Paul O'Byrne, Greg Logan, Rose Tun Tin, Scott Millwood, and Gulsum Akcay, thanks for being the best wing-buddies any doctoral candidate could have.

I'd like to thank Leica Camera AG for loaning me a camera during a crucial shooting period while repairs were made to my own, and the University of Technology, Sydney, for supporting this research through their doctoral scholarship program.

And finally, to my father Frank, my mother Anna, and my brothers Matthew and Michael, this is for you.

PREFACE

Sections of Part Two of this exegesis have been previously published in a different form in *Digital Experience Design: Ideas, Industries, Interactions*, edited by Linda Leung, 2008, published by Intellect Ltd, Bristol, UK.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS AND TABLES

ABSTRACT

Nostalgia Ti Frega is a photographic exploration of memory, identity, and place focusing on the migrant community of a Sicilian village destroyed by an earthquake 40 years ago. It is a creative photographic project that aims to capture a sense of the virtual space the destroyed village now inhabits within the community, the memories and stories that form this space, and the role of photographs in its construction. The project is designed to depict and engage the community of villagers and their descendants, inviting them to engage and interact with photographs and experience a sense of what 'the village' is today.

Broadly the project consists of over 1200 individual postcard-sized printed photographs, presented in (and across) 5 specially designed, handcrafted wooden boxes; a 'bespoke' iPad application showcasing the same photographs within an emergent digital environment, allowing users to discover images playfully and leave comments if desired; and an exegesis that outlines the theoretical and creative framework of the project, including literature of influence during creative development, methodologies employed during research and production, and the project's contribution to broader creative practice within a hybrid post-digital media context.

Conceptually *Nostalgia Ti Frega* examines notions of memory, identity, and place by tracing a path through the disruption of post-war migration and loss of homeland to current thinking around a 'reflective nostalgia' that makes space for hybrid identities and histories. It also examines the evocative potential of objects and touch, both physical and digital, as sources of identity, connection, and belonging.

Creatively the project has sought to experiment with storytelling, exploring how non-linear, fragmentary, hybrid story platforms may operate to enhance formation and maintenance of a collective hybrid identity. To that end it positions itself as the inheritor of a creative trajectory beginning with literary experiments in book publishing in the 1960s, moving through broadcast television during the advent of media convergence, and arriving at the mutable “post-artefact” potential of current digital technologies.

Finally, while employing an emergent creative process, the digital design and development of *Nostalgia Ti Frega* has drawn heavily from the Agile methodology of software development. The synergies and resonances with this methodology are outlined.

KEYWORDS: reflective nostalgia, memory, community, immigration, Australia, Italy, digital design, digital development, creative practice, photography, material culture, agile, storytelling, post-artefact, Sicily, comfortableness, hybridity.

INTRODUCTION

In 1968, two years to the day before my birth, the inhabitants of my father's Sicilian village, Salaparuta, trundled into a snow-covered countryside, set up camp and waited. Earlier that day an earth tremor had caused the bell of the church tower to dislodge and fall into the village square. Deeply religious, and deeply superstitious, the *paesani* took this as a portentous omen. They packed food and a few belongings, and evacuated into the cold.

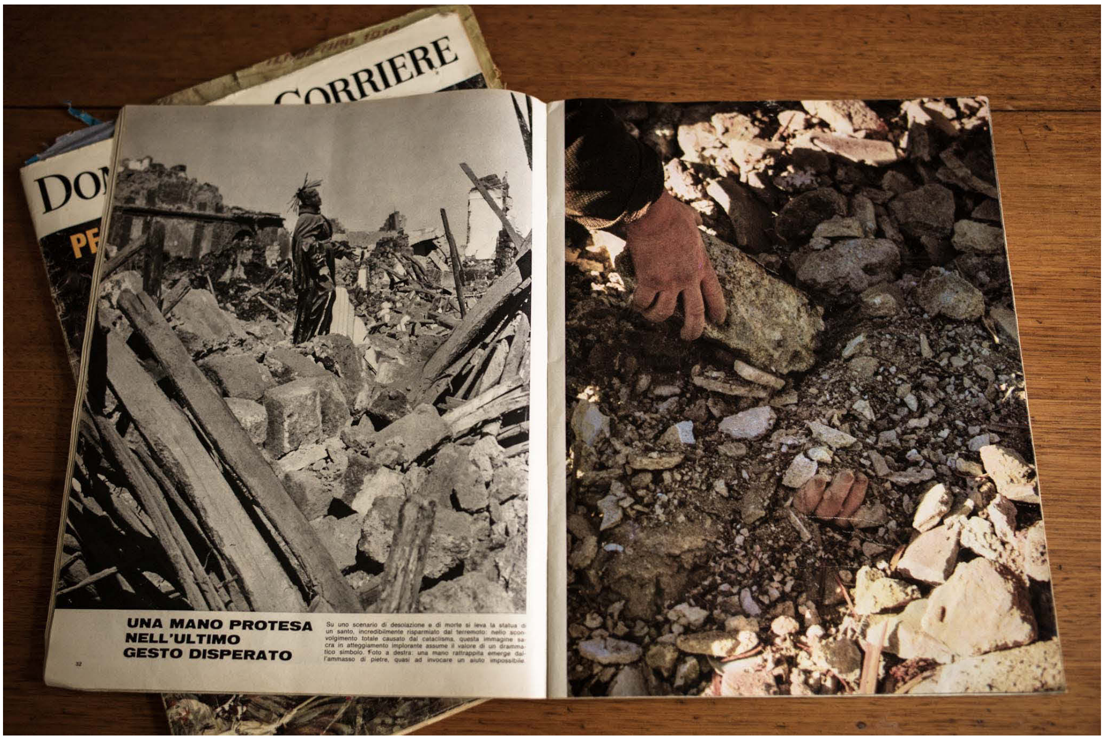
The messengers they sent to warn the neighbouring villages were met with derision, for the tremor had been slight, and had not emanated far. So, out in the countryside, as nothing happened and night fell, the *Salitani*[1] wondered whether their initial panic was mistaken. They were cold and hungry and what of all the things they'd left behind which perhaps they should return to save?

It was my grandfather, Matteo, who advised everyone to stay. A smart farmer, involved in local politics, he knew about earthquakes - that small tremors were always followed by larger ones - and it was lucky he did. For sure enough, at two in the morning, underneath a full moon bouncing luminescence off the snow, the earth shook again, only this time stronger and harder and much like a post-dinner tablecloth firmly shaken of its crumbs. For half a minute (which seemed much longer) the ground convulsed and swayed, knocking some to the ground, sending others into supportive

[1] Italian parlance used to describe people who come from the place Salaparuta. Like *Romani*, for Romans, or *Napoletani*, for Neapolitans.

clutches, and causing all to instinctively mutter some prayer or other.

When the shaking finally stopped, the villagers regained their footing (and their selves), and looked across the hills to where their village stood. Only it wasn't a village anymore. It was a huge cloud of dust, ballooning into the moonlit sky. Dumbstruck, they watched it rise. It looked terrifying and beautiful all at once.



UNA MANO PROTESA NELL'ULTIMO GESTO DISPERATO

Su uno scenario di desolazione e di morte si leva la statua di un santo, incredibilmente risparmiato dal terremoto: nello sconvolgimento totale causato dal cataclisma, questa immagine sacra in atteggiamento implorante assume il valore di un drammatico simbolo. Foto a destra: una mano rattrappita emerge dall'ammasso di pietre, quasi ad invocare un aiuto impossibile.

Figure 1: An Italian photo magazine from 1968 depicting the aftermath of the Belice earthquake.

I don't remember when, as a child, I first learnt about the earthquake. I have no memories of hushed tones, halted conversations, or awkward silences. My home was boisterous and opinionated, purposeful and somewhat melodramatic. Silences were seldom, usually angry, or when everyone was otherwise asleep.

I can't construct a moment when the earthquake's presence arrived in my own life as an Australian child. Unlike the evening news flash that interrupted The West Wing on September 11, or the phone call, during a holiday in Tasmania, that brought news of my grandmother's death, the earthquake was never a new, confronting reality. I don't remember it appearing. It was always simply there.

I do remember old magazines, frayed at the bindings, foreign, with graphic photo-spreads of desperate headscarved women, collapsed townscapes, and dusty body parts - like the hand revealed, encased beneath piles of rubble, reaching for a surface it never found. These magazines were better hidden, kept in a private drawer in my parents' room that was off limits. I only ever saw them once or twice and didn't want (or need) to see them again. The image of the clawing hand was imprinted. I sometimes thought about it at night, in bed, before I fell asleep.

So although the earthquake was always in conversation, a harbinger of bad fortune, a before-and-after demarcation, a thing that could be railed at and (somehow) contained, pictures of the event were different. Kept hidden, saved as mementos but handled with more discretion. Pictures, it seemed, were more powerful than words, harder to contradict or reshape. Pictures provoked uncomfortable emotions. Pictures always had the final word.

In broad terms, *Nostalgia Ti Frega* is a creative doctoral project about memory, identity, and place. It focuses on a specific community - one that experienced a cataclysmic event, became geographically dispersed, but which nevertheless remained somehow connected to a shared, virtual, sense of place. To that end the project seeks to capture a sense of this virtual space - the memories and stories that form it, the role of photographs in its construction, and its status, not as an abandoned historical relic but as something contemporary, emergent, and inhabited.

As a whole, the project consists of over 1200 individual postcard-sized printed photographs, presented in (and across) 5 specially designed, handcrafted wooden boxes; a ‘bespoke’ iPad application showcasing the same photographs within an emergent digital environment, allowing users to discover images playfully and leave comments if desired; and this exegesis which outlines the theoretical and creative framework of the project, including literature of influence during creative development, methodologies employed during research and production, and the project’s contribution to broader creative practice within a hybrid post-digital media context.

This exegesis comprises of three parts:

PLACE - which focuses on the conceptual and theoretical ideas around identity, memory, and nostalgia, and their relationships to the artifacts, locations, and pastimes, both physical and imaginary, that have informed the creative framework of the project;

PLATFORM - that delves into the new relational contexts our contemporary post-digital media context affords and what these mean for the future of storytelling and self/shared-articulation;

And PROCESS - where I discuss the modes of engagement deployed in tackling the creative challenges posed by the theoretical approach and desired outcome. This includes research, photography, product design, digital development, and modes of collaboration.

The notion of 'place' launches the Exegesis because, for the community of *Salitani* at the centre of the project, the geographic presence of their village, together with their physical presence within it, was, and is, deeply connected to their sense of identity. The question this poses - of what, in the wake of the village's destruction, was created in its place - is unpacked through the work of Julia Creet and Gokcen Karanfil (memory, migration, loss and the inability to return), Svetlana Boym and Frederic Jameson (nostalgia, identity and hybridity) and Greg Noble (national identity and comfortableness).

Boym's writing on different forms of nostalgia, namely, restorative nostalgia (characterised as an impossible attempt to restore a lost identity), and reflective nostalgia (a strategy of acceptance of paradoxical and hybrid identity), is particularly significant in formulating a theoretical understanding of how to approach the subject matter.

The manner in which photographs might operate within this theoretical framework as objects of containment, memory, and identity, is informed by the work of Greg Noble and Daniel Miller (containment and the emotional power of objects), Mikael Petterson (the phenomenology of photography), and Roland Barthes, Susan Sontag, Meredith Jones, and Annette Kuhn (the ability of photographs to confound time, act as repositories for memory, and enforce an historical authentication).

Upon this anchor of theoretical ideas around identity, nostalgia, comfortableness, and hybridity, the Exegesis then goes on to explore the kind of storytelling structure *Nostalgia Ti Frega* would be best served by, and the kind of platform it would be best deployed upon.

The work of Forlizzi & Battarbee and Robert McKee (drawing correlations between story and experience), Hillary Lawson, Richard S Wurman, and Deleuze & Guattari (examining non-linear data, individual story paths, and rhizomatic structures), Craig Mod (tracking the evolution and influence of interactivity on creative works), and Sherry Turkle and Anne Cranny-Francis (outlining the potency of touch as an interaction method) all informed decisions about how the overall ‘experience’ of *Nostalgia Ti Frega* should unfold.

The final section on process gives an overview of the methodologies employed to create and develop the art work. My background in traditional and digital media production has informed the way this has been realised as a creative project. The research interviews and production of photographs was, primarily, structurally loose, aping a *verite* documentary production style. The development of the digital component, on the other hand, followed an ‘agile’ and ‘lean’ methodology. Significant writing on the subject included Roman Pilcher’s work on Agile product management and Eric Reis’s thinking around lean software development.

On a fundamental level the inspiration behind *Nostalgia Ti Frega* was to work within a family-chronicle tradition - to photograph a broad set of images that would record time, place, and people. This was because I wanted the photographs, above all, to have meaning for the people I was photographing, i.e. the community itself. I was not interested in creating a project that lived in a gallery, or on a library bookshelf. I wanted to create something that would live in peoples’ homes and be something they used, interacted with, and contributed to. All of the consequent creative, technical, and theoretical decisions about how to create *Nostalgia Ti Frega* were made in the light of this basic intention. During research, photography production, and software development I would always return to my knowledge and understanding of this audience and ask myself what kind of experience the consequences of following a particular creative, technical, or theoretical path would afford them.

In presenting a photographic document of the community, and providing a platform upon which their stories, reflections, and memories can be shared, I like to think of *Nostalgia Ti Frega* as an elaborate gift to all the people that participated. In the words of Antoine de Saint Exupéry, “[One’s] task is not to forsee the future, but to enable it."[2] I am therefore hopeful that *Nostalgia Ti Frega* will be an artwork the community of *Salitani* can use to contribute to, create, and build an ongoing shared idea of themselves and each other.

[2] de Saint Exupéry, A. 1948, *Citadelle*, Gallimard, Paris

PART ONE - PLACE

> *"To feel at home is to know that things are in their place and so are you; it is a state of mind that doesn't depend on an actual location. The object of longing, then, is not really a place called home, but this sense of intimacy with the world; it is not the past in general, but that imaginary moment when we had time and didn't know the temptation of nostalgia."* [3]
>
> Svetlana Boym

The aftermath of the 1968 earthquake left several villages and towns in the Belice Valley, Sicily, destroyed and damaged, 380 people dead, 1000 injured, and 70,000 homeless. With few local accommodation options available, over half the 3000-strong population of the village of Salaparuta left, migrating to Australia, Canada, the US, and cities in the north of Italy. The rest camped in makeshift barracks and waited for new housing to be built nearby. Due to bureaucratic mismanagement and mafia corruption, this took more than 15 years.[4]

As earthquakes go, the one that struck that winter was not particularly strong, and despite the tragic human casualties and destruction of property, did not come close to the catastrophic earthquakes that the island's urban centres had experienced through their histories. However, it is important to

[3] Boym, S. 2001, *The Future of Nostalgia*, Basic Books, New York p251

[4] Ricucci, A. 2008, *La Storia Siamo Noi: Belice '68*, RAI Educational Television, Rome & Baldassare, G. Fr. 1992, *Salaparuta Ieri e Oggi*, Comitato Communale per l'Emigrazione di Salaparuta, Salaparuta.

note that unlike cities, which by their very nature are always in transformation, and where destruction and redevelopment are customary elements and expectations of urban culture [5], villages – usually understood as small hubs of human life in largely unpopulated landscapes – are very different entities. The entire physical presence of a village – its squares and meeting points, churches and cafes, thoroughfares and back-alleys – is intrinsic to how its inhabitants socialise and interact, shelter from the environment and from strangers, and generally differentiate between the familiar and the unknown. In short, the physical space of the village is vital to an understanding of who, as a community, its inhabitants are and what their place, as a collective, in the world is. The destruction of Salaparuta therefore had significant consequences for the villagers in terms of identity and belonging. One interview subject, approaching 70 years old, lamented that the new village was never, and would never be his village. For him the destruction of the old village, and the cultural change this brought about, felt like a part of himself, and his own history, had been extinguished.[6]

While it can be argued that all spaces of habitation shape the identity of their inhabitants, a village and its association with community, intimacy, and belonging is understood to shape identity in a particular way. The notion of ‘village’ has in fact been used as a way of understanding and negotiating ideas around community and connection in many different contexts, from the

[5] See,for example, Stewart Brand’s writing on how the functionality required from urban buildings continually changes over time. Brand, S. 1995, *How Buildings Learn: What Happens After They're Built*, Penguin, London.

[6] Interview subject #1 October 2011. Please note interview subjects are numbered rather than named in order to maintain anonymity.

destruction and redevelopment of urban landscapes[7], to the ability of mass communication to create a so-called "global village."[8]

Traditional Sicilian culture is no different in this regard. The physical endurance of a village is a cultural reference point to the community that calls it home. It maintains continuity, is a reassuring anchor, and persists. The anthropologist Loretta Baldassar, who has written extensively in the area of transnational families and migration with a particular focus on the Italian experience, examines this phenomenon in her book *Visits Home*. Focusing on a group of Italian migrants from San Fior, a village in the Treviso region of northern Italy, Baldassar explores the way ties to the ancestral village are a persistent thread in the fabric of transnational identity.[9]

So when a village is destroyed and laid desolate, and, as in the case of Salaparuta, its people are subsequently scattered across the globe, we might ask what, if anything, takes its place?

All in the Family

Even though not everyone from Salaparuta is directly related by blood, to be a *Salitano* means to be part of an extended family of *paesani* (villagers) and to be treated as such. This extends to descendants so that a child or grandchild (or great-grand-niece-twice-removed) is considered a member of

[7] Jacobs, J. 1961, *The Death and Life of Great American Cities*, New York: Random House.

[8] McLuhan, M. & Powers, B. 1992, *The Global Village: Transformations in World Life and Media in the 21st Century*, Oxford University Press.

[9] Baldassar, L. 2001, *Visits Home: Migration Experiences Between Italy and Australia*, Melbourne: Melbourne University Press.

the village, even if he or she has never been there. As children, my brothers and cousins and I, all born in Australia, were considered *Salitani* by other *Salitani* because our father was a *Salitano*. Even now as adults, when we talk about the village, it's clear that we understand it as part of who we are. For one brother it's important that he visits the village with his own children soon[10]. Another cousin, despite having other relatives in a larger neighbouring Sicilian town, always makes sure to visit the village every time she travels to Sicily[11]. Clearly, many years after its destruction, a sense of Salaparuta persists, otherwise new descendants would not be identified, nor identify as, *paesani*. Obviously the village is no longer the same physical place it once was (at least not an inhabitable one) but it is certainly something, somewhere, in the physically lived experiences and virtual imaginings of its people.

Geographically distant, physically destroyed, but kept alive and present nevertheless. How did Salaparuta become an important part of how members of this extended, dissipated, global and inter-generational community understand themselves?

In *Memory and Migration: Multidisciplinary Approaches to Memory Studies*[12], Julia Creet, feminist theorist and historiographer, outlines how western thought has traditionally drawn a strong connection between memory and place. Charting a course from Frances Yates' revival of the physical "memory

[10] Interview subject #3, January 2012

[11] Interview subject #4, December 2012

[12] Creet, J. Kitzmann, A. Eds. 2011, *Memory and Migration: Multidisciplinary Approaches to Memory Studies,* University of Toronto.

places" of antiquity, through Pierre Nora's dissecting of memory into "true" and "artificial"[13], to a contemporary context, where ubiquitous migration largely precludes geographic stability and where place, in the thinking of Paul Ricoeur[14], has come to mean something relational, rather than something fixed, Creet examines the ways in which memory, and its connection to place, operates within the migratory experience. Using examples such as artist Yvonne Singer's installation work that creates a physical work which "installs memories she never had", Creet cites a "triple-role" for memory:

> ...it is identity-forming by maintaining the original identifications; it is therapeutic because it helps bear the hardships of transplantation into a foreign culture; and it is also community-forming, by creating a bond among those recollecting together.

And she continues:

> Fourth, the experiences of all the other elements of emigration are intensified by the final shared characteristic: an inability to return.[15]

In the case of people from Salaparuta, all that effectively remained of the village they once knew *were* their memories. These memories, at first symbolic vestiges of the past, were clung to in the absence of a homeland as

[13] Where 'true' memory is contingent on geographic concretization, and 'artificial' memory a consequence of unstable geographic and generational environments.

[14] Ricoeur, P. 2006, *Memory, History, Forgetting*, University of Chicago Press, Chicago

[15] Creet, J. Kitzmann, A. Eds. 2011, *Memory and Migration: Multidisciplinary Approaches to Memory Studies,* University of Toronto. p34

a locus of identity, then became a virtual imagining, able to transform and adapt as the inability to return was accepted and new homelands and senses of national identity were forged. In other words, place, initially a geographically fixed location, shifted and transformed into something eventually more relational. This is a key concept in my project, one that I will explore further in the next section of the exegesis. Place became something that could be carried and transported collectively, something that, as Maurice Halbwachs articulates, "becomes significant, becomes a place at all, because it helps [the community] to remember what happened there."[16]

This progression, however, was not a straightforward one.

Nostalgia and Tradition

In his 1985 essay on post-modernism and consumer society, Fredric Jameson argued that the formal features of the post-modern world expressed the deeper logic of its social system, namely that it had:

> begun to lose its capacity to retain its own past, [had] begun to live in a perpetual present and in a perpetual change that obliterated traditions of the kind which all earlier social formations have had in one way or another to preserve. [17]

[16] Ibid. p8.

[17] Jameson, F. 1983, "Postmodernism and Consumer Society" in *The Anti-Aesthetic: Essays on Post-Modern Culture*, Foster, H. Ed. p118-125

Jameson was writing about architecture, which is not specifically relevant to my project, but is nevertheless meaningful in the context of the identity crises I witnessed in the community around me that arose from migration. My family, my relatives, the migrant community I was reluctantly part of, struggled desperately to cling to their traditions. It seemed as though their traditions were all they knew, that they were “stuck in the past”, trapped in a nostalgia that precluded their identities from evolving or changing. They were flung into a modern world where their pasts and histories were both physically absent and certainly not part of the cultural context they had immigrated to. They had, as Jameson articulates, lost an ability to retain a sense of history and became trapped in a perpetual nostalgic effort to retain this receding past.[18]

As a consequence, as an eccentric miserable kid growing up in the comfortable, western, educated, more-or-less-secular, conservative, clean-running-water suburbs of modern Australia - a child of a modern, electric, well-fed environment - I often butted against the traditions of my migrant relatives, traditions that clearly meant a lot to them, but which made no sense to me as a child or teen. For example:

- Specially decorated pastries, made annually, not for eating but for displaying on a purpose-built altar, giving thanks to St Joseph (Salaparuta’s

[18] To clarify here, this is my interpretation of my own experience of my own community. I am aware there is scholarly debate around migration and identity formation that questions the notion of being 'trapped' in the past, however, my own *personal* experience while growing up was of a community that clung to a way of life that did in fact feel stuck and felt like a prison to me.

patron saint). *Superstitious, geographically irrelevant, religious mumbo-jumbo, and a waste of good food.*

- My 16-year-old female cousin, whose weekend toil at Coles netted her a small sum, but who was not allowed access to her own bank account. *Sexist, controlling, unenlightened, slavery.*

- The mandatory requirement, at family gatherings, of greeting rooms-full of (even vague) relations with a kiss on the cheek, sometimes two. *Impractical, embarrassing, time consuming, maybe not even hygienic. Couldn't I just wave my hands and say hello.*

- Gauchely flamboyant wedding receptions. Dry-iced entrance processions, real doves flying out of fake cakes, Marie Antoinette champagne glass pyramids, scantily-clad-leggy-las-vegas-style dancing girls ruffling feathers seductively to B-grade music, the tasteless jokes of the creepy, middle-aged, toupeed Master of Ceremonies. *Always the same venue; always the same running order; always the same menu. And everywhere the same women in newly-purchased dresses they would only ever wear once.*

What was it all supposed to mean? And why was I forced to endure it?

Jameson's view, set against and reinforced by the rise of multinational capitalism,[19] was that people were simultaneously living in an age that was

[19] This was a world where advertising, television, and media had gained enormous cultural

radically disconnected from its past, while also, paradoxically, because of a pervasive nostalgia toward this lost past, they were unable to articulate or describe the present or the very nature of their own environment.

> ...we do not yet possess the perceptual equipment to match this new hyperspace, as I will call it, in part because our perceptual habits were formed in that older kind of space [of high modernism]. The newer architecture therefore... stands as something like a new imperative to grow new organs, to expand our sensorium and our body to some new, yet unimaginable, perhaps ultimately impossible dimensions.[20]

Jameson referred to the space he was describing as a *hyperspace* - a fragmented world whose fabric had shifted dramatically but whose workings we were not yet able to perceive. He was essentially describing an identity crisis - “...the collapse of an established order and the resulting void, from which anything was yet to emerge."[21] For the *Salitani*, their village litterally collapsed, was destroyed, their lives and identities in crisis.

Jameson's ideas around identity crisis more-or-less describe the trajectory experienced by the migrant villagers of Salaparuta. The loss of old identity

influence, where old tensions between city/country and centre/province were replaced by universal standardisations, and where the rise of car culture and the proliferation of superhighway networks enabled new kinds of travel.

[20] Jameson, F. 1997, “Postmodernism, or the Cultural Logic of Late Capitalism” in *Rethinking Architecture: a reader in cultural theory*, by Leach, N. Ed. Routledge, New York p242

[21] *Ibid*

precipitated through a physical disconnection to homeland; a disconnection that created a pervasive nostalgia towards the past; and a nostalgia which in turn prevented the emergence of new identities, are all hallmarks of what many migrants experience when adopting a new country as their home.

Svetlana Boym, artist, migrant and academic, characterises this type of nostalgia as "restorative" - a form of nostalgia which seeks the return to an original state and the recovery of an absolute truth.[22] (I'll discuss some of her ideas about alternative kinds of nostalgia later in this section). Essentially, with the weddings, the greetings, the decorative food, my relatives were simply remembering who they were. The re-enactment of their cultural practices and rituals was in part their way of being, but also a commemoration of who they were, an assertion of tradition and of their history: what they understood themselves to have always been, even within the context of a very different culture.

Cultural theorist Ien Ang takes a more political view of this bind, saying that the "reaching back into one's ancestral roots" and turning them into symbolic capital is actually an effective political strategy for those "marginalised or excluded from the structures of white or Western hegemony."[23] But she also wonders about the cost of this practice, for while the symbolic capital it produces (which recently has come to be understood in terms relating to the idea of "diaspora") may offer support, emancipation, and solidarity, it also can

[22] Boym, S. 2001, *The Future of Nostalgia*, Basic Books, New York p49

[23] Ang, I. 2003, 'Together-in-difference: Beyond Diaspora, into Hybridity', *Asian Studies Review*, 27(2): 141.

operate in a converse manner.

> While the transnationalism of diasporas is often taken as an implicit point of critique of the territorial boundedness and internally homogenising perspective of the nation-state, the limits of diaspora lie precisely in its own assumed boundedness, its inevitable tendency to stress its internal coherence and unity, logically set apart from "others."[24]

The decorative pastries, control of daughters, kissing of newly-introduced strangers, and kitsch wedding ceremonies of the Australian-based *Salitani*, while affirming and reassuring for most, were a completely alienating experience for me. Whatever I felt myself to be didn't have a place in that world. One interview subject, also born in Australia and a similar age to me, commented that she was eager to leave home because she felt the migrant community her family was part of to be constrictive and controlling, that she was unable to properly express who she was, or share her interests with others within that community.[25]

Giuseppe Tornatore, the Italian filmmaker, echoes the potential pitfalls of clinging too heavily to the tradition and the past. In his 1988 film, *Nuovo Cinema Paradiso,* Salvatore, a young man caught between the comfort and familiarity of his home village, which offers little future, and a life in the

[24] Ibid p142.
[25] Interview subject #5, November 2012

unknown urban north, which will be foreign but where he will have more opportunity, is counselled by a village elder, Alfredo, to leave the village forever. To not look back, to not even write. "Nostalgia ti fotte," Alfredo warns: "Nostalgia will fuck you".

For Alfredo there was no middle ground, no space where Salvatore could exist as part of both worlds. It was either/or, one or the other, a similar predicament to the one Ang identifies when she highlights the way diasporic formations create divisions, rather than new identities.

While the extended community of family and villagers around me clung to their traditions, effectively re-enacting a version of their village life in suburban Sydney, the formation of my identity was taking its cue from a different set of traditions.

Predicated mainly by television, often sport, and usually the combination of both, I eschewed the nostalgia of my relatives in favour of different cultural narratives. The advent of World Series Cricket[26], an affront, at the time, to cricket 'traditionalists' and the cricketing 'establishment', was for me, instead, a rallying call. Exciting, heroic, and boisterously glamorous, it may have been disruptive to the established order, but implicit in its fanfare were still notions of nationalism, identity, and belonging. Kerry Packer's coup only worked because it was able to align itself with the symbols and practices of

[26] *World Series Cricket,* viewed 2 March, 2013 <http://en.wikipedia.org/wiki/World_Series_Cricket>.

nationhood, while branding itself as modern, exciting and new.[27] It wasn't new so much as a repackaging and rebranding of the old.

Greg Noble, in an article about national identity, cites sporting events as specific sites of national narratives:

> To be an 'imagined' community, or for us to see ourselves in the narratives of national continuity, the nation must also be experienced and 'felt', and analysis of nationalist imaginings needs to be augmented by consideration of the affective ground of national identification. It is common to feel most attached to the nation when the experience of national difference is most acute—such as when we travel overseas, watch a sporting match between national teams, and participate in rituals of national commemoration.[28]

Although I was an 'Italian' kid, with olive skin, my love of cricket - a game that symbolically marked white cultural imperialism[29] - may have been as much about resistance to cultural stereotyping as it was about wanting a place to belong[30], but it didn't feel that way at the time. As a child I wasn't properly aware of any cultural hegemony in operation. I certainly didn't realise how

[27] Contrast its success with the failure of Murdoch's attempt to establish a rebel Super League, where beloved, historically significant, but commercially unviable clubs were dismantled, leaving fans with nothing familiar to connect or identify with.

[28] Nobel, G. 2002,*Continuum*, Vol.16 No.1, "Comfortable and Relaxed: furnishing the home and nation", Carfax Publishing, p53

[29] Malcolm, D. 2012, *Globalizing Cricket: Englishness, Empire and Identity*, Bloomsbury, London

[30] The cultural critic Ben Carrington outlines the way in which sport has been used as a form of cultural resistance to white rascism. See for eg. "Sport, Masculinity and Black Cultural Resistance", *Journal of Sport & Social Issues*, (1998) vol. 22 no. 3 pp 275-298

much the imagined idea of modern Australia was built on ideas and policy around the inclusion and exclusion of colour.

It turned out that colour - specifically skin colour - had always figured strongly in issues around immigration in Australia. At the time of federation immigration was a particularly important issue. Federation saw the birth of the White Australia Policy, a term that comprised "various historical policies that intentionally favoured immigration to Australia by white people from Britain and strongly discouraged non-whites."[31] This policy, it could be argued, was actually an example of nostalgia in action - the assertion of Australia as culturally and historically British.

At the time (and arguably even now) Italians were not considered white. In her article 'Olive or White? The Colour of Italians in Australia', Helen Andreoni, a historian, traces the history of representations of Italians in Australia in popular culture, and outlines the way in which Italians were treated as 'non-whites'.[32] I recall being asked by peers as a young adult whether I was "half-caste", but not really knowing what the term meant. It wasn't until I read a Malcolm X speech that I understood the way in which the colour of my skin marked my difference:

> Hannibal... had with him ninety thousand African troops, defeated

[31] *White Australia Policy,* viewed 10 May, 2013 <http://en.wikipedia.org/wiki/White_Australia_policy>

[32] Andreoni, H. 2003, *Journal of Australian Studies, Vol. 27, Issue 77* 'Olive or White? The colour of Italians in Australia'

> Rome, and occupied Italy for between fifteen and twenty years. This is why you find many Italians dark—some of that Hannibal blood. No Italian will ever jump up in my face and start putting bad mouth on me, because I know his history. I tell him when you talk about me, you're talking about your pappy, [*Laughter*] your father. He knows his history, he knows how he got that color... Now what do you think ninety thousand Africans are going to do in Italy for twenty years? [*Laughter*] It's good to know this because when you know it, you don't have to get a club to fight the man—put truth on him.[33]

My colour marked my exclusion. Despite wanting to participate in the national imaginings prevalent in popular culture at the time, olive-skinned faces weren't present. My fierce assertion of my all-sporting Australian-ness as a child, as opposed to what felt like a very foreign Italian-ness, was in response to an absence of any other viable, hybrid identity to attach to, at least not one I was able to perceive in the fragmented (and fragmenting) cultural context around me. In this world, where anglo culture was privileged, and non-Anglo culture excluded, it seemed natural I would identify with the imaginings of the majority. Jameson saw no immediate way out of this bind, at least not immediately. Dubious of our ability to experience what he saw as impossible dimensions, he waited to see what the "new architecture" would bring.[34] I waited to turn 18, when I would be legally able to step into my own life.

[33] Malcolm X, "Afro-American History", speech given in 1964, transcript published in the *International Socialist Review*, Vol.28 No.2, March-April 1967, pp.3-48.Vol.28 No.2, p5

[34] I am aware that there have been many responses to Jameson in the last 20 years, however, I am choosing to focus on the source material in this instance.

But by the end of the 1980s Italian cultural stereotypes in Australian had moved through various incarnations, from the ‘Olive Peril’ of the 1920s (popularised by cartoonists of the Labor and Trade Union press and popular newspapers like *The Bulletin* and the Brisbane newspaper *Truth*[35] and also reinforced by governmental institutions[36]) to ‘Olive Trash’ humour from the 50s to the 70s (evidenced by the pen-written books *They’re a Weird Mob* by ‘Nino Culotta’[37] or the *Italian Joke Book* by ‘Tommy Boccafucci’[38]), to ‘Wog’ humour in the 1980s.

The ‘Wog’ humour of the 1980s (the decade of my own adolescence) was differentiated from previous humour, as it first developed within ethnic communities, yet appealed to an audience outside of these communities, and signalled a shift from external ridicule to internal self-parody.[39] It was also was a sign that Australian “whiteness” was beginning to be challenged. The stereotypes these parodies represented were intrinsically tied to the migrant experience of Australia, and were an assertive representation of what it meant to be both a migrant, *and* Australian. Their familiarity and banality demonstrated something akin to what Noble describes as “ontological security - a familiarity with and confidence in the world as we

[35] Church, J. 2005, *Migration per l’Australia: The Story of Australian Migration,* Italian Historical Society & Miegunyah Press

[36] Dewhirst, C. 2014, "The 'Southern Question' in Australia: The 1925 Royal Commission's Radicalisation of Southern Italians", *Queensland History Journal, vol.22, no.4, pp316-332*

[37] O’Grady, J. 1957, *They’re a Weird Mob: A Nove*l, Ure Smith, Sydney

[38] Boccafucci, T. 1975, *The Italian Joke Book,* Belmont Tower Books

[39] Andreoni, H. 2003, *Journal of Australian Studies, Vol. 27, Issue 77* ‘Olive or White? The colour of Italians in Australia’, p 81-92

know it."[40] Noble illustrates how nationhood and national identity is borne out not only through large-scale cultural rituals and spectacles, but also through the banal and everyday. Citing remarks made by John Howard, Australian Prime Minister from 1996 to 2007, about a vision for Australia as a "comfortable and relaxed" nation, Noble points out the pertinence of this sensibility in capturing a truism about the experience of nationhood:

> [Howard's] use of the affective states of comfort and relaxation is not purely metaphorical—rather, it evokes the embodied experience of homeliness requisite to a successful sense of nationhood. This underlines the link between home and nation: the process whereby we produce the sense of being 'at home' is at the same time the way we articulate ourselves to a larger social environment and its social relations and cultural categories.[41]

For many ethnic communities the stereotypes portrayed by the likes of Nick Gianopolous' in "Wogs Out of Work,"[42] were relaxed and comfortable caricatures, representations they could relate to, and which consequently made them feel at home. It was a step beyond the relocation and transportation of their village culture/s to their new homeland. It signalled the moment in which 'wogs' became part of the mainstream culture they were now part of, when they were accepted as part of Australian life (albeit

[40] Nobel, G. *Op. Cit.* p54.

[41] Nobel, G. *Ibid.*

[42] *Wogs Out of Work* was a successful stage play by Nick Giannopoulos, Simon Palomares and Maria Portesi. It featured comic characters based on Australian migrant stereotypes and spawned several spin-offs, including the TV show *Acropolis Now*, and the feature film *The Wog Boy.*

sometimes as ‘jokes’).

And so while the stereotypes this humour portrayed weren’t ones I could relate to - the Effie character from *Wogs Out of Work*, with her cacophonous wog-fro hair, thick makeup, high heels and painted nails, was a long, long way from where and how I identified - they did signal a shift in cultural imaginaries, and demonstrated that hybrid forms of identity were beginning to take shape.

New World, New Nostalgia

"With all due respect to my family, and the culture that I belong to, because it is my culture, I truly feel blessed, because I don't think that I'd be in the situation I am in here, over there." [43]
Interview Subject, February 2012

Certainly by the turn of the twentieth century global cultural perspectives on identity and nostalgia were well and truly transforming. The cold war had ended; terrorism had become an internal (not just external) sovereign threat; mass telecommunications technology had proliferated, moving from analog to digital, accelerating content reproduction and distribution while

[43] Interview Subject #2, February 2012

transforming social and cultural connections; sheep were cloned; postage stamps became electronic; the traditional borders of nation states blurred with the deregulation of global trade and the movement of corporations into foreign markets.

Gökçen Karanfil, a Turkish sociologist, writing about the experiences of Turkish-Australians during this period, describes the effect of satellite broadcast media as one of a “second loss” of national identity.

> Satellite television from Turkey, by connecting Turkish-Australians to contemporary Turkey and showing them how ‘their’ Turkey has changed, brings the realization of their second loss. For many Turkish-Australians, television from Turkey, with its unfamiliar images, drives home the notion of ‘no return’ (Brah, 1996). It clarifies to them that the precious home they had so clung to and kept alive in their imaginations no longer exists... rather than bringing the familiar into the living room, it actually acts as a reminder of the loss of the homeland.[44]

But he argues that while this second loss has tragic connotations,

> the experience itself actually forces these subjects to rethink, renegotiate and rearticulate their Turkishness and Australianness

[44] Karanfil, G. 2007, 'Satellite Television and its Discontents: Reflections on the Experiences of Turkish-Australian Lives', *Continuum*, 21:1, p59 - 69

> through detaching them from a mythologized homeland. Through the being reminded of the loss of the mythologized homeland, the immigrant is left with no choice but to find new languages for self-identification. These new languages on the one hand make references to a ‘conventional’ national belonging, and on the other transcend it.

Karanfil draws a distinction between the experience of Turkish migrants in Australia and those in Europe, citing Australia’s geographic distance as a major contributor to the Australian migrants’ sense of loss. Unlike European migrants, who are easily able to travel back to their homeland and are therefore accustomed to any changes that take place there (and indeed, often feel that they are part of those changes), Australian migrants are not able to return with frequency. And so the disconnection is more acute.

But even in Europe, for example, within the cultural identities that were transformed (or that disappeared) in the wake of the collapse of the Soviet Union, ideas around nostalgia were being similarly re-conceived and re-configured.

Reflective, Hybrid, Comfortable

Boym dissects nostalgia into two prevalent types: "restorative", which, as stated above, seeks a return to an original state and the recovery of an absolute truth (somewhat similar to Jameson’s idea of how nostalgia operated within a post-modern context), and "reflective," which Boym describes as a meditation on history, on the passage of time, and which perpetually defers a homecoming by cherishing shattered fragments of memory:

> Reflective nostalgia does not pretend to rebuild the mythical place called home; it is enamoured of distance, not of the referent itself. This type of nostalgia is ironic, inconclusive, and fragmentary.[45]

Reflective nostalgia sits neatly in Boym's artistic, Off-Modern art practice:

> There is in fact a tradition of critical reflection on the modern condition that incorporates nostalgia, which I will call off-modern. The adverb off confuses our sense of direction; it makes us explore side shadows and back alleys rather than the straight road of progress; it allows us to take a detour from the deterministic narrative of twentieth-century history. Off-modernism [offers] a critique of both the modern fascination with newness and no less modern reinvention of tradition... For many displaced people from all over the world, creative rethinking of nostalgia [is] not merely an artistic device but a strategy of survival, a way of making sense of the impossibility of homecoming.[46]

The Turkish migrants in Karanfil’s work, jarred by the unfamiliar images of

[45] Boym, S. Op.Cit. p49.

[46] Ibid p xvi-xvii

their home country beamed via satellite to them, or the *Salitani* villagers, coping with the physical destruction of their home, are both examples of groups who have been forced to rethink their own nostalgia in this reflective, off-modern way.

While Boym’s own photographic art practice engages with these ideas in a manner to produce abstract, rather than documentary, photographic pieces, her ideas are evidenced in all kinds of creative practices, including representational ones. Take her example of the Berlin craftsman, originally from Koenigsberg (an East German city), who created models of Berlin's destroyed buildings from old photographs and descriptions. After spending several years in a Soviet labour camp, during which time Koenigsberg was destroyed, he was resettled in East Germany, from where he made his way to Berlin.

It was clear for him there was no returning to his native city. Now he was a real Berliner. Remaking the destroyed interiors [of Berlin], shaping every detail with his own hands, was his grief work, a manual labour of memory. Never mind that these were not the interiors of his native city, not the interiors he remembered. For him there was an integral connection between psychic and urban interiority. He was not building megalomaniacal monuments and symbols of political battles lost or won. He was meticulously reconstructing miniature organs, the curves of capitals, mending the destruction carefully and joyfully in his public dollhouse of memory. For him buildings had become anthropomorphic, innocent victims of ideological battles. The only way he knew how to "manage the past" was with his hands.[47] In other words, he had to create a partly-true, partly-fictitious version of his past that could accommodate the grief caused by what was irretrievably lost.

While Boym (like Tornatore, like Jameson) is well aware of nostalgia's dark side, “its capacity to confuse the actual home and the imaginary one [and create a] phantom homeland"[48], she champions its reflective side as a way forward:

> Reflective nostalgia dwells on the ambivalences of human longing and belonging and does not shy away from contradictions of modernity. Restorative nostalgia protects the absolute truth, while reflective nostalgia calls it into doubt - it does not follow a single plot but explores ways of inhabiting many places at once and imagining different time zones; it loves details, not symbols.[49]

In the generation that followed my aunts, uncles, and parents, I could see traces of this reflective nostalgia sprouting - cousins forging their own independence, challenging the cultural expectations they were raised within, creating spaces for newer histories and memories to accompany and parallel existing inherited ones. Like the way my devoutly Catholic cousin was proud to have friends who were gay, or my other cousins, whose unmarried adult

[47] Ibid p175
[48] Ibid p xvi-xvii
[49] Ibid pxviii

children still lived at home, *with* their respective boy/girlfriends (a typically Aussie *Packed to the Rafters*[50] scenario, they would often boast).

Ang frames this in terms of *hybridity*, which she marks as a departure from what she calls the absolutising tendency in the vision of diaspora:

> ...hybridity also has radical ramifications for how we think of different “peoples”. Indeed, as hybridisation consists of exchanges, crossings, and mutual entanglements, it necessarily implies a softening of the boundaries between “peoples”: the encounters between them are as constitutive of who they are as the proceedings within...The result, after many centuries of contact history, is a profoundly hybridised world where boundaries have become utterly porous, even though they are artificially maintained.[51]

Hybridisation was what the cousins were recreating and embracing. As the fabric of the community of *Salitani* in Australia began to change, the sense of place the village inhabited began to change with it. Geographies shifted and transformed, home was more than one place, psychological landscapes mutated, difference and hybridity became possible. Everyone could relax and breathe a little. Things were becoming *comfortable*.

[50] *Packed to the Rafters* was an Australian commercial TV drama that featured a modern-day middle-class family of parents and adult children (and their partners) all living under one roof. They were, in the words of network promotional material, a “family every Australian can relate to.” <http://au.tv.yahoo.com/packed-to-the-rafters/> viewed 4 April 2013

[51] Ang, I. Op. Cit. p6-7

And so, the concepts of reflective nostalgia, confortableness, and hybridity have provided a useful angle from which to approach my project. Their capacities to encompass ambiguities and contradictions, to transcend absolutes, to accommodate alternate perspectives, opinions, and histories, have enabled me to work on a canvas of openness and possibilities. Using these ideas as a basis for the creative framework allowed the work to avoid making definitive statements or assertions about its subjects, and instead invite them to be part of an unpredictable, randomly formed, unfolding narrative. It has also suggested (and supported) an approach to what I'm calling the story-design of the final work, which I discuss further in Part Two of this exegesis.

Photographic Objects, Photographic Places

"I always have a feeling with photographs that they exert a pull on the viewer and in this entirely amazing manner draw him out, so to speak, from the real world into an unreal world. That is, a world which one doesn't exactly know how it is constituted but of which one senses that it is there." [52]
W.G. Sebald

While reflective nostalgia and hybridity, as philosophical concepts, were useful in forming a bedrock for the project, there were still unanswered questions about the medium through which the project would manifest. I planned from the outset to work with photographs, so it was important to be

[52] Patt, L. Dlilbohner, C. Zizzelsburger, M. Lerm Hayes, C. 2007, *Searching for Sebald: Photography After W. G. Sebald*, W.G.Sebald in conversation with Christian Scholz p105

clear about how they might operate in relation to memory, identity, nostalgia, and place. An initial and crucial question was: were there any specific cultural traditions around sharing, storytelling, and identity that might be significant when designing a photographic project/product?

Historically, photographs and photographic objects[53] have played important roles as repositories for both collective and individual memories, binding people to a sense of self and to each other.[54] Hiroshi Sugimoto, a Japanese photographer, calls photography "a system of saving memories. It's a time machine in a way to preserve the memory, to preserve time." [55] In the digital realm photographs continue to perform such roles and continue to be culturally significant. For example Facebook, with its 500+ million users, attributes the enormous growth of its social network to its built-in photo sharing application. While having limited functionality (tagging, commenting, sorting), the application's photographic and social aspects mean it is used five times more than all competitor photo-sharing applications combined.[56]

From a more culturally specific perspective, it was important for me to note the Sicilian photographic tradition placing family and community at the centre of photographic practice and consumption. Ferdinando Sciannia, Sicily's

[53] Here I'm distinguishing photos (either printed or on a screen) from photographic objects like photo albums, framed photos, lockets etc.

[54] Badger, G. 2007, *The Genius of Photography: How Photography Has Changed Our Lives*, Quadrille Publishing, UK.

[55] The Museum of Contemporary Art, Los Angeles. Viewed 11 May 2013. <http://www.moca.org/pc/viewArtWork.php?id=96>

[56] Seigler, MG. "Zuckerberg: Facebook Photos Used 5 Or 6 Times More Than Competitors — Combined." *TechCrunch*. Viewed 27 August 2010. <http://techcrunch.com/2010/08/27/facebook-photos-usage/>.

preeminent photographer, began his career by documenting the changing culture of his own hometown, Bagheria. In the 2002 retrospective publication of this work he stated: “I believe that the ultimate ambition for a photograph is that it end up in a family album.”[57]

Within Sicilian culture photographs remain cherished objects, and have enormous emotional power. Research conducted with my extended family members bears this out. In the wake of the earthquake, for example, when it was deemed safe to return to the village and recover what could be salvaged, photographs were considered highly valuable. They contained so many memories and emotions of lives now destroyed, that it is not surprising this would be the case. Loretta Baldassar cites the experience of "seeing" loved ones as crucial to maintaining a sense of connection across distance.

> People regularly spoke of the need to "see" their loved ones..."Seeing" each other also reassures... that despite the distance, they still have a son or daughter, mother of father.[58]

Clearly then photographs were an eminently appropriate medium through which to express the stories, memories and emotions of this particular community, in particular of the way in which it has dealt with its own dislocation. What was interesting, in the course of my research , was the extent to which physical photos (as opposed to digital ones) still occupy a

[57] Sciannia, F. 2002, *Quelli di Bagheri,* Peliti Associati, Italy, p 13.

[58] Baldassar, L. 2007, "Transnational Families and the Provision of Emotional Support: The Relationship Between Truth and Distance," *Identities: Global Studies in Culture and Power, Vol. 14 p404*

prominent role socially, emotionally, and psychologically.

Photographs are also important tools in connection to oral histories, which for less literary-based cultures, like Sicilian culture, become important objects in framing oral narratives that both construct and and challenge set notions of identity and the past.[59] Oral narratives in Sicilian culture are, in fact, a 'folkloric' cultural tradition stretching back centuries and employed in part as an assertion and affirmation of an distinct identity, separate from the hegemonic Italian identity reinforced and promoted by the Italian state's official cultural institutions.[60]

On a personal level photographs have played a significant part in my understanding of my own identity. The things I know about my early childhood don't come in the shape of memories at all. They're more like a listing (not even chronologically) of facts. This was, and this was, and this was. They're things people have told me: that I loved to look at books, that I was always marching out ahead of everyone, that my first words were Italian not English. They make sense, but I can't form any kind of picture of them, I can't feel them, I don't remember them.

Any thing I do remember seems to be shaped by old photos. A party for my third birthday, playing rowboats in pyjamas with my brother in the front yard,

[59] Freund, A. & Thomson, A (Eds), 2012, *Oral History and Photography*, Palgrace Macmillian, New York.

[60] See Joseph Pugliese's chapter titled "Noi Altri: Italy's Other Geopolitical Identities, Racialised Genealogies and Inter-Cultural Histories" in Gaetano, R. & Turcotte, G. (Eds) 2008, *Literary and Social Diasporas: An Italian Australian Perspective,* Peter Lang International, Brussels.

saying goodbye to an aunt on the front steps of our house. The snapshot helps me flesh these scenes out a little more, add extra images, close-ups, wide-shots, different angles. And some sound. Everything's out-of-focus and muffled. Faded, like the photographs themselves. But when I look at the photos my body creates a vague physical memory of what it was doing at the time. It's something the Swedish philosopher, Mikael Pettersson, has explained in terms of proximity and state:

> Photographs... somehow have a unique capacity to, phenomenologically speaking, put us in the proximity of what they are photographs of. We feel somehow close to the subjects of photographs and this feeling of closeness to the subjects depicted is typically not present when we view a comparable painting or a drawing...photographs are traces [and] traces are (often, though not always) states. A footprint in the sand is a state of the sand, and sometimes we say it is a trace of the person who stepped there, or of his or her foot (both being cases of things), and at other times we say it is of the event of the stepping. [61]

Without photographic triggers, many of the memories I have of my childhood would not exist. I would not experience an intimacy with those moments and feel them as part of myself. My life as a young child would be predicated solely by the stories of others, not by my own senses.

[61] Pettersson, M. 2011, "Depictive Traces: On the Phenomenology of Photography" *Journal of Aesthetics and Art Criticism* 69 (2):185-196

This is no doubt why photographs are treasured by so many. Daniel Gilbert, in his non-fiction book “Stumbling on Happiness”, writes:

> When people are asked to name a single object they would try to save if their home caught fire, the most common answer is ‘my photographs.’ We don’t just treasure our memories, we *are* our memories.[62]

My aunt’s utter relief at the recovery, from the earthquake rubble, of her wedding photo albums is testament to this. She’d been married just the year before and, as tradition dictated, a set of photographs of the occasion had been commissioned, at considerable cost, and presented in two opulent albums that took pride of place in the sitting room where guests were received. It didn’t matter so much that the photographer, from a nearby city, had the negatives anyway, and that any lost photos could have been reproduced. What mattered to her were the photos and the albums themselves, which were inexorably linked to what was (and probably still is) the most important day of her life. For my aunt, photographs were somehow intrinsic to her sense of self. For me, photographs have evoked a psychological and emotional connection to who I was.

Greg Noble includes photographs in his discussion of how familiar and familial objects have a crucial role in shaping our identities:

[62] Gilbert, D. 2007, *Stumbling on Happiness*, Vintage, London

> We are, in a sense, always making ourselves “at home” in a larger context—a city, the market, the nation—through our experience of home and family... The extent to which we furnish our house and make it our home, our own, entails in part weaving into everyday experience objects of personal and familial significance which also carry, in often submerged ways, a national experience—as ornaments, memories of childhood, photos, touristic memorabilia, aesthetic artefacts, and so on. This constructs the nation not so much as a project of active affiliation and identification but as the furniture of everyday life, or what we might see as a ‘very banal nationalism’.[63]

Noble is speaking specifically about national identity here, but his comments really apply to the creation of any type of identity. What is also significant in Noble’s observations is the role of intimate and personal objects in this construction. Anthropologist Daniel Miller has devoted much of his research to the study of our relationships to objects. In *The Comfort of Things* he demonstrates how people express themselves through their possessions, using them to mediate interpersonal relationships.[64] In a study that focused on the inhabitants of a London street, for example, the only subject who appeared devoid of subjective identity was also the only one who possessed no objects whatsoever.[65]

[63] Nobel, G. *Op. Cit.* p55

[64] Miller, D. 2008, *The Comfort of Things* p8-17.

[65] Ibid. In Chapter One of his book Miller recounts the story of George (not his real name), whose house had absolutely no decoration: “The emptiness in someone’s surroundings, that leaches away one’s own sense of being, was only enhanced by our experience of George himself…” p14

This significance is exemplified by the fact that today my aunt's wedding photo albums take pride of place in her formal living room, and by all the other photographs hoarded by my family and relatives. They are objects that carry with them an underlying experience of national identity, but which also mediate relationships within the complex, fractured network of community connections.

Interestingly, the photographs that seemed most treasured amongst the *Salitani*, were of the banal, the ordinary, the everyday: family dinners, children playing, people photographed with other family, friends, holidaying, going to school. Candid snapshots. Neither beautiful nor striking. But markers nonetheless of memories, connections, and history. They seemed to operate as mementos, memories of moments in time, as illustrated by Susan Sontag, in her writing *On Photography*:

> All photographs are *memento mori*. To take a photograph is to participate in another person's (or thing's) mortality, vulnerability, mutability. Precisely by slicing out this moment and freezing it, all photographs testify to time's relentless melt.[66]

In another classic text on photography, *Camera Lucida*, Roland Barthes likens the photograph to a primitive kind of theatre, "a kind of *Tableau Vivant*, a figuration of the motionless and made-up face beneath which we see the

[66] Sontag, S. 2001, *On Photography,* Picador, New York p25

dead."[67] Unlike other forms of theatre, performances that would all eventually end, recede into memory, and disappear entirely, the photograph's permanence, its ability to confound time, acts as vivid reminder of its subject's passing. "The photograph mechanically repeats what could never be existentially repeated."[68]

Barthes posits that while a painting, or drawing, or any other form of visual art, was similarly persistent, the subjects in these works were never a facsimile of reality, merely a representation. For Barthes, the photograph's power to represent was exceeded by its power to authenticate[69]. He claimed that a photograph's essence was to ratify what was photographed:

> One day I received from a photographer a picture of myself which I could not remember being taken, for all my efforts... and yet, *because it was a photograph*, I could not deny that I had been *there* (even if I did not know *where*). This distortion between certainty and oblivion gave me a kind of vertigo... I went to the photographer's show as to a police investigation, to learn at last what I no longer knew about myself. [70]

A certificate of presence, he calls it.

[67] Barthes, R. 1981, *Camera Lucida: Reflections on Photography*, Hill and Wang p31-32
[68] Ibid. p4
[69] Ibid. p89
[70] Ibid. p85

Barthes, of course, was writing before the advent of Photoshop, where it would be entirely possible to create a photographic image that included him, even if he was never actually present. In the digital age a photograph can no longer automatically claim any kind of authentic status, but the phenomenon he describes, photography's ability to account for a kind of truth absent in older forms of representation, is still culturally relevant. Technology may have changed the material fact of the photograph - or at least made photographic alteration more ubiquitous (for photos have actually always been manipulated[71]) but our brains still assume a truth when we look at a photographic image, even if we know it may have been altered digitally. Or, as Meredith Jones, a cultural studies theorist writing about Photoshop and what she calls "media bodies" puts it, even if we *expect* an image to have been adjusted, "we continue to demand that photographs represent the real."[72]

Before ubiquitous camera phones and cost-free digital reproduction, photographs were reserved for special occasions: birthdays, weddings, funerals, graduations. A verification and reminder of passages passed, lives lived. Photos were a way of locating yourself, bearings that tied you to a path and a name, to who you were, your reality. Before they were any kind of "art" they were a record. Even if they lay in gold-embossed albums gathering dust

[71] The work of Gustave Le Gray in 1855-56, where he developed a technique for producing "plausible" cloud effects in landscape pictures with over exposed skies, or the 1957 composite creations of Oscar Rejlander, made up of 32 different negatives, are both examples of early photographic manipulation. See Gerry Badger's *The Genius of Photography*, p242

[72] Jones, M. 2012, " Media-bodies and Photoshop", in *Controversial Images: Media Representations on the Edge*, Palgrave McMillian, London

in formal sitting rooms, or in edge-worn shoe boxes in the corner of dark cupboards - and seldom actually looked at - they attested to experiences, presences felt, histories that happened.

Photographs have always been more complex than merely statements of fact. They invite ambiguity and interpretation. Annette Kuhn and Kirsten Emiko McAllister, cultural theorists writing about photography and memory, say that photography “as a medium, disrupts the possibility of asserting a single, final Truth.”[73] The time shift between when the photo was taken, and when the photo is viewed, recognises that there are at least two viewpoints. And there is something about the flexibility of the photograph, as an object open to interpretation, that makes it a useful tool in the reconstruction of memory and identity. Susan Sontag describes the way photographs became a method through which families construct portrait-chronicles of themselves: portable kits of images that bear witness to the family’s connectedness. As the institution of family began to undergo radical surgery in western cultures, Sontag notes photography’s arrival as a tool to memorialise and symbolically restate the disappearing continuity and extendedness of family life. “Those ghostly traces, photographs, supply the token presence of the dispersed relatives. A family's photograph album is generally about the extended family- and, often, is all that remains of it.”[74]

[73] Kuhn, A. & McAllister, K. E. Eds. 2006, *Locating Memory - Photographic Acts*, Oxford, Berghan Books p15.

[74] Susan S. 2006, “On photography”, *Communication in History: Technology, Culture, Society* (5th Edition) Part V • Image Technologies and the Emergence of Mass Society, David Crowley (Author), Paul Heyer (Author), Ally & Bacon, p174 - 178

A Family Album

Family photo albums, then, become objects that represent family, as Sontag suggests, especially during periods when the continuity and comfort of family life is disrupted. They also contain family, as Noble might characterise, operating as a intimate objects that reinforce a sense of comfortableness and home.

If my goal was to deploy photographs to explore notions of homeland, community, and identity, and do this in a way that was relevant to the community of people at the centre of this research, it was evident that the most meaningful form any photographic objects could take would be in the shape of an album-like collection. An album-like collection would attest to the community's history and presence through generations, it would be familiar and comfortable, it would live in a family home (rather than in an art gallery) and be something to be touched, read, maybe even added to. An Album-like object was also quite culturally appropriate.

But I had a further question: If an album-like object could contain and represent, and be something familiar and solid, could it also incorporate ideas of hybridity and reflective nostalgia, reinforcing a sense of the hybrid identities and histories the community now experienced, and the geographically dislocated reality the community now inhabited?

Given all of this it seemed logical to allow the photographs to exist in both a physical and digital context. Several boxes, in the form of shoe boxes (of the type that maybe stored under a bed etc), handcrafted and made of wood, containing their own set of individual photos, could inhabit a physical realm. There could be space in the box for additional photos to be added, thereby allowing for future additions and permutations of images. The boxes could live in separate family homes, their photos available to be perused and browsed, mixed and re-ordered. The wood of the boxes would make the containers both special and tactile.

It also seemed logical to create a digital album for a touch-screen device that contained all the photographs. Its touchability would play into all the psychological and cultural associations around intimacy and identity that touch evoked (see Part Two - Platform), and it could offer a form of interaction suitable for a range of different age groups[75]. A digital object would also have the possibility of being deployed across geographic distance, allowing users in different locations to experience the photographs and create conversations around them. Its digital nature would allow hybridity and reflection to be developed as an interaction model. Instead of searching for specific photos, users could be asked to select specific keywords and be presented with relevant but random images. These images could form collections they would be able to share with others.

The photographic objects ultimately created for *Nostalgia Ti Frega*, then, can

[75] Research with older members of the community indicated that user interaction with touch-based devices was intuitive and easy to comprehend.

be understood not just as archival records, but also as tools through which a community of villagers are able to form a complex sense of who and what they have become 40 years after the destruction of their village.

PART TWO - PLATFORM

> *"Official culture still strives to force the new media to do what the old media did. But the horseless carriage did not do the work of the horse; it abolished the horse and did what the horse could never do."*
>
> Marshall McLuhan[76]

> *"Unlike the graphic arts, drawing or photography, unlike tracings, the rhizome pertains to a map that must be produced, constructed, a map that is always detachable, connectable, reversible, modifiable, and has multiple entranceways and exits and its own lines of flight."*
>
> Gilles Deleuze and Felix Guattari [77]

What has always excited me about new media technologies is their potential to enable new kinds of storytelling. Just as the advent of cinema eventually led to a screen-storytelling language and completely new kinds of story experiences for audiences, the use of digital technologies as storytelling media are similarly allowing new languages to develop and new kinds of stories to emerge.[78]

Notwithstanding the fact that, at the outset digital technologies tended to ape

[76] McLuhan, M. 1970, Quoted in *Hertzian Tales*, Anthony J. Dunne (2006) MIT Press, Cambridge, p30

[77] Deleuze, G & Guattari. F, 1987, *A Thousand Plateaus: Capitalism and Schizophrenia*, Continuum Press Edition, 2004, London. p7

[78] See Jenkins, H. 2006, *Convergence Culture: Where Old and New Media Collide*, New York University Press, New York

rather than innovate, they are still capable of producing stories that were never possible with older storytelling forms.[79]

The most significant of these new possibilities are, firstly, a result of readers/viewers now being able to actively and immediately interact with stories and affect their outcome and/or contribute their own content and, secondly, the ability for digital platforms to easily present non-linear stories that can be highly customised.[80] It's very possible, for example, for a *Mad Men*[81] viewer to supplement their engagement with the story by following a number of Twitter fans who tweet as *Mad Men* characters.[82] These (often comic) tweets are not part of 'official' *Mad Men* storytelling, but they add layers, texture, and narrative offshoots to the overall story experience, and are accessible by anyone with an internet connection. They're not necessary story fragments, but rather augmentations that a fan can participate in, if they so choose. Similar augmentations are found for *Twilight*, *Buffy*, *The L Word*, etc.

Digital platforms also now offer storytellers and their audiences the ability to create complex non-linear information structures that allow for singular, highly customizable user experiences which are open-ended, constantly evolving,

[79] I'm meaning here analog technologies (TV, radio, video, audio etc) and printed literature (books), but not oral and non-recorded storyteling forms that predate the development of printing and electronics, and in which interaction and non-linear progressions formed part of narrative structures.

[80] Analog platforms are able to do this, as I will demonstrate with examples later in this section, but not with the ease and simplicity that digital platforms can.

[81] A popular television drama produced by the AMC cable network in the US <http://www.amctv.com/shows/mad-men> viewed 20 July, 2013

[82] Pickhard, Anna. 2009, "I Wanna Be Your Blog", *The Guardian* newspaper, viewed 14 September 2009 <http://www.guardian.co.uk/technology/2009/feb/07/mad-men-twitter>

and where the users themselves have a significant influence on the evolution of both the information and structure of the experience itself. This means audiences are now storytellers themselves.

These platforms are less predictable, more adaptable and open to change, and therefore potentially more dynamic and exciting. They encourage participation from multiple users, offer several narrative-threads simultaneously, and are non-hierarchical in emphasis. To that end they are adept at replicating a type of interactive story structure that is prevalent in organic and social systems, but which has been largely absent in traditional media formats. They can be entered and exited at any point, can grow and evolve, and, significantly, can affect the very nature of the content they relay. These web-like, or 'rhizomatic' structures, afford multi-faceted, complex systems that, if designed well, take on lives of their own and become, in effect, part of their own story. They provide a framework for innovation both in form *and* in content. As a storyteller, this is very exciting.

This type of open-ended, evolving story experience, while arguably part of oral storytelling traditions,[83] has been, prior to the proliferation of digital social media technologies, historically absent from traditional broadcast media storytelling. This section looks at these new possibilities, charting pre-digital and current digital experimentations with narrative and user engagement/participation, and working to position the storytelling and user

[83] See Ong, W. 1982, *Orality and Literacy: The Technologizing of the World*, Routledge, London

experience approach of *Nostalgia Ti Frega* within this context.

Story, Narrative, and Experience

In the digital space ‘story’ or ‘narrative’ are often folded into the term ‘experience’. That is, ‘experience’ can be thought of as ‘story’. This is partly because the presentation and organisation of content in digital media has traditionally been done by information architects, who historically have been concerned with the structural design of shared information environments. But as storytelling in the digital space has become more complex, with technical advancements allowing for the inclusion of richer and more varied media, and increased levels of participation and interaction by users/viewers, the creation of content has become intertwined with its organisation. The somewhat dry discipline of Information Architecture has evolved to include User Experience (UX). And when UX designers are asked to define 'experience', what they describe seems similar to what filmmakers or television producers or writers mean by 'story.'

Jodi Forlizzi and Katja Battarbee, Human-Computer Interaction (HCI) usability researchers, describe experience in these terms in their examination of interactive systems:

> An experience is more coalesced, something that could be articulated or named. This type of experience may be characterized by a number of product interactions and emotions, but is schematized with a

> particular character in one's memory and a sense of completion. An experience has a beginning and an end, and often inspires behavioral changes in the experiencer.[84]

Note the significance of a beginning and an end. For Forlizzi and Battarbee experience design takes a very broad definition. They liken television news producers or even dinner party hosts to experience designers. They are all creating a context, inviting engagement and participation, aiming for affect, and letting a story unfold from the process.

Meanwhile, from point-of-view of authors and screenwriters, ‘story’ is very much understood as ‘experience’. Robert McKee, the film scriptwriting guru, describes story in this way:

> To be entertained is to be immersed in the ceremony of story to an intellectually and emotionally satisfying end. To the film audience, entertainment is the ritual of sitting in the dark, concentrating on a screen in order to experience the story's meaning and, with that insight, the arousal of strong, at times even painful emotions, and as the meaning deepens, to be carried to the ultimate satisfaction of those emotions.[85]

[84] Forlizzi, J. and Battarbee, K. 2004, *Understanding Experience in Interactive Systems*, Cambridge, MA: Association for Computing Machinery

[85] McKee, R. 1997,*Story: Substance, structure, style, and the principles of screenwriting*. Methuen, Chatham, UK,p4.

Given the correlation then between experience and storytelling, I understood that wearing a UX Designer cap on *Nostalgia Ti Frega* would be more than just organising how the 1000+ photographs would be presented. It would be to design a compelling experience that users could engage with, contribute to, be part of, and would ultimately tell an interesting story.

Before I could determine how I would do this I needed to understand how people create stories and form their own experiences. In particular I wanted to observe how the users I was targeting connected and engaged with the world around them. It would provide clues into what kind of design might work best to facilitate engagement with the interactive experience/narrative of my project, but also shed light on how the design might engender the ideas around reflective nostalgia, hybridity, and comfortableness that I wanted to explore.

Making Order Out of Chaos

In his book *Closure*, Hilary Lawson, the contemporary philosopher, describes the world as a mass of random chaos, or a 'page of dots,' which we connect into recognizable shapes in order to make sense and form meaning. He calls this process 'closure', and describes it as the following:

> Suppose that we are looking at a random pattern of dots on a page. If asked what can be seen amongst the dots, we can imagine scanning

> the pattern looking for some combination of dots that allows the formation of an image of some sort. To begin with nothing may be seen other than the dots, but in due course let us suppose that an image of a face is identified. Having found the face the dots are no longer a random pattern. Instead we have the experience of seeing a face, of discerning perhaps the eyes and nose, or even an expression. The page of dots is now not what it was. The dots appear to be the same yet we see something which we did not previously see, which we can identify and which was previously absent. This thing which we see is an example of a closure: the outcome of a process of closure... closure can be understood as a process which generates something from a space of possibility.[86]

For Lawson, closure is the method we use to construct and understand experience, it is what we use to shape the constant stories we tell ourselves. It follows, therefore, that closure is deployed when people engage in reflective (as opposed to restorative) nostalgia. Looking back in time, through photographs and shared stories, personal histories and understandings are woven together and sense is made of the disparate, sometimes contradictory factors that make up a life. On each occasion I had the opportunity to look through photos with any of the villagers/relatives I was photographing, I understood that the process of looking and being reminded of things forgotten meant that memories were being re-remembered, and even often reconfigured. Memory was not fixed, it changed a little with every new

[86] Lawson, H. 2001, *Closure*. Routledge, London, p5.

(present) context the photo was viewed within.

But on each occasion it was *analog* (i.e. printed) photographs that we looked at. Upon viewing images I observed there was a distinct pleasure in re-remembering what had been forgotten. It was as though the photograph itself was where the individual's memory resided, brought back to life upon viewing. Sitting at kitchen tables, or on living room sofas, sipping espresso and snacking on home-made amaretti, it was a slow, reflective process, conducted without hurry. The analog nature of the photographs, their *physical* presence, and hence their durability, afforded the process a broad temporal dimension. The photos, as repositories or containers of memories, would always be there. They would endure. So there was no need to rush. The memories were preserved.

Too Much Information

What then of a digital context? Data moves rapidly, feeds are updated constantly, content is churned and replaced, images are transient (and possibly, probably, photoshopped). Images in this space aren't so solid, they can be altered, and they're easier to lose. They don't operate so well as containers or repositories because there's no guarantee they'll be there the next day. In fact, you can safely assume they won't be.

Coupled with this is the sheer volume of data now available in the digital space. Not only does information come at us faster than ever, and stays with us for shorter periods, there’s also so very much of it.

Richard Saul Wurman, an architect and graphic designer, coined the term ‘information anxiety’ to describe a state where the amount of data available becomes so overwhelming, and its ability to be contained no longer tenable, that a state of anxiety arises.

> The glut [of information] has begun to obscure the radical distinction between data and information, between facts and knowledge...Take the news as an example. Everyday the media seek to deliver us larger amounts of news at a faster rate. We are besieged with accounts of the world in amounts that are impossible to process. And as we scramble to keep up with the news race, we are more likely to make errors of perception...the more time we spend with reports of separate events, the less time we have to understand the "whys and wherefores" behind them, to see the patterns and relationships between them...Instead we are lulled by a stream of surface facts, made numb, passive, and unreceptive by a surfeit of data that we lack the time and resources needed to turn into valuable information.[87]

This is a system where Lawson’s ‘page of dots’ (to recall that example) are so numerous and overwhelming that closure or containment is no longer possible.

[87] Wurman, R, S. 1989, Information Anxiety. New York: Doubleday, p37.

Wurman sees a solution to this ever-expanding page of uncontainable dots. In laying out a framework for the 'anxiety-proofing' of information he suggests, amongst other things, that users need to be enabled to find personal pathways to understanding, to lay claim to a feeling of "owning" information, to select the pertinent from the superfluous, and to find the appropriate organizing principle for different subjects. He cites access as the primary antidote to information anxiety, made possible by structures unique to their specific subjects and that allow users to find what interests them and feel no guilt about what doesn't.

Unlike the traditional tree data structures[88] of many information systems, where information is organised into fixed, ordered paths that leave little choice for the user other than to follow that path (an example is the directory structure of a personal computer), Wurman's advocation of structures that enable personal pathways is echoed by Sigmund Rinde in his Thingamy Blog post "Tree Structures: Are We Hard-Wired?"

> If I say "Book" and "42", what comes to your mind? Some "Hitchhiking" perhaps? What with "Umbrella" and "Film"? Any film titles coming to mind? Add "Nanny", "Flying"... would that be "Mary Poppins"? See? Your brain is quick and naturally wired to intercept... tags! Nevertheless, I often hear that tree-structures are necessary and that we are hard-wired for tree-structure organizing: Hierarchies,

[88] For further discussion on data structures see Knuth, D. 1997, *The Art of Computer Programming*, *Volume One, Fundamental Algorithims*, Addison-Wesley Professional, London

> folders...That I would argue, humbly of course, is bogus. Would "Film > British location > Female lead > Disney > Family > etc." trigger "Mary Poppins!" as promptly? Nah, didn't think so... We're naturally inclined to tag imprecisely and freely - and locate any object or subject intercepting those tags. Fast, efficient and without training. And without any quest for standards. In other words, we need no classic tree structured data sorting. We would be better off without.[89]

In a period where traditional forms of narration/storytelling/sense-making are in a state of transformation, and where digital technologies are enabling proliferation and mutability, what kinds of structures can enable the kinds of things Wurman and Rinde are seeking? Is it possible to enable experiences that are more than just the reception of surface facts, that empower users to construct their own meanings and form their own ‘closures’? And are the structures of these experiences able to act also as containers, or repositories, to be both enduring but allow space for the changeable, transient nature of the digital ebb and flow?

Rhizomes can Act as a Place

Rhizomes are web-like, non-linear, de-centralized structures that can be entered or exited at multiple points. In botany, a rhizome is a particular kind of root, "a horizontal, usually underground stem of a plant that often sends out

[89] Rinde, S., 2005,"Tree structures, are we hard-wired?", Thingamy Blog, viewed 19 August, 2005, <http://thingamy.typepad.com/sigs_blog/2005/07/treestructures_.html>

roots and shoots from its nodes."[90] Ginger is an example of a rhizome. And importantly, just like fingerprints, no two rhizomes are ever identical.

European philosophers Gilles Deleuze and Felix Guattari developed a theory of Rhizomatics in their 1987 publication, *A Thousand Plateaus: Capitalism and Schizophrenia*. They were offering alternatives to understanding how power, and its action, operated. Instead of hierarchical systems, like Rationalism, Essentialism, or Capitalism, which Deleuze & Guattari likened to trees (i.e. centralized roots feeding the one plant), the rhizome was an "acentered, non hierarchical, non-signifying system without a General and without an organizing memory or central automaton, defined solely by a circulation of states."[91]

They summarized the principal characteristics of rhizomes as follows:

> ...unlike trees or their roots, the rhizome connects any point to any other point, and its traits are not necessarily linked to traits of the same nature; it brings into play very different regimes of signs, and even non-sign states. The rhizome is reducible to neither the One or the multiple. It is not the One that becomes Two or even directly three, four, five etc... When a multiplicity of this kind changes dimension, it necessarily changes in nature as well, undergoes a metamorphosis... The rhizome is an antigenealogy. It is a short-term memory, or antimemory. The rhizome operates by variation, expansion, conquest,

[90] *The Free Dictionary*, viewed 4 April, 2013 <http://www.thefreedictionary.com/Rhizomatic>
[91] Deleuze, G & Guattari, F, Op. Cit.

> capture, offshoots.[92]

Although Deleueze & Guattari developed this theory before the advent of digital information networks, it's not hard to see how, with their capacity to have nodes connecting to other nodes, their potential for non-hierarchical structures, their multiple entry and exit points, digital information networks can easily be likened to rhizomes. The internet itself is like a huge, ever expanding rhizome.[93]

It is also important to note, in relation to my own project, the resonances here with the ideas around reflective nostalgia. Reflective nostalgia, with its ambiguity, lack of conclusion, fragments, back-alleys and detours, is meandering and rhizomatic. Part of why presenting the photographs of *Nostalgia Ti Frega* in a hybrid analog-digital context is so appropriate is the way that non-linear, rhizomatic, reflective story experiences are so readily engendered. The analog can be part of the digital (and vice versa), both spaces referring to and impacting on each other. And the rhizome, as Greg Nobel and Daniel Miller[94] might argue, can act as a container, a place where memories can be stored, relived, retold, reconfigured. In this sense, rhizomes can act as a 'place'.

[92] ibid.

[93] See Hamman, Robin B. 1996, *Rhizome@Internet: Using the Internet as an Example of Deleuze and Guattari's "Rhizome"* (online). Available via World Wide Web, URL http://www.swinburne.infoxchange.net.au/media/halm316/gallery/david/pg11b.htm (accessed July 2013).

[94] See previous section Place where both Noble and Miller's work on material culture and how objects can act as containers of identity.

Artists and authors have long experimented with rhizomatic forms of storytelling. There are thousands of examples, but some of the most notable include:

The Fluxus Collective's *Anecdoted Topography of Chance*,[95] published in 1962, is an non-linear adventure through associations, memories, and anecdotes evoked by a group of friends from a collection of objects lying at random on a table. Described as a novel of digressions, the book is at once a game, an encyclopedia, a cabinet of wonders, and a story of friendship and creativity.

Similarly, Mark Z Danielewski's 2000 novel *House of Leaves* literally embodies the labyrinthine multi-dimensional journey it describes. Telling its story from many viewpoints and in many different literary styles (poetry, prose, journal, even academic criticism) the text breaks up, is inverted, or pushed to extremes on certain pages as readers flick from the front, to references at the back, to divergences in the middle, to the front again. As the central character's consciousness is fragmented, so too is the reader’s as it dips in and out of different contexts.

The Unfortunates by B.S. Johnson, first published in 1969, is an experimental book where each chapter is printed as a separate leaflet and all are presented together in a box. The chapters range from single paragraphs to multiple pages and are designed to be read in any order. Johnson was

[95] Spoerri, D. 1995, *An Anecdoted Topography of Chance*. London: Atlas Press.

attempting to convey the “mind’s randomness”[96] when approaching thoughts and memories, which Johnson observed didn’t arrive in any kind of sequentially ordered manner.

All these examples allow the user to contribute to his or her own experience by giving agency in how the experience will unfold. As rhizomatic structures they achieve Wurman's objectives of being formations unique to their specific subjects (no two rhizomes are the same) and allow their users to engage with what interests them and feel no guilt about what doesn't. There are no set paths. The experience is one of possibility. They also act as repositories of information, organic and malleable, but distinct and discrete. The myriad connections of their structures mark a place.

These examples have all had an influence on the creative evolution of *Nostalgia Ti Frega*. Firstly they support the ideas and theoretical framework that the project was conceived through, namely reflective nostalgia and containment. And secondly, as analog examples of experiments with narrative form which have attempted to redefine narrative logic, they’ve helped demonstrate ways in which my own project might manifest.

A New Language

While a novel affords a certain scope for experimenting with story forms, in traditional broadcast media there is usually little room for experiment.

[96] Coe, J. 2002, “Death by Naturalism” 17 October 2002, weblog, viewed 24 February 2011, <http://www.jonathancoewriter.com/oddsAndEnds/deathNaturalism.html>

Storytelling in film and television production is often a question of adhering to established generic conventions[97]. Film stories, for example, are required to be told (usually) in three linear acts, to have a ‘hero’, who goes on a ‘journey’, and experiences some kind of emotional enlightenment[98]. Or instead, stories should pivot continually around a dramatic question or conflict, with different characters representing different viewpoints and the plot working toward an ultimate resolution, or not[99]. Television, when confined to a single broadcast platform, requires even more generic adherence as programming has strict time constraints and stories must work around advertising segments.

The proponents (Field, Campbell, McKee) of these storytelling approaches naturally draw connections between their approaches and the history of storytelling in (western) culture, and the psychological, social, ethnographic reasons storytelling came about: its very purpose for being. But their rules and guidelines are steeped deeply in a western hegemonic mindset, and so the resulting stories speak profoundly to that.

The story limitations of older broadcast media developed also through a combination of platform capabilities and commercial goals. With cinema, for example, conventions arose to make the high cost of production and distribution as streamlined and cost efficient as possible.[100] What this meant was that only certain types of stories began to be told, and only in certain

[97] See Field, S. 2005, *Screenplay: The Foundations of Screenwriting*, Delta, New York
[98] See Joseph Campbell’s *The Power of Myth* and Syd Field’s *Screenplay: The Foundations of Screenwriting*
[99] McKee, R. Op. Cit.
[100] See Guynn, William Ed. 2010, *The Routledge Companion to Film History*, Routledge, London

types of ways. The Hollywood Studio system saw the rise of genre stories. Experimentation narrowed and, while there were periods where non-conventional storytelling gained critical prominence (like the French *New Wave* or Italian *Neo-Realist* movements), they were not block-buster popular, and any techniques that proved effective were anyway soon adopted by mainstream producers (the jump-cutting of Goddard or the documentary-style techniques of Rossellini, for example, became part of the language of studio films). But what never changed was the linear flow of story, or the ability for viewers to be anything other than passive participants. To consume cinema was to often enter a huge dark cave and passively watch an enormous projection of light - any interactivity (eg. Vivian Sobchack's claims that a viewer actually acted *upon* a film in order to give it meaning[101]) was not done consciously.

Television meanwhile, began to break the stranglehold of the studios[102], creating its own set of conventions for production, distribution and storytelling. But unlike cinema the audience for television was never completely passive[103]. Television in fact had to compete for attention with a myriad of other distractions happening around the television set at any given moment. It was an object within the home, its proportions didn't subsume it. This meant that television networks were more open to experimenting with content and programming and to use anything that would help capture the

[101] See Sobchack, V. 1991, *The Address of the Eye; a Phonomenology of Film Experience*, Princeton University Press, Princeton

[102] See Stephens, M. 2010, 'The History of Television', *Grolier Encyclopedia*, viewed May 31, 2013 <http://www.nyu.edu/classes/stephens/History%20of%20Television%20page.htm>

[103] Ang, I. 1985, *Watching Dallas: Soap Opera and the Melodramatic Imagination*, Routledge, Oxford.

attention of viewers.

By the turn of the century, with the rising ubiquity of mobile phones and internet and the digital revolution taking flight, this attention was becoming harder and harder to both attract and maintain. Television had always been part of the mundane fabric of people's everyday lives. As this fabric was transforming with the introduction of new communication technologies, television understood that to maintain relevance it would need to incorporate new platforms and deliver stories in new ways across them. And since the new communication technologies were providing a new level of interactivity and participation for users, television too would need to provide a similar experience.

Big Brother[104] is probably the best example of this. Unlike the conventional dramas or documentaries) before it, *Big Brother* encouraged participation by a range of users (viewers, contestants, producers, media commentators) in a range of different ways that intentionally had a material impact on the show's various narratives. Whether its was which housemate was nominated for eviction, which live audience member would win best costume, how the housemates would fare in their latest challenge, who the audience would decide to evict, what opinions were expressed by commentators, how housemate identities were constructed by the show producers, the story outcomes were never fixed or predictable. Also, in addition to the television

[104] *Big Brother,* a show that filmed the interaction of 'contestants' living together in a house they weren't able to leave, was created by a Dutch company for Dutch TV. It was later franchised to other parts of the world, becoming a huge international hit. See the Endemol website, viewed 20 July 2013, <http://www.endemol.com/programme/big-brother>

broadcast, which was packaged in a variety of ways (live nomination/eviction, daily update, late night uncut), the show used mobile phone and web technology to broaden the overall structure. Viewers could vote with their mobile or download live video streams from the Big Brother house. They could engage with the website to learn more about the contestants. They could receive updates on events in the house via SMS. In essence the show's producers created a rhizomatic framework against which the many stories of Big Brother would play out, not knowing themselves what the outcomes would be but instead letting them unfold organically.

While many cultural critics have decried the advent of reality television[105], its popularity has been a firm consequence of the sense of agency and connection enabled through the use of interactive technologies. As the use of interactive, customisable applications that allow us to share or re-post content has increased (facebook, wordpress, tumblr, twitter, instagram et al.[106]) our need to have a voice and be part of a conversation has become important to public performance of oneself.[107] The way digital software products allow us to organise and access content in a myriad of ways, creating infinite story paths, has developed our capacity, and indeed our expectations, to engage with non-linear story spaces. Applications like iTunes, which allow multiple options for organizing and accessing songs and

[105] eg “Fear and Loathing on Reality Television: An Analysis of “America's Most Wanted” and “Unsolved Mysteries”” by Cavender and Maupin, *Sociological Enquiry*, Volume 63, Issue 3, pages 305–317, July 1993

[106] MacMillian, G. 2013 "The power of Tumblr: Site now hosts 100 million blogs" weblog, viewed 28 March, 2013 <http://wallblog.co.uk/2013/03/28/the-power-of-tumblr-blogging-site-now-hots-100-million-blogs/>

[107] See McGrath, J. (2004) *Loving Big Brother: Surveillance Culture and Performance Space, Routledge, London*

playlists, or the web-based *flickr*[108], an ever-expanding database of user photos that similarly offers infinite ways to organize, collect, and link photos between users, are good examples. Elizabeth Day, a journalist writing in *The Observer*, explains the needs of modern viewers:

> Yet it seems that 21st-century viewers are looking for more than just simple entertainment. Part of the attraction is the sense of control the *X Factor* gives us: the sense that we can put right wider social wrongs by voting for our favourite contestants and that although our lives are being shaped by forces beyond our control – such as government cutbacks, widespread job losses or social deprivation – the ability to have a say in what happens to others in reality TV shows gives us back a much-needed sense of agency.[109]

It is clear then, that in a context where users are well versed in established narrative forms, and no longer persuaded by the artifice of their conventions[110], new structures are needed to entice and engage users into "leaning forward," i.e actually physically leaning toward their screen in order to interact with it.[111]

[108] *Flickr* website, <www.flickr.com>

[109] Day, E. 2010, "Why reality TV works" *The Observer*, online article, viewed 21 November 2010 <http://www.guardian.co.uk/tv-and-radio/2010/nov/21/why-reality-tv-works>

[110] The BBC Online article "Reality TV enters Mainstream" says: "Drama, lifestyle and documentary programs have all been crushed by the alluring mix of multi-camera, multi-star TV." Viewed 14 September, 2005. <http://news.bbc.co.uk/1/hi/entertainment/tv_and_radio/3020275.stm>

[111] *The State of the News Media 2004 - Annual report on American journalism*, states in its section on Audience:"When it comes to audiences of network news, the headlines are generally grim. Only one or two programs are increasing their audiences. For most, flat ratings are a victory. The networks are showing no real signs of innovation or of creating genuinely new kinds of news programming that might win new audiences." Viewed 14 September, 2005, <http://www.stateofthenewsmedia.org/narrative_networktv_audience.asp?

Nostalgia Ti Frega was never intended to simply be a passive experience of looking at photographs. In a sense, the basic premise of the project, that of creating a family photo album within a digital space, already framed the work within an inherently interactive practice - the shared experience of looking through photos and telling stories about them (as demonstrated by my interview/photographing sessions with the photographic subjects/relatives). And so while some types of interactivity, like user-generated content (e.g.) was outside the scope of what could be reasonably developed within the given time frame for the project, there were many 'lean forward' activities - collecting, curating, commenting, sharing - that could be developed as part of the narrative experience.

In the converged digital space, artists and writers have been exploring the ways in which technical developments might extend and transform story experiences. For example, HBO's *Voyeur*[112] was a cross-platform story experience that combined content on HBO channels, fictional web pages, photos and video clips on Flickr and YouTube, blogs, and mobile. Focusing on the lives of people living in 8 different apartments in the same New York City apartment block, *Voyuer* was aimed at increasing HBO's brand image and promoting the customisable extras and platform offerings their subscribers could choose from[113].

cat=3&media=4>

[112] HBO Voyeur website, viewed 13 November 2010, <http://archive.bigspaceship.com/hbovoyeur/>

[113] Vranica, S. 2007, "Tony Gone? HBO Fakes It" Wall Street Journal news article, viewed 2 April 2013, <http://online.wsj.com/public/article/SB118299651798651029-AVcij9oS7GUH3Y5i7sRDBZoviSQ_20070705.html>

Jonathon Harris, an American artist who combines computer science, statistics, and visual art to craft compelling digital stories, is similarly interested in allowing his audience to customise their own experience of his work. *The Whale Hunt* [114], a web-based photographic chronicle of Harris' nine day journey with Inupiat Eskimos in Alaska, where he documented their traditional whale hunt, offers a pulsating experience through over 3000 photographs, all mostly taken at regular five-minute intervals. Harris' aim here was to match the pace of his own heartbeat. He then created a dynamic framework to experience the story that allowed viewers to rearrange photographic elements, navigate through them in a variety of ways, and "extract multiple sub-stories focused around different people, places, topics, and other variables."[115]

Harris' 2008 interactive installation, *I Want You to Want Me*[116], commissioned by New York's MoMA, focused on online dating and used real dating profiles collected from several dating websites. Shown on a large touch monitor, viewers are able to organise and re-arrange visual elements to reveal specific data and highlight different themes. The ability to create unique story paths through the data helps viewers form the emotional response to and connection with online daters that Harris was seeking to evoke.

These examples all demonstrate nascent experimentations with digital and

[114] The Whale Hunt website, viewed 2 May 2010, <http://thewhalehunt.org/>

[115] Harris, J. Portfolio website, viewed 2 May 2010, <http://www.number27.org/whalehunt.html>

[116] I Want You to Want Me website, viewed 2 May 2010, <http://iwantyoutowantme.org/>

cross platform story forms, all created in the last decade. While the story frameworks are complex and emphasise customised, rhizomatic pathways, none actually incorporate functionality that allows viewers to lend their own voices to story content. As outlined above, however, in the case of many web applications (facebook, flickr, et al) this type of functionality has become a matter of course. Even many seemingly ‘dry’ platforms have increased the options for users to become part of stories (or conversations)[117]. The ability to share, connect, and contribute to stories has become an expectation amongst users (as noted above by commentators on reality TV), and it is leading to an entirely new form of story object, one that Craig Mod, a writer, designer, and publisher, describes as “post-artifact”.

The Post-Artifact

> *The future book — the digital book — is no longer an immutable brick. It's ethereal and networked, emerging publicly in fits and starts. An artifact ‘complete’ for only the briefest of moments. Shifting deliberately. Layered with our shared marginalia. And demanding engagement with the promise of community implicit in its form. The book of the past reveals its individual experience uniquely. The book of the future reveals our collective experience uniquely.* [118]

[117] LinkedIn, for example, began as a fairly basic social network in 2002, with a focus on professional work. Users could fill out a CV-like professional profile and "link" with other users. By 2012 the platform had incorporated functionality that allowed users to post updates, incorporate their facebook or twitter feeds, and join professional "groups".

[118] Mod, C. 2011, “Post Artifact Books & Publishing: Digital’s effect on how we produce, distribute, and consume content” weblog, viewed 20 July 2011, <http://craigmod.com/journal/post_artifact/>

Craig Mod

Mod is principally focused on books, and on what emerging digital platforms herald for the book as an object. He charts the transition books have made from self-contained systems to systems concerned with engagement, sharing, marginalia, and community[119]:

He describes the pre-artifact system as one where the book is made in a system of isolation, involving very few people: the author, the editor, the publisher, a muse (perhaps), but not the reader. The end product of this system is the artifact - the book - which is an immutable object, self-contained, and practically impossible to extend beyond its bindings. Mod calls it a "souvenir of a private journey."[120] Mod goes on to describe the *post*-artifact system as one with more user engagement and community discussion. This system pre-dates the advent of digital technology, but its relative stasis only changed with the advent of the digital.

> Most fundamentally digital removes isolation... The corollaries: an increase in connectivity. Mutability of artifact. Continuous engagement with readers. And most excitingly, a potentially public record of change, comment, discussion — *digital marginalia* — layered atop the artifact, adding to the artifact, and redefining 'complete'... The post-artifact system [is] A system of unlocking. A system concerned with

[119] Ibid.
[120] Ibid.

> engagement. Sharing. Marginalia. Ownership. Community. And, of course, reading. It's the system that transforms the book from isolated vessel for text into a shared interface... Manifested properly, each new person who participates in the production of digital marginalia changes the reading experience of that book for the next person. Analog marginalia doesn't know other analog marginalia. Digital marginalia is a collective conversation, cumulative stratum.[121]

The book, in this instance, becomes an endlessly emergent relational entity - never quite fixed, a story continually evolving and shared.

Readmill[122]*,* an ebook reader for the iPad and iPhone, is the perfect example of this. Inspired by the photograph of a heavily annotated copy of *Ulysses*, Readmill allows users to share annotations with other users, as well as post them to their social media streams.

[121] Ibid.

[122] Readmill website, viewed 12 Jan 2012, <https://readmill.com/>

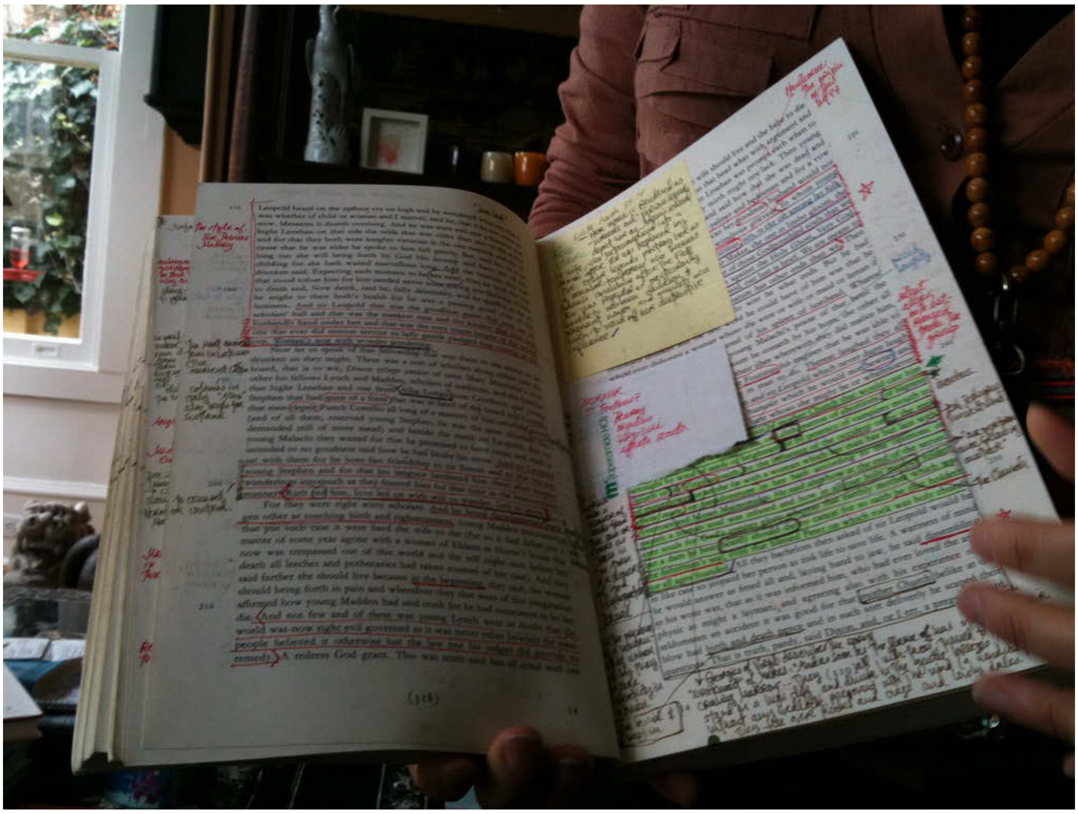


Figure 2: The inspiration behind the Readmill e-book reading application. A key function of the application allows readers to highlight and comment on sections of books and then to share these with others.

The Silent History[123] is a serialized, exploratory novel for iPad and iPhone that is dubbed by the producers as a "new kind of novel." *The Silent History*'s fictional story is presented in two forms, testimonials (written by the authors) and field reports (contributed by readers). The user-generated field reports are only viewable by other readers at the specific location they were created. This means that if you travel to a new city you're able to open the map section of the app, locate any field reports that may have been created there, and then visit those actual locations to unlock the reports. While it's still possible to appreciate the story by reading only the testimonials, the field

[123] *The Silent History* website, viewed 19 February 2013, <http://www.thesilenthistory.com/>

reports add an extra dimension to the overall story world, and allow readers to participate in its evolution.

While these examples for text-based books show the new direction the post-digital artifact is headed in, photography books (or photobooks) have not yet begun experimenting in this way. While there is a swathe of examples of photobook apps (*Capitolio*, *Full Metal Jacket Diary*, *CCcell*, *As It Is Again*, *Bronx Boys*, *Everyday*, *Personal Best* and many others), none of these could be described as post-artifact. Each example presents a portfolio of photos, with a variety of information attached to the photos (captions, text, EXIF data), but none afford the reader an opportunity to either form their own story path through the content, leave personal comments, or contribute to the creation/emergence of the narrative.

Hybrid Form

Given the community focus of my project, and my interest in experimenting with digital storytelling forms, I was keen to create a "photobook" that would challenge the prevalent digital forms that photobooks were taking, and would be an object that could claim a position within the "post-digital artifact" space. But I also wanted the project to exist in an analog space. This felt important, partly because from a Sicilian cultural perspective printed photographs remain significant objects, but also because, in an increasingly digitized world, a nostalgia for analog objects and experiences is growing.

Despite an increasing move into digital realms, we, and our world, are physical entities. We exist in and in relation to a material world, and as significant as our virtual connections may have become, our material context is still present and crucially significant. We understand this materiality as 'real' and as having a solidity and permanence that things in the digital realm do not. Old analog printed photographs certainly problematise veracity far far less than a digital photo seen on a screen. Bernard Steigler, the French philosopher, in addressing the radical change in our relationship to images precipitated by digital photography, calls the analog photograph a "positive" artefact. This is because it is unlikely to have been altered significantly in the darkroom, and therefore carries a more 'objective' weight (i.e. truthful representation of reality).[124] Echoing this is Carol Wilder, who has speculated that the resistance to digital technology (e.g. her son's coveting of her vinyl records and the NY Subway corporation encountering resistance when changing from brass tokens to Metrocards) is because the experience of what is perceived to be more 'real' is more valuable (still) than the experience of the high tech.[125] Sherry Turkle's recent publication, *Alone Together* (2011), surmises that technology's increasing domination of our lives is threatening to make us less human: under the illusion of allowing us to communicate better, it is actually isolating us from real human interactions in a cyber-reality that is a poor imitation of the real world.[126]

[124] Digital Media Archive, NYU Dept of Media Culture and Communicaiton, website, viewed 20 January 2011, <http://cultureandcommunication.org/deadmedia/index.php/Analog/Digital_Transition>.

[125] Ibid.

[126] Harris, P. 2011, ”Social networking under fresh attack as tide of cyber-scepticism sweeps US”, *The Guardian* news article, viewed 23 January, 2011, <http://www.guardian.co.uk/media/2011/jan/22/social-networking-cyber-scepticism-twitter>.

Manifesting the project partly in analog form seemed appropriate, given the rising prevalence of analog nostalgia and the project's focus on reflective nostalgia. And by occupying both analog and digital spaces this hybridity would allow the work to have a thematically appropriate dynamic texture: in part solid and stable but also in a state of cumulative change, fragmentary, transformed by rearrangement, offering multiple experience pathways, being in many places at once.

Using the work on objects by Daniel Miller and Greg Noble as a cue (see Part One - Place) and my interest in creating a non-linear story experience, it seemed logical to base the physical aspects of *Nostalgia Ti Frega* on the age-old custom of storing old photos in shoe boxes. Greg Dyer, the author and novelist, describes the act of looking through a box of photos as one of simultaneity and random juxtaposition:

> You rummage in the box. You pick a photograph and then another one and the way they are combined makes you view each of them in a different way.[127]

To augment this random potential I decided to use five boxes, each of which would contain a set of unique photographs. Five seemed like a limit that would both accommodate a collection of photos large enough to provide an ongoing experience, but small enough to be practicably achievable given the

[127] Dyer, G. 2006, *The Ongoing Moment*, Vintage, New York, p38.

time and resource restraints I was working within. I decided the boxes would be hand-crafted from wood, something solid, permanent, tactile and special. Their rich materiality would juxtapose against the digital iPad app, and would provide not just a contrast in terms of presentation, but also in terms of physical presence. They would be nice objects to touch. The wood would also give the boxes a particular cultural status making them important, and something to display, rather than a box to put in a cupboard for storage. They would be designed to be used objects, living in separate family homes, and not as objects that would sit in art galleries. The photographs the boxes contained would also play into this, being printed in a vintage style with borders around the frame. Any design markings on the box would take their cues from the same visual references that were to inform the design of the iPad app. There would be space inside the boxes for photographs to be added, and this space would also allow for the mixing and random retrieval of images. Overall the boxes would be comfortable, rhizomatic objects through which the viewer would be able to form their own closures and openings.

If the mode of analog presentation was something as simple as a hand-crafted wooden box, what form then should the digital container take?

Given the rapid uptake of touch-tablet devices, and the consequent increase in the use of touch interfaces, it seemed timely that the digital component of the project be developed for these technologies. Also, as touch-based interfaces are currently “bleeding edge” technology, using them would make the research current and relevant. But would this platform serve the story? Would the story experience be enhanced or diminished?

Of all the human senses, touch is the one most closely linked to intimacy. In Sherry Turkle's collection of essays *Evocative Objects*, she quotes Susan Yee, an architecture scholar from MIT, recounting her experience when visiting the Le Corbusier archive:

> One day, I asked to see the overall plan drawing for the unbuilt design. I was escorted to a special room where Le Corbusier's largest drawings were viewed. [The drawing] was so large it spilled over the edge of the table... I expected to be given gloves, but I was not... I didn't know if I could or should touch it. And then the curator touched it, so I went ahead and touched it too with my bare hands... There were marks on it, smudges, fingerprints, the marks of other hands, and now I added mine. I felt close to Le Corbusier as I walked around and around the drawing, looking at the parts I wanted to replicate, to bring home with me, touching the drawing as I walked.[128]

Yee's experience demonstrates the power that touch has to create intimacy and draw a connection to something that feels like 'home' (as Yee wants to take the drawing home).

In her work on transnational Italian families, Baldassar argues that the notion

[128] Turkle, S. 2008, *Evocative Objects* MIT press, Cambridge, p33.

of "staying in touch" is a way geographically separated families maintain not just communication but a also sense of emotional connection.[129] Her studies also reveal that new technologies are supplementing older communication forms and becoming part of the available means families use to keep in touch with kin overseas.[130]

Anne Cranny-Francis, a cultural theorist who has written extensively on the politics of human sensory engagement, examines touch and how it operates as a articulator of values, assumptions, and beliefs. In *Touching Technologies: The Biopolitics of Emerging Technologies* (2013), she writes that “touch is not just a physical gesture, a functional act that makes something happen; it is also a way of meaning and of being.” Touch is more than a metaphor for how we feel emotionally, it is also an agent in our mental process:

> Even if we simply stand still, we touch the air around us, and are touched by it – and so perceive it as wet or dry, hot or cold. That bodily touch (or touching of our bodies) is the basis of our sense perceptions; it grounds the information we gather – perceptually and conceptually – about the world. Further it is an embodied deconstruction of the Cartesian dictum, 'I think, therefore I am', which is usually read as signifying the primacy of rationality and of mind (over body). If we understand embodiment (and not ‘the mind’) as the fundamental

[129] Baldassar, L. Op. Cit. *p387*
[130] Ibid. p400

> condition of being and of knowledge formation, then we must reverse the terms: I am embodied, therefore I think. And, if that embodiment is understood as connectedness that is experienced as touch, then we might reword this as: I am embodied, therefore I touch, therefore I think.[131]

Cranny-Francis looks at how this touch-enabled sense of embodied connection and identity is further played out through objects and technology. She cites the example of Richard Attenborough, who, in describing a 1.8 million year-old chopping tool details its weight, its feel to hold, the fact that it fits his hand so nicely. Attenborough is 'excited' as he describes what the tool feels like and Cranny-Francis identifies with this emotion:

> We know this same excitement, where the touch of an object from the past - precious to our culture or society, or even to our own family history - links or connects us to that past... [the] tool that fit the hand of a human from approximately 1.8 million years ago also fits [Attenborough's] hand; as a result he feels a (tactile) connection with that earlier human...[132]

It's not unlike Susan Yee's connection to Le Corbusier as she handles his drawings.

[131] Cranny-Francis, A. 2013, *Technology and Culture: the biopolitics of emerging technologies*, Palgrave, London, p 16.
[132] Ibid. p 27.

Extrapolating her observations to machines and more modern technology, Cranny-Francis notes that while sensory engagement with technology has not always been positive (the factory floors of the industrial revolution, for example), connecting to digital technology has often given human users access to previously inaccessible things, processes or information, and therefore to positive experiences. The incorporation of technologies (particularly mobile touch-interface devices) into daily life is an example of this. There is a sense that this new relationship to technology is an equal one, certainly more equal than during the industrial revolution, and while Cranny-Francis ultimately seeks to question this notion of equality, it is clear that touch functions "to erase boundaries between the individual and others – whether other people, other species, objects or machines - [and] creates a kind of continuity that challenges the human to connect and engage with others."[133]

As Cranny-Francis has shown, when utilized by artworks touch can deftly create an embodied engagement with the audience, deploying the audience/user's sensory responses around cultural knowledge, memory, social experience and understanding. Meanings and feelings are generated. Connections are felt. Touch is a powerful tool for user engagement.

It follows that incorporating touch into both the physical and digital objects of *Nostalgia Ti Frega* would be a strategy through which to create an intimacy with the user and a more emotionally evocative experience. Deploying the

[133] Ibid. p 34.

digital component of a touch-interface device would leverage cultural, social, and psychological factors that would contribute to the themes the project would explore, as well as create an opportunity to experiment with new, developing technology. The biggest challenge would be to experiment with and create a new kind of photographic album experience within the constraints a touch tablet device afforded. Would it be possible to create something on a digital platform that both aped/mirrored/echoed the old-world *experience* of looking through a box of old photos, but that was not simply a graphical (skeuomorphic) replication of that experience (eg. a graphical duplication of a box, with photo images arranged in a random manner within this box, and the user given the ability to move them around this space and browse through them)?

Would it be possible to create something that worked with its own style and language, with a rhizomatic structure, that engendered an experience of discovery, serendipity, and play, all within the confines of a touch-interface tablet application?

PART THREE - PROCESS

> *"Memory is not an instrument for surveying the past but its theatre. It is the medium of past experience, just as the earth is the medium in which dead cities lie buried. He who seeks to approach his own buried past must conduct himself like a man digging."*
> *Walter Benjamin* [134]

> *"If it sounds inefficient, I've described it properly. I cannot imagine how I could have been less efficient. But inefficiency is the point. You can use a map and drive to a destination. Or you can follow roads - trusting yourself, trusting the car, and trusting the logic of the pavement - and end up where you couldn't have realised you wanted to be until you got there. My trip to Trachimbrod would have been better served by some smart plans. But I wouldn't have written a novel. Writing hates such intelligent preparation."*
> *Jonathon Safran Foer* [135]

My creative growth, professionally speaking, took place within the formalised conventions of film and television production. In these spaces the creative process goes something like this:

[134] Benjamin, W. 2006, *Berlin Childhood Around 1900*, Belknap Press, Boston.

[135] Jonathan Safran Foer interviewed by John Mullan in *The Guardian*, <http://www.guardian.co.uk/books/2010/mar/20/jonathan-safran-foer-everything-illuminated> Viewed 20 March 2010.

An idea is followed by a synopsis and a treatment and, if you're lucky, you'll be able to research and develop the idea, fleshing out characters, story arcs, plot points. You'll then write an outline, followed by scene breakdowns, followed by a first draft. Followed by many, many more. Long before anything is actually committed to film or tape.

While you're doing this you'll negotiate your way through a myriad of rules, from a myriad of experts, which integrate, agree, contradict, or confound, but which all steer toward a correctly formatted document whose story adheres to a generic convention.

Then, if you're even luckier, you'll commence production, with mood boards and storyboards and shot lists and rehearsals filled with character analysis, and intentions, and call sheets and daily schedules and wet weather contingencies and mad panics to catch the light, or avoid overtime, and 10-hour minimum turnarounds and rushes screenings and everything timed to the last detail, priced to the nearest cent, and cost-reported daily.

And then post-production will start. Assembly edits, rough cuts, special effects rendering, final cuts, sound editing, sound mixing, colour grading, release printing.

One step following another, and then another, all dependant on the previous, all hurtling toward the final released product.

When I transitioned into digital media production, there were slightly different development and production processes, but they generally followed a similar, formalised, path:

Business requirement gathering, business requirement documentation, functional requirement gathering, functional specification documentation, graphic and user interface designing, back end development, front end development, testing, deployment.

Like film production, one step following another, one contingent on the last, all moving toward a final, releasable product.

Commonly termed a “waterfall” methodology in software development circles, and simply referred to as “filmmaking” in film production circles, this practice, born from a need, essentially, to track and therefore control costs, became a way my creative brain was wired.

It didn’t matter that, when writing a script the process was never *actually* linear. That new ideas about character background would emerge long after their breakdown had been completed, that feasible plot points or story events would present during research and become part of the story well before they were due to, that scenes would require re-shoots, or additional scenes would need to be added, or dropped, or six helicopters for five days would turn to one helicopter for half an hour, that it was always, like the act of creation, or even like death, a random, unpredictable, back and forth.

Commercial web production was no different. Specifications would be set, and then changed, approvals would come late, scope would creep, designs would need amending once user testing was finally done, hardware limitations would be considered way too late in the day.

However systematic on paper these processes seemed, the experience of actually living the process was anything but. And I guess if there wasn't so much money on the line, and so many heads above bearing pressure to make sure none was lost, the process might have felt more fun and creative. But it didn't.

Since *Nostalgia Ti Frega* had no pressure to exist as a commercial product, and since there was a generous amount of time within which to complete the project, I was eager to exploit this opportunity by exploring a less rigid, less predetermined creative path.

Embarking

As a researcher for this project I had a lot of prior knowledge of the people I wanted to photograph. In a sense, as a member of the community myself, I was already embedded in what I was researching. What this meant was that while all the interviews I conducted were ethically approved[136] and would begin with a set of standardised questions[137], as I knew many of the people I

[136] Approval was gained from the UTS Research Ethics Team in May of 2011 (HREC 2011-165A)

[137] See Appendix A

spoke to it would be possible to adopt an informal conversational tone. Further, these interviews weren't conducted with the intention of using material from them as primary data, as might be done in a traditional PhD, but with the intention, from an artistic point-of-view, of identifying what stories would be important, what media would be relevant, and whether the project was even going to be worthwhile to the community that would be its audience.[138]

I was also aware that, given the academic nature of the project, my process would also be well served by an Autoethnographic approach. Given the partly autobiographical, partly ethnographic nature of the material, this methodology would serve me well in helping to analyse my personal experience in order to understand a broader cultural experience.[139]

The interviews were hybrid, in that they involved both conversations and photographic sessions. I interviewed over 50 people in both Sicily and Australia. At the time I wasn't sure what I would do with either the interview material or the photographic material, but because I was able to spend time with the subjects and share their home spaces, I could not only branch off from my initial set of questions but was also able to collect a lot of information about how photographs were being used, shared, displayed, and stored, and consequently what they meant to the community in general. I was able to observe in what instances they were used for fun, for reflection, for

[138] The interviews may, however, form part of a future project.

[139] Ellis, C. Adams, T.E, & Bochner, A. P, 2011, "Autoethnography: An Overview", *Forum: Qualitative Research*, Vol 12, No 1.

commemoration, the kinds of containers they were held in, the different ways and different places they were displayed. And I was also able to observe how digital devices (if any) were present in the household, how they were being used, and whether older members of the community were able to engage with touch screen devices (they were).

I began by looking at how photographs were used - both offline/physical photos, and online/digital photos. In domestic settings physical photos were everywhere. On walls, in albums, in frames on coffee tables, stored in boxes. They were a way of presenting, remembering, and sharing. While older generations possessed more examples of printed photos, younger generations still took part in the tradition of framing, displaying, commemorating, photographs of family members. In digital contexts Facebook was a popular communication medium for 20 - 50 year olds. It was a way relatives in distant countries were kept updated of important life events, like birthdays and marriages. Looking at other peoples photos was enjoyed as much (if not more) than uploading photos (the upload process was often regarded as somewhat annoying and time consuming). The ability to comment on photos was important, and was greatly enjoyed, as was the use of a ‘slideshow’ or fullscreen mode. Images that were new, whether analog or digital, were always eagerly consumed. There was a sense of discovery and play, a delight in recognising something or someone familiar. But there was also a comfort in looking at previously viewed photos, recalling memories and stories and shared histories. Digital photos tended to be scrolled through more quickly, analog photos pondered over more. Having the option to switch between languages was important as was allowing a device to have more than one user: touch tablet devices were heavily shared amongst family members - unlike mobile phones - and older family members were always guided by younger ones. Whether analog or digital, photos were heavily shared across generations, and older family members were equally as interested in viewing digital photos on touch-based digital devices, as they were to look at physical photographs. A digital image was no less interesting to them.

Photographically speaking, I wanted to work within what I understood was a family-chronicle tradition, to create a broad set of images that would be a record of a time, a place, a people - presences felt, lives lived, and things that would attest to my own self. I would work digitally, for convenience, but wouldn't engage in much digital alteration (nothing more than colour-balancing, dodging and burning, exposure adjustments etc, all standard analog darkroom techniques). I would keep the scope of what I photographed as broad as possible, covering people, objects, and places that had even a vague connection to the village, and remain open to what presented. I would shoot first, edit later, and hopefully gather a collection of images that could act as a type of topography, that when viewed in entirety would give a sense of the space I was exploring. The photographs would somehow make 'real' what was ephemeral and hard to describe. They would, in the words of Sontag, capture, or "imprison", a reality that was actually impossible to possess:

> Photographs are a way of imprisoning reality...One can't possess reality, one can possess images--one can't possess the present but one can possess the past.[140]

While I understood my process was ultimately an act of pinning down, of making a sense of something fixed and real, I also wanted the bare mechanics of it to be ones of discovery. I wanted to call up the experience of looking through an old box or album of photographs, to make the process an act of discovery, play, and recollection, rather than a set path pre-prescribed. Not direct or linear, but something more like an ebb and flow, something emergent. As I wanted the final work itself to somehow embody the qualities of reflective nostalgia and rhizomatic non-linearity (see Part I - Place & Part II - Platform) I felt this outcome would be enhanced if the process I employed while taking the photos was similar. Further, in making the process open and unknown, it would be more personally interesting, more like an unfolding narrative: a journey to somewhere undiscovered, not to a place I already knew. As the project was, in part, about self-discovery (which is always arguably a journey into the unknown) this kind of rambling process seemed appropriate to the subject matter. Photographs would appear and present themselves as a way forward, as what photo critic Geoff Dyer describes as an “ongoing moment”[141], or as renowned Sicilian photographer Ferdinando Scianna calls an uncontrolled discovery:

[140] Susan S. 2005, “On photography”, *Communication in History: Technology, Culture, Society* (5th Edition) Part V • *Image Technologies and the Emergence of Mass Society*, David Crowley (Author), Paul Heyer (Author) p174 - 178

[141] Dyer, G. 2007, *The Ongoing Moment*, Vintage, London.

> Even though you have the illusion of controlling things, in the end it’s another kind of accident - fate. You control it but you don’t control it. Finally, even if you prepared the shoot, the picture is still something you find.[142]

A Project Methodology

As I planned to produce a digital product as part of the work, I was also interested in seeing the degree to which an instinctive, step-by-step, exploratory approach resonated with some of the newer, anti-waterfall, agile methodologies of software development that were formulated as a response/solution to the inefficiencies and failures of waterfall’s rigid process.

Agile development methodologies espouse a small, simple, iterative approach to product development, one that includes input from the users upfront, favours the initial development of small feature sets, and prefers the product to evolve in an emergent manner (rather than be conceived in full before development even starts).[143] It also understands that the nature of the digital paradigm, where hardware technologies and software platforms are constantly changing, necessitates products that similarly change too.

Initially developed out of a frustration with the inefficiencies and failures of

[142] Boot, C. Ed. 2004, *Magnum: Stories* Phaidon Press Ltd, London. p 419

[143] see Beck, Kent; et al. 2001, "Manifesto for Agile Software Development". Agile Alliance <http://agilemanifesto.org/> Retrieved 14 June 2010.

standard, cumbersome, “waterfall” production methodologies[144], where weighty product specifications would describe the product in full before any development actually commenced, and where users would not test the product until just before release, Agile has become an established (almost textbook) approach in the area of digital product development. Despite its seemingly radical “maximising the amount of work not done” mantra, Agile is now an important aspect of how many of the software products we use today are created. Further, many large corporations, like GE and British Airways, have adopted Agile processes to increase productivity.[145]

Having started my creative practice in the area of film and television, where the production process is deeply formalised, with correlations to a “waterfall” process easily evident (outlined above as script development, followed by pre-production, followed by production, followed by post-production, followed by release), and where, in digital media, I’d been part of projects using waterfall methodologies, I was curious to explore the degree to which an emerging set of ideas in the area of software development resonated and/or synthesised with the intuitive, collaborative creative approach I wanted to adopt.

One text that proved particularly useful was Roman Pilcher’s *Agile Product Management with Scrum.*[146]

[144] see Royce, W. 1970, "Managing the Development of Large Software Systems", *Proceedings of IEEE WESCON* 26 (August): 1–9

[145] King, R. 2012, “GE Becomes More Agile” published in *The Wall Street Journal*, viewed online July 21, 2013 <http://blogs.wsj.com/cio/2012/05/30/ge-becomes-more-agile/>

[146] Pilcher, R. 2010, *Agile Product Management with Scrum: Creating Products that Customers Love*, Addison-Wesley, London

Pilcher, a leading Scrum and agile product management expert, outlines the new role of “Product Owner”, one distinct from the Product Manager or Project Manager roles that are part of waterfall methodologies. Pilcher proffers the Product Owner role as one specific to agile development methodologies, explaining the diverse skill-set a product owner must have, including fundamental project management skills, knowledge of customers/users, understanding of the marketplace and stakeholder interests, a fluency in UI/UX principles, etc.

Using examples from existing successful software development projects Pilcher demonstrates the crucial role a product owner takes in a project’s success, outlining the way in which product owners are pivotal in a range of agile practices - from the management of emergent requirements, to the formulation of a minimal marketable product, to gathering early customer feedback, to facilitating close team collaboration.

The Agile focus of developing a product slowly, simply, and iteratively, always seemed like a potentially more “creative” experience than developing something using a waterfall methodology. Agile’s minimalist structure, open approach, and collaborative focus gives it a much more spontaneous, dynamic, emergent feel and on this basis alone it felt relevant to my own process. But its positioning of user needs at the centre of its process was interesting too. For although I was producing an explicitly creative and artistic work, one without a commercial aim, and one I envisioned, in part, as a personal journey of sorts, the end product did have a specific audience in mind, namely that of the community that formed the “citizens” of the village. In my mind, the only real external measure of success was whether this audience would enjoy the work and engage with it in an ongoing and meaningful way.

Further, Pilcher’s concept of the “Product Owner” was not unlike the role of a director on a feature film, or a creative producer in television, both of which I’d played professionally. I thought it would be useful to compare what was understood as a “creative producer” in traditional media production with what pioneering software development methodologies understood as the new role of “concept owner”.

The chapter of particular interest in Pilcher’s book, “Envisioning the Product”, outlines a process for embarking on a project. It begins with a set of questions for defining the product vision: Who is the product for? Which needs will it address? What value will it add? What attributes must the product have to meet user needs? What will it look like? What will it do? What will it excel at? How does it compare to existing products? How feasible is the product? Although I may not have asked myself the exact same questions, and had additional questions in my mind when sketching out how the project would manifest, my answers were sufficient to address any of Pilcher’s questions.

The product (project) was for the community and descendants of Salaparuta.

It would be a photographic ‘artwork’ or ‘album’ that would give this community a locus around which to consume and share stories about the village today and the people from it. It would be a bespoke application, designed specifically for this community, and exist outside of existing commercial ‘story-sharing’ platforms. I would also have a collection of photos unavailable anywhere else. Its visual appearance would also be individually crafted according to culturally relevant criteria. The visual appearance would be designed to contribute to the overall story experience. The product would be simple and intuitive to use, enabling a cross-section of users with varying degrees of digital experience to engage with the digital aspects of the project, and would also provide a more familiar non-digital experience as well. It would deploy cutting-edge technology, be a first in terms of platform, act as a new historical record of a specific community, create a photo-consuming experience not previously offered within a digital context and experiment with non-linear story. Finally, it would be developed within the resource and time constraints presented.

Obviously, before having a firm purchase on what direction the project would take, I spent several months posing questions to myself. What kind of creative project did I want to engage with? What were avenues of possible support? What reading would help develop clarity about the context within which the creative work would be situated? What kind of photographic style should I employ? What equipment would be needed? Should the work be analog, or digital, or both? If the work was to include analog and digital components how should these manifest and interact? What kind of experience would the work offer the user? What might their ‘needs’ be? What level of user-generated content would it accommodate? What kind of graphic elements and design should the work employ? What would its touch points be? Would it include sound? Recorded interviews? Black and white or colour photography? What digital platform would I develop on? What constraints would this platform present? What kind of resources would I need? What team members were necessary? What potential desired collaborators were available?

In order to answer these questions, or at least have a launching point from which to begin to answer them, I embarked on a period of multi-pronged, interdisciplinary research: I conducted a literature review, a technology review, and user/community interviews (mentioned earlier).

Theoretically the notions of nostalgia, emotional objects, touch, and rhizomatics became important (these ideas are outlined in more detail in Parts One and Two of this exegesis). Talking to a cross section of eventual end users, observing their photo-consuming and sharing habits, their interaction and engagement with digital technologies, the stories they told about their shared histories (and the way their photographs were a part of this), their aesthetic likes and dislikes, etc, helped guide decisions about the project content, the platform it might be presented on, and how this platform would function.

The community’s focus on family, for example, the omnipresence of family photographs - in frames, on walls, in shoeboxes, in photo albums - signalled that I should be working within a similar photographic tradition. The continued appreciation of physically printed photos indicated that part of the project should include printed photographs. The way discovery and playful engagement formed a significant part of they way photos were consumed indicated a similar user experience would be appropriate for the project to embody. The ability for very elderly (75+) users to intuitively engage with simple touch-screen interfaces signalled a touch-screen tablet was a suitable digital platform to develop on. The fact that I would be working in several different locations and photographing a range of subject matter - from portraits, to landscapes, to still lives - meant I would need a range of different photographic lenses. The fact I would need to travel to photograph the community meant a small, light camera kit would be preferable. As the photos were to have a particular ‘commemorative’ feel, I wanted to use professional photographic equipment. Software I used in post-processing would help approximate, digitally, an analog film look that would subtly enhance the photographic images. It would be better for any printed photographs to be small, easy to hold, easy to share, personal, intimate, like mementos. Any container that held the images should be individual and handcrafted - to give it significance in and of itself - and an object suited to display (like the embossed albums or ornate frames of other displayed photographs). The digital app should also be handcrafted, designed specially for the community it featured, and not simply ape other photo applications, as its purpose was to engender a sense of unique identity. This wouldn’t be achieved as effectively with an out-of-the box, generic app, or an off-the-shelf physical photo container.

Also a factor in defining the project were the general resource constraints it would need to be produced within. These entailed time, personnel, and financial factors, and would impact what Pilcher characterises as the project's feasibility.

While I would be the creative director of the project, conduct the principal photography, and manage the project's production, I would need to engage collaborators: a product manufacturer who could make the containers for the physical photos, a graphic designer and software developer for the digital component, a photography lab to print the photographs, translators when, and if, necessary.

I knew, due to my experience producing and project managing commercial digital projects, the most resource intensive aspect of production would be developing the digital component. The only viable touch tablet platform to develop on, at the time (2011) was the iPad. Its screen size, coupled with the seamless performance of its operating system, made it an obvious choice. While tablets running the android operating system have improved markedly in the previous three years, when I commenced my research this was not the case (and it is arguable that even today android-based tablets are not quite up to the performance and interaction design standards of iPads and the iOS operating system). So if the digital component was to be a custom built application for a touch screen tablet device, what, from a resource point-of-view, were the development considerations?

After initial research it was clear that any ‘cost-effective’ development solution - i.e. using a third-party development framework, with pre-existing software libraries - wouldn’t offer the ability to design unique user interfaces. This was an important conceptual aspect of the project, and wasn’t one I was prepared to relinquish. It also felt important in terms of pushing the boundaries of what a touch-screen interface could offer. I wanted my research to investigate new interaction models, and contribute new knowledge, rather than just re-hash existing conventions. This meant the app would need to be developed in ‘native’ code, most likely using a 2d graphics framework (similar to developing a game, for example), and this would require more development time and a need to collaborate with a software developer who had expertise in this area. These kind of developers were (and still are) comparatively rare and often unavailable. Finding someone suitable would make or break the project.

Graphic design was also a consideration. I wanted to collaborate with someone who could bring a visual interpretation to my overall creative vision, and to the interaction design and information architecture I would design for the iPad application. The designer would need to understand the digital paradigm, with ideally some insight into visual possibilities around the manipulation of digital data. But I was also hoping to engage someone who had experience of physical design too, who understood physical as well as digital touch, and understood that the design of objects plays a crucial role in our identification and attachment to them.

Before I could start talking to potential collaborators, I needed to develop a concise way of speaking about the project, in order to effectively communicate the inspiration behind it, the philosophical and creative framework around it, and the overall vision for the final product/s.

In the creative sphere this sort of communication is part-and-parcel of all collaborative endeavours. Without clear, efficient, transparent communication across the team a project's success is much more difficult. This is, of course, also something recognised as important in other fields where collaboration is involved, and in software development too. In his work on agile product management Pilcher states that the product vision should be shared and unifying, broad and engaging, and provide both an inspirational goal and enough flexibility for individual creativity. Finally, it should be brief and concise, containing only information critical to the success of the product. [147]

Apart from providing a broad creative direction - an overall sense of what I thought would be appropriate visual, cultural, and theoretical references - I was keen to keep things as open as possible to begin with. As stated above, I wanted to employ an emergent creative process, something in contrast to the constricting methods I'd been part of before. I wanted my own process to be one of discovery, but I also wanted each collaborator to be involved in the

[147] Pilcher, R. *Op.Cit.* p 26-27

process, contributing ideas, providing opinion and criticism, and having a stake in the finished product.

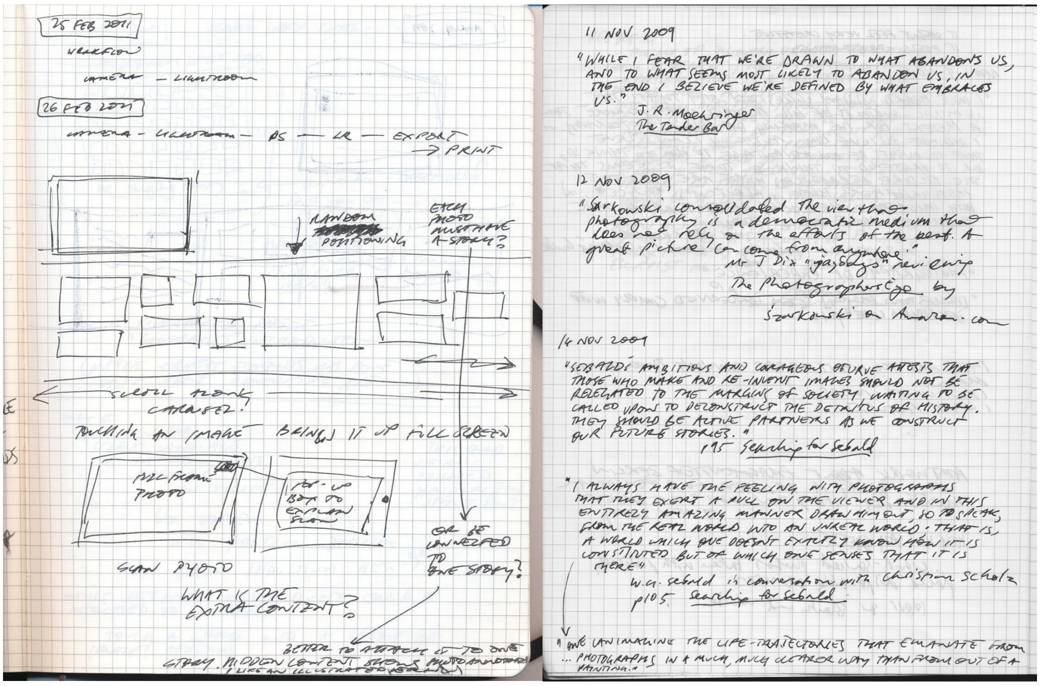
As I commenced research on the project while living in Turkey, and was unlikely to find all suitable collaborators there, I knew it would be necessary to set up a system of distance communication. Luckily, the project's core team would be very small - myself, a designer, a software designer. Others would fill specific gaps when needed - eg model makers or translators - and would probably only need to communicate directly with me (rather than with other team members). So a simple system reliant on free services, such as Skype (IP Telephony), Dropbox (Cloud storage and file sharing), BitBucket (Cloud-hosted git repository management and issue tracking), Pinterest (visual and graphic ideas and inspiration), and, of course, e-mail, would be sufficient.

Distance collaboration had become an increasing characteristic of my professional work. During my time working as a digital producer, processes were often characterised by distance working. From large corporate projects, involving different parties based in different countries, to smaller projects where freelance agencies or individuals were engaged to complete aspects of production, distance working was often a necessity.[148] I wasn't phased at the prospect. But I knew it would be preferable to collaborate with people I would have the opportunity to meet with in real life from time to time, which

[148] For example, one of the bigger projects I produced involved four different stakeholder companies working across three continents with conference meetings all conducted remotely. The most inconvenient aspect was juggling the time differences, which invariably meant at least one party would need to attend the meeting outside of normal office hours.

meant finding collaborators based in Europe.

Meanwhile I started a journal to track thoughts, ideas, inspiration, progress.[149] I started reading extensively. I travelled to Sicily to conduct some initial research into the village and community. I kept tabs on iPad application development. I started looking for potential collaborators. I got my equipment sorted.



25 FEB 2011

WORKFLOW

CAMERA — LIGHTROOM

26 FEB 2011

CAMERA — LIGHTROOM — PS — LR — EXPORT → PRINT

RANDOM ~~[illegible]~~ POSITIONING
EACH PHOTO MUST HAVE A STORY?
SCROLL ALONG CAROUSEL?
TOUCHING AN IMAGE BRINGS IT UP FULL SCREEN
FULL FRAME PHOTO
POP-UP BOX TO EXPLAIN STORY
SCAN PHOTO
OR BE CONNECTED TO ONE STORY?
WHAT IS THE EXTRA CONTENT?
BETTER TO ATTACH IT TO ONE STORY. HIDDEN CONTENT SHOWS PHOTO AND STORY (LIKE AN ILLUSTRATED BOOK)

11 NOV 2009

"WHILE I FEAR THAT WE'RE DRAWN TO WHAT ABANDONS US, AND TO WHAT SEEMS MOST LIKELY TO ABANDON US, IN THE END I BELIEVE WE'RE DEFINED BY WHAT EMBRACES US."
J. R. Moehringer
The Tender Bar

12 NOV 2009

"Szarkowski consolidated the view that photography is a democratic medium that does not rely on the efforts of the elect. A great picture can come from anywhere."
Mr J Dix "jayBdys" reviewing
The Photographer's Eye by
Szarkowski on Amazon.com

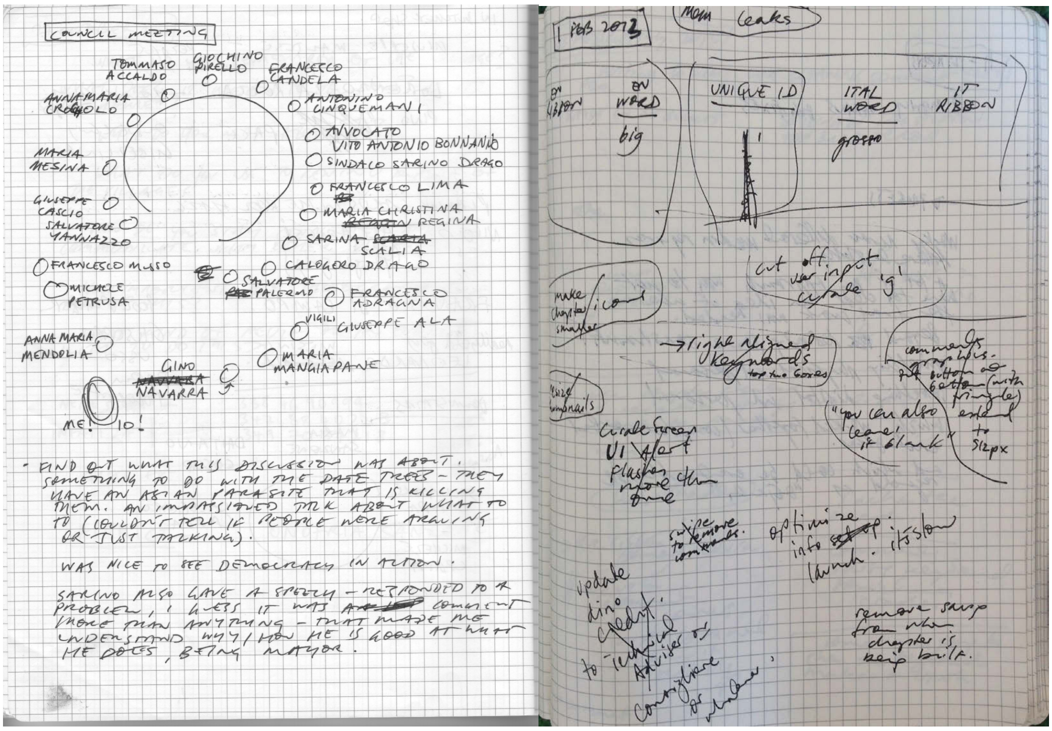
14 NOV 2009

"SEBALD'S AMBITIOUS AND COURAGEOUS OEUVRE ATTESTS THAT THOSE WHO MAKE AND RE-INVENT IMAGES SHOULD NOT BE RELEGATED TO THE MARGINS OF SOCIETY, WAITING TO BE CALLED UPON TO DECONSTRUCT THE DETRITUS OF HISTORY. THEY SHOULD BE ACTIVE PARTNERS AS WE CONSTRUCT OUR FUTURE STORIES."
195 Searching for Sebald

"I ALWAYS HAVE THE FEELING WITH PHOTOGRAPHS THAT THEY EXERT A PULL ON THE VIEWER AND IN THIS ENTIRELY AMAZING MANNER DRAW HIM OUT, SO TO SPEAK, FROM THE REAL WORLD INTO AN UNREAL WORLD · THAT IS, A WORLD WHICH ONE DOESN'T EXACTLY KNOW HOW IT IS CONSTITUTED BUT OF WHICH ONE SENSES THAT IT IS THERE"
W.G. Sebald in conversation with Christian Scholz
p105 Searching for Sebald

"ONE CAN IMAGINE THE LIFE-TRAJECTORIES THAT EMANATE FROM … PHOTOGRAPHS IN A MUCH, MUCH CLEARER WAY THAN FROM OUT OF A PAINTING."

[149] See <http://www.carladrago.com/dca/journal/> for a blog that contains scanned pages of all journal volumes.


COUNCIL MEETING
TOMMASO ACCALDO
GIOCHINO PIRELLO
~~FRANCESCO~~ CANDELA
ANNAMARIA CROCOLO
ANTONINO CINQUEMANI
AVVOCATO VITO ANTONIO BONNANNO
MARIA MESINA
SINDACO SARINO DRAGO
FRANCESCO LIMA
GIUSEPPE CASCIO
MARIA CHRISTINA REGINA
SALVATORE YANNAZZO
SARINA SCALIA
FRANCESCO MUSSO
CALOGERO DRAGO
SALVATORE PALERMO
MICHELE PETRUSA
FRANCESCO ADRAGNA
VIGILI GIUSEPPE ALA
ANNA MARIA MENDOLIA
MARIA MANGIAPANE
GINO ~~NAVVARA~~ NAVARRA
ME! IO!

- FIND OUT WHAT THIS DISCUSSION WAS ABOUT. SOMETHING TO DO WITH THE DATE TREES – THEY HAVE AN ASIAN PARASITE THAT IS KILLING THEM. AN IMPASSIONED TALK ABOUT WHAT TO DO (COULDN'T TELL IF PEOPLE WERE ARGUING OR JUST TALKING).

WAS NICE TO SEE DEMOCRACY IN ACTION.

SARINO ALSO GAVE A SPEECH – RESPONDED TO A PROBLEM, I GUESS IT WAS A COMMENT MORE THAN ANYTHING – THAT MADE ME UNDERSTAND WHY/HOW HE IS GOOD AT WHAT HE DOES, BEING MAYOR.

cut off user input cyrale 'g'

make chapter/icon smaller

→ right aligned keywords top two boxes

resize thumbnails

comments graphics. put button as bottom (with triangle) extend to 512px

"you can also leave it blank"

circle screen UI Alert flashes more than one

swipe to remove comments.

optimize info ~~set up~~: launch. it's slow

update dino credit: to technical Adviser of Conzigliare of [illegible].

remove swap from when chapter is being built.

Figure 3: Scanned pages of my creative journals for *Nostalgia Ti Frega*

By the end of the first year of the doctoral period I had developed a strong vision for the project. This became the basis of my doctoral assessment, and in a more concise and simplified form, was how I spoke about the project when describing it to the developers and designers I approached. I found a London-based graphic designer to work with: Stefanie Posavec, an American print designer who also works, by hand, on data visualisation projects[150]. And I found a Berlin-based antipodean iOS developer to work with: Constantine Karlis, who I had collaborated with before on commercial projects, and whose background in psychology informed his user-oriented approach to software development[151].

[150] More about Stefanie Posavec's work can be seen at <http://www.itsbeenreal.co.uk>.
[151] More about Constantine Karlis' work can e seen at <http://www.constantinekarlis.com/>.

The Minimal Viable Product

The next step was to work out exactly what to make. I decided early that the analog component of the project would be relatively simple to produce. Given the complexity of producing a digital application I wanted to keep things feasible, but I also wanted the physical component to act as a counter to the digital product. Something solid, familiar, grounded and earthy. I'd base the design on a shoebox (a common container for old photographs) but build it with a beautiful wood. I'd add some subtle graphics to the design that would brand it. There would be five boxes, enough to house the large quantity of photos I intended to include, and each box would have small design differences. The lid of the box would feature different triskelion[152] designs, the fonts used on the side of the boxes would be different too, and chosen from fonts made iconic through their use by Italian companies (like Alfa Romeo) or by their use during influential design periods in Italy. The slight differences were a way of reinforcing a sense of hybridity. I was able to find a local model maker who, once provided with CAD drawings, would help build the boxes.

[152] A treskilion design is used as the symbol of Sicily, but is also used by other cultures including Gaul and Celtic. See <http://en.wikipedia.org/wiki/Triskelion> viewed 26 April 2010

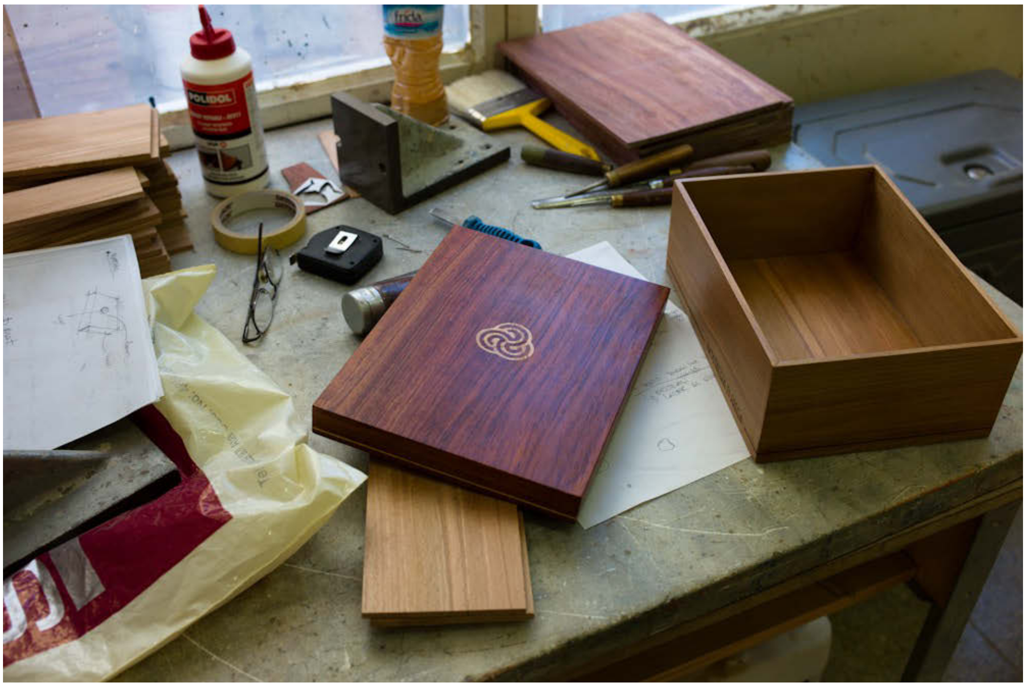


Figure 4: The initial prototype of the wooden boxes.

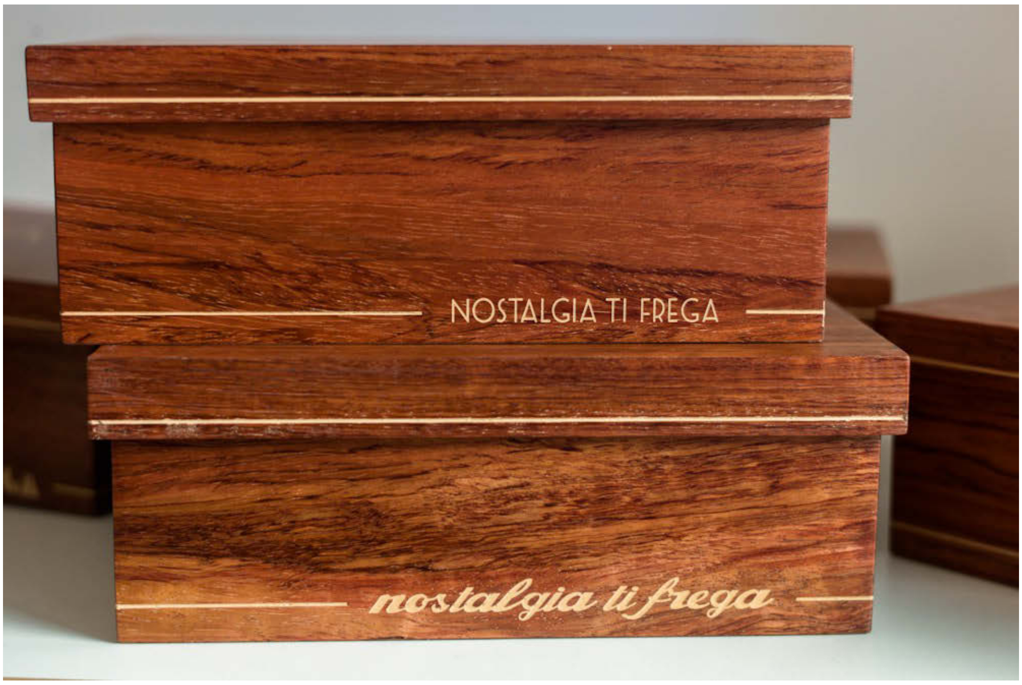


Figure 5: Some finished boxes, showing some of the alternate fonts used.

That left designing the digital product.

One thing Pilcher is keen on is the concept of a Minimal Marketable Product. This is not his concept but one coined by software engineers Mark Denne and Jane Cleland-Huang[153], and popularised by Silicon Valley entrepreneur Eric Reis[154]. It denotes the smallest amount of functionality in a product that can create value for a customer. This product can then be released to the public, and valuable user behaviour can be measured and learned from. Pilcher cites things like the iPod and iPhone, devices that upon first release had a limited feature set (eg it was not possible to cut and paste on the first generation iPhone). Apple focused on the minimum amount of functionality for these products, was able to release the product far quicker than competitors, and were consequently able to develop further functionality as it received feedback from users. This meant it wasn't wasting resources developing functionality or features that would later prove unimportant for users.

From a business perspective the intrinsic value of a minimal viable product is the ability to release it quickly. The valuable user information it returns, and the product’s presence in the marketplace, give the company a competitive advantage over rivals. This kind of concern is not necessarily as important for an art project, especially one with a very limited audience (or ‘market’) in mind. What is of great value however, in an art-practice context, is the way product features are kept to a bare minimum, and incremented iteratively,

[153] Denne, M. and Clelnd-Huang, J. 2003, *Software by Numbers*, Prentice Hall, New Jersey.
[154] Reis, E. 2011, *The Lean Startup*, Crown Business Publishing, New York p76-77

step by step, enabling costs to be kept low and managed. Using this approach was an incredibly useful way to manage the limited resources I had as I was able to add, drop, or adjust features in accordance with their usefulness and/or viability. It meant I was able to make informed and calculated decisions, not spend too much time on things that would later be dropped, and proceed with production fairly painlessly.

Pilcher emphasises the importance of user needs when defining product attributes. So once I had, through my literature review, established a theoretical framework the project would work within, I spent time with some eventual users of the product in order to gather their so-called needs.

Based on all of this research I formulated a plan for the minimum feature set. The first version of the digital application would:

- showcase new photographs that I took, rather than provide functionality for users to upload their own photographs.

- engender a sense of discovery and play for a user viewing the photographs.

- have graphic elements that were in some way generative, in order that the application appear dynamic and ‘alive’, rather than static and fixed.

- allow users the ability to view photos in full screen.

- allow users the ability to leave comments on photos.

- not allow users to upload their own photographs (as this would be too difficult to develop within the resource constraints) but instead allow them some way of sharing their photographic discoveries with other users.

- given that many families shared a single iPad, allow a single device to be sharable across different users, specifically, designing an easy way for the user name to be changed. This would likely mean no user “sign-up/sign-in/sign-out” process.

- be offered in two languages: English and Italian.

- Given the range of ages using the product, and the different language backgrounds, have simple user interfaces that did not rely heavily on text whenever possible. Buttons would be big, and few in number. The app should be easy to install and start using (another +1 for no user sign-up/sign-in process).

- somehow convey a sense of place, or a virtual world, that represented the community it featured, and somehow embody the ‘place’ they were all from.

I decided, once this functionality was organised and grouped, that the application would be broken into different functional parts and that each part would be developed separately. This is a typically Agile approach.

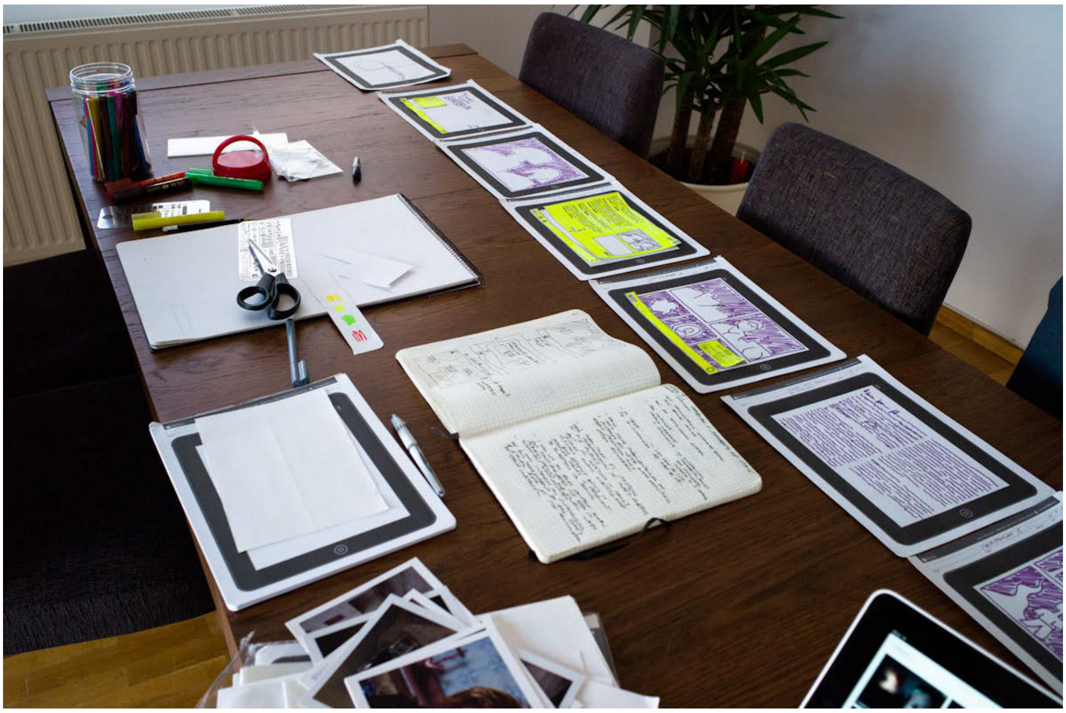
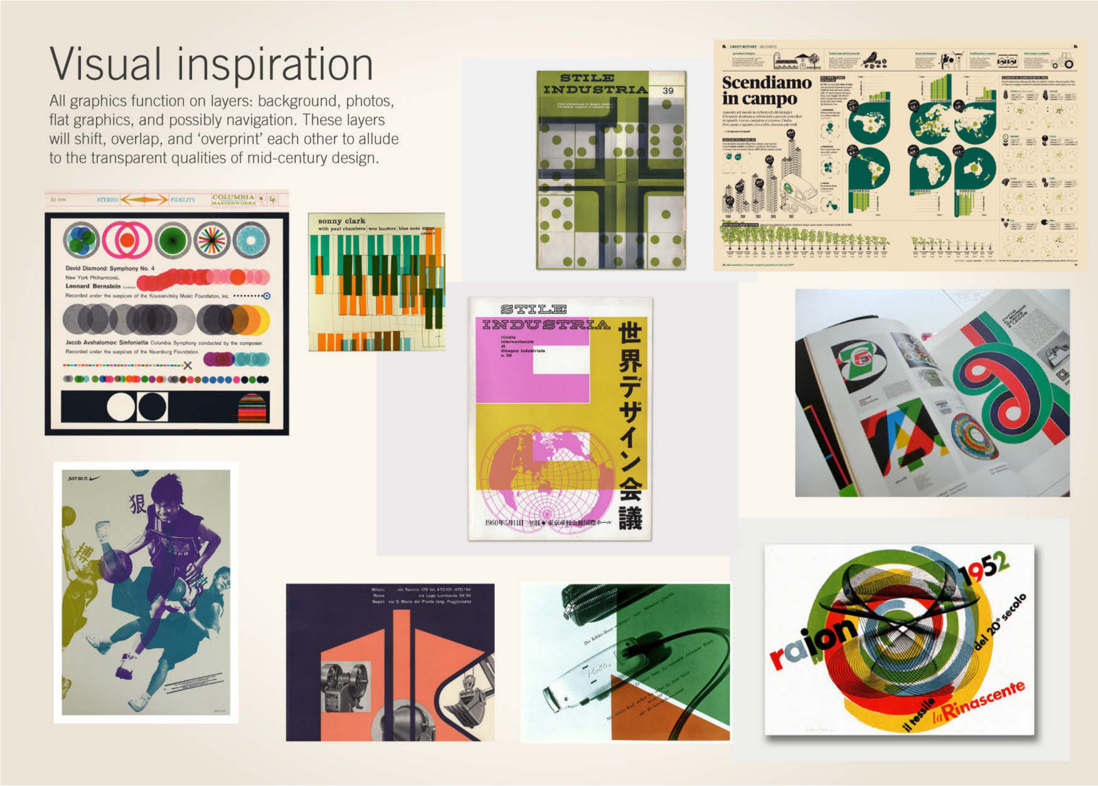


Figure 6: Paper-prototyping in digital application.

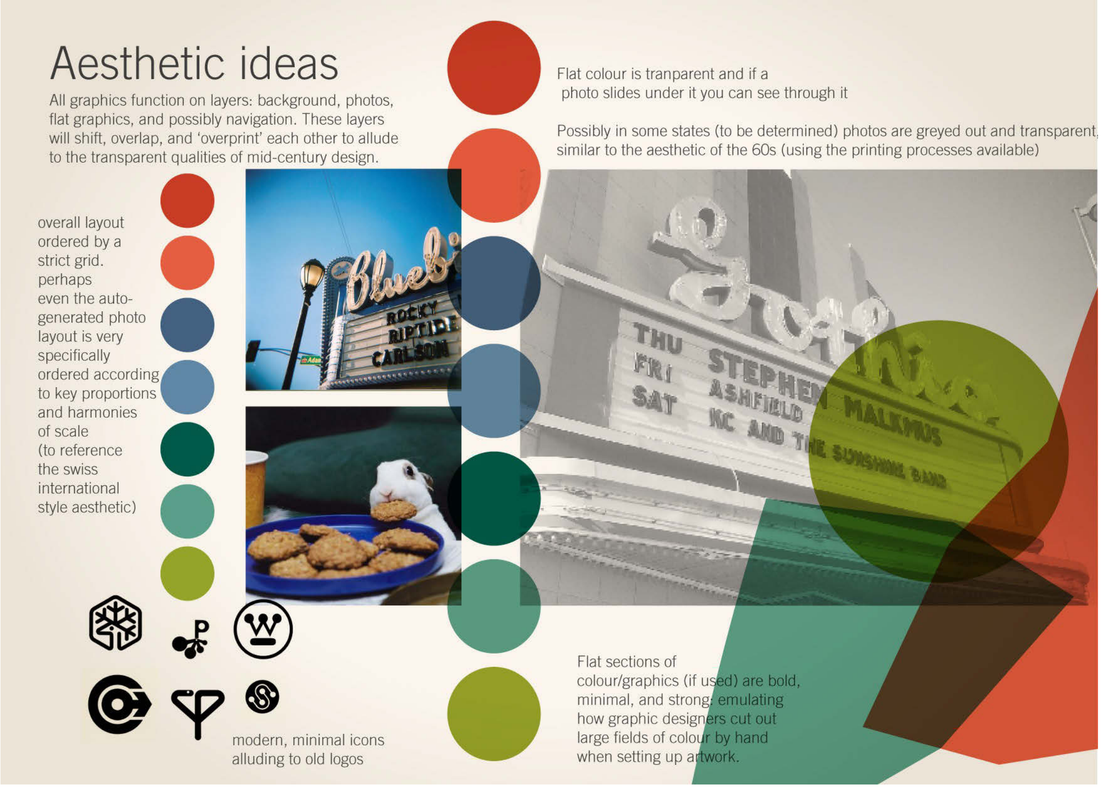
Work began on a style guide/graphic approach, with principal inspiration deriving from modern Italian design of the twentieth century (1950s - 1960s). Apart from being a design style that developed during the post WWII period, which is when many migrants left Italy, and so both recalls this period and even evokes nostalgia toward it, its aesthetic characteristics - clean lines, sans serif fonts, bold graphics, minimal icons, overlayed colours - would (conveniently) be quite adaptable to a digital context. The application would be able to heavily reference the mid-century design style without stumbling into the dubious digital design territory of skeuomorphism.[155]

[155] “Skeupmorphism” is a term used to describe the design practice where a digital object is made to resemble its analog counterpart. Eg the *Notes* application of iOS, which looks like a yellow legal note pad and uses a sketch-like handwritten font. Or the iBooks library bookshelf. This practice has been derided by many commentators as inappropriate but employed by many (notably Apple) as a psychological ploy to make users think an application is easy to use, and also tap into any analog nostalgia they may have. See Andy Gold’s post “Skeuomorphism: The Opiate of the People”, Viewed 20 February, 2013. <http://andymangold.com/skeuomorphism-the-opiate-of-the-people/>



Visual inspiration

All graphics function on layers: background, photos, flat graphics, and possibly navigation. These layers will shift, overlap, and 'overprint' each other to allude to the transparent qualities of mid-century design.



Aesthetic ideas

All graphics function on layers: background, photos, flat graphics, and possibly navigation. These layers will shift, overlap, and 'overprint' each other to allude to the transparent qualities of mid-century design.

Flat colour is tranparent and if a photo slides under it you can see through it

Possibly in some states (to be determined) photos are greyed out and transparent, similar to the aesthetic of the 60s (using the printing processes available)

overall layout ordered by a strict grid. perhaps even the auto-generated photo layout is very specifically ordered according to key proportions and harmonies of scale (to reference the swiss international style aesthetic)

modern, minimal icons alluding to old logos

Flat sections of colour/graphics (if used) are bold, minimal, and strong; emulating how graphic designers cut out large fields of colour by hand when setting up artwork.

Figures 7 & 8: Inspiration and aesthetic ideas for the digital application.

I also started to think about the two key questions that had arisen following the formulation of the basic feature set, namely:

- how would the photographs be discoverable?
- how would the app have a dynamic, emergent feel?

With physical photos stored loosely in (e.g.) shoeboxes there's a sense of chance when you look through them. Even with photos kept in albums, trying to find a specific photo often involves paging forward and paging back. There are often loose photos kept inside covers, or in between pages for later placing. Photos fall out of their lodgings. Photos you thought were in one album turn out not to be and so several albums need to be looked through.

Physical family photographs tend not to be indexed and sorted, tagged and organised, archived in a system that allows easy access. Finding a specific photo takes time, it's an exploration, and something no one seems to mind. The exploration, conversely, is actually pleasurable. It's a moment for recollection and reminiscing. A small ritual. In all the research interviews I conducted, no one seemed particularly perturbed at this.

In a digital context recreating this sense of discovery and exploration is tricky. Most examples of digital albums conform to very rectilinear systems - both visually and in the way they are stored. Slideshows or streams move from start to finish, one image after the other. Photos are stored in albums which are stored in collections which might be stored in another parent folder.

These branching hierarchies don't allow for movement between branches. When browsing one has to go all the way back before one can go forward, into another branch. Searching is about relevance and precision, matching query to string (title, or tag, or description).

How could I change this customary approach in my app?

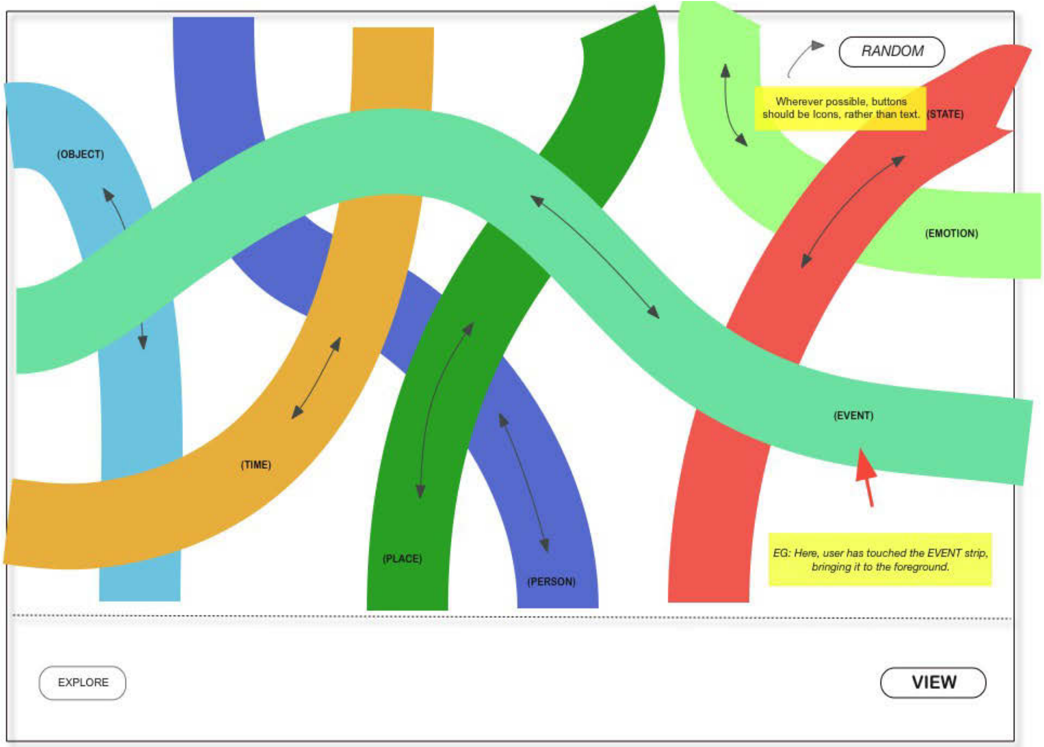
Obviously, some form of random discovery would operate. But I wanted to engage the user in the process, make them an active agent in what ~might~ be discovered. I also wanted to signal to the user the range of photos within the app. If they were unaware of what actually existed to be discovered, they might be less inclined to continue to explore the 1500+ photographs that were in the app.

I decided to tag each photograph with relevant keywords, which is standard photographic archival practice[156]. If those keywords were visible to the user, the user would get a sense of the range of photos the app contained (many keywords = many photos). And if they were then able to select from those keywords, they would be able to 'narrow' the randomness and have a sense of agency over the photos that their keyword selection returned. If they were able to see all the keywords it would increase the chances of them selecting keywords they had not selected before. Further, if these selections were sharable - meaning that other users of the app would be able to view them,

[156] Anderson, R. and Russotti, P. 2009, *Digital Photography Best Practices and Workflow Handbook*, Focal Press, London.

and identify who had ‘found’ them - the user would then become a curator, and be more than just a ‘user’. This feature would also give the app a dynamic, emergent feel. Each new curation would be part of the app’s evolution.

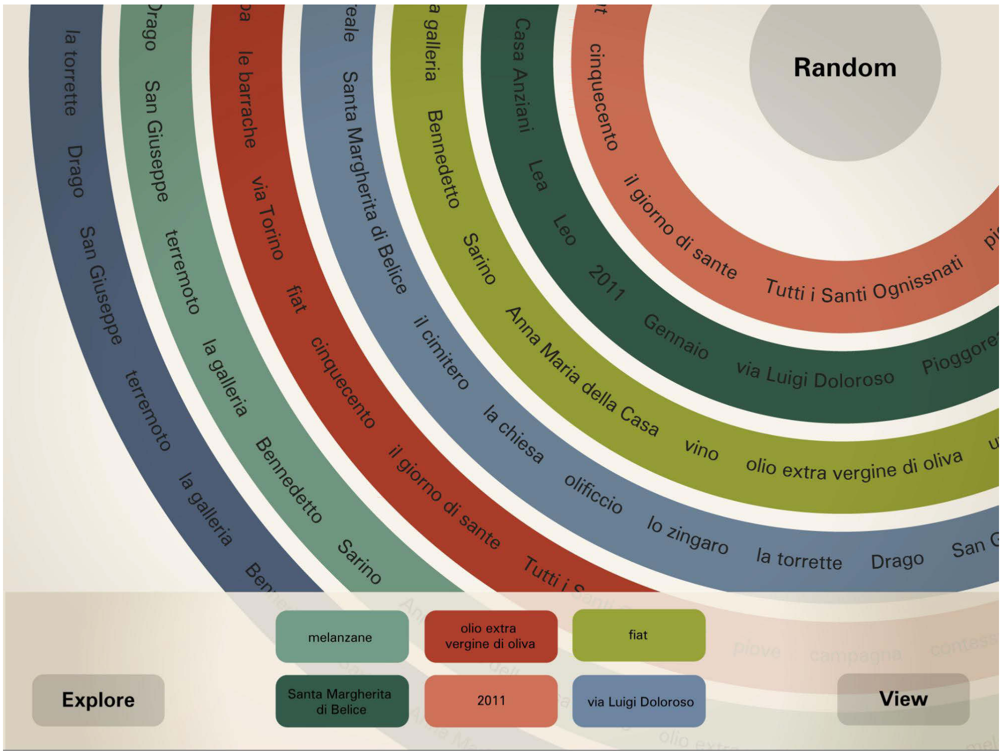
In order to encourage engagement and playfulness this curation process would need a special visual interface. I didn’t want mere alphabetical lists of words. I wanted something special, that would be fun to use, and that would be organised according to its own logic.

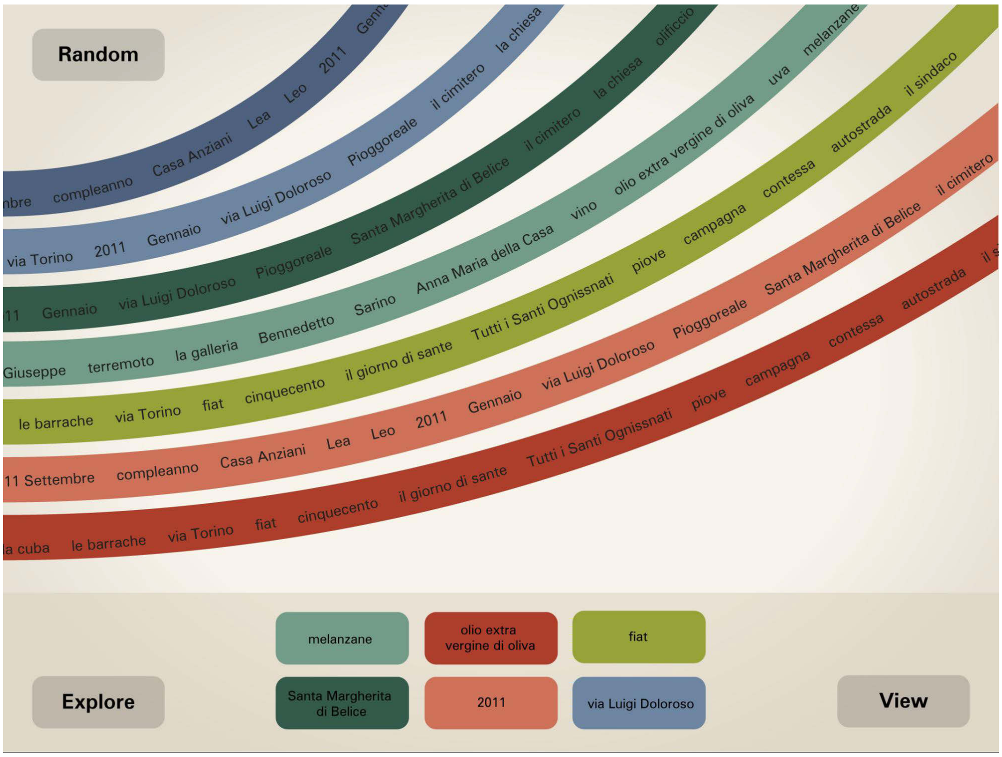


Wherever possible, buttons should be Icons, rather than text.

EG: Here, user has touched the EVENT strip, bringing it to the foreground.

Figure 9: The initial Curate Screen wireframe.





Figures 10 & 11: Posible final approaches for the Curate Screen.

It was important that the user knew that there were many many keywords to select from. I didn’t want to have all the keywords on the screen at once (which would have been an easy way to demonstrate their number to a user). I could categorise the keywords and utilise something like drop down menus, but this felt clunky and rather too tree-like in structure. An idea that did appeal was to categorise the keywords into groups, and place them, according to group, on scrollable ribbons. This would allow the user to understand, on a single screen, without drop down menus, the number of keywords available. The scrolling ribbons would provide a playful and aesthetically pleasing interface. This concept would also provide an opportunity to create a totally original interface, with a unique graphic style. Users would be able to select keywords from these ribbons, then touch a button to view a random selection of photographs tagged with these keywords.

Once the graphic design approach was finalised, this was tested with users as a paper prototype and development work began on the screen’s implementation. This screen became known as the Curate screen.

The technical challenges this approach posed were mainly related to the shape of the ribbons. It was clear this screen would need to be programmed using iOS Cocos2D graphical framework, rather than using the pre-existing UI Libraries that are part of the iOS SDK. The curves of the ribbons, and the fact that they all had their own unique arc, meant that complex algorithms were required to plot each one. Other issues, such as number of keywords, inconsistencies in physical length of keywords after translation into Italian, and repeated keywords (different words in English, like niece, grandchild, and nephew, for example, were just the single word *nipote* when translated into Italian. This needed to be dealt with programmatically so that the application wasn't confused).

Development on the front end of the screen was completed first. Once this was more or less functional the next step was to solve the problem of how the photo selections each user made would be discoverable by other users. What kind of interaction and graphic design would help communicate that there were many other users of the app, creating many other photo selections (or Chapters, as they became known).

There were different ways this could be approached. Chapters could be presented in a list, or as tables, but this felt too linear and pedestrian. But what if each chapter was represented by a graphical icon, and this icon was positioned in a graphical space (not in a list or table). A user could then select an icon to launch and view that chapter. This seemed more in keeping with the fluid, non-linear, organic experience the design was attempting to achieve. Also, the app was part of a project that explored notions of place (physical and virtual), but this operated on a sub-textual level. Ostensibly the app would seem like a new and experimental family photo album, but having an open graphical space to place chapter icons would be an opportunity to help reinforce the central philosophic themes of hybridity, reflective nostalgia, and comfortableness.

Given that the places the photographic content explored were geographically dispersed, or in some cases didn't even exist anymore, it seemed appropriate to create a fictional space, that referenced geographic map symbolism, allowing this to be the conceptual world all the curated chapters lived in. Using a portion of an actual contour map of rural Sicily, a fictional geographic map was created, complete with fictional Italian place names that had a symbolic and poetic relevance and reflected aspects of the migrant experience.[157] This would be where each chapter symbol would be displayed, and act as the overall landing page for the app - the screen where all users arrived upon launch. This screen was to be called the Explore screen, as it would allow users to explore other photographic chapters.

So, if the chapters were represented iconographically, what would the icon look like?

[157] Eg. *Rifugo Vuoto* (Empty Refuge), *La Speranza di San Giuseppe* (St Joseph's Hope - St Joseph is the village's patron saint), *Porta delle Origine* (The Doorway of Beginnings), *Baratro della Nostalgia* (The Chasm of Nostalgia)

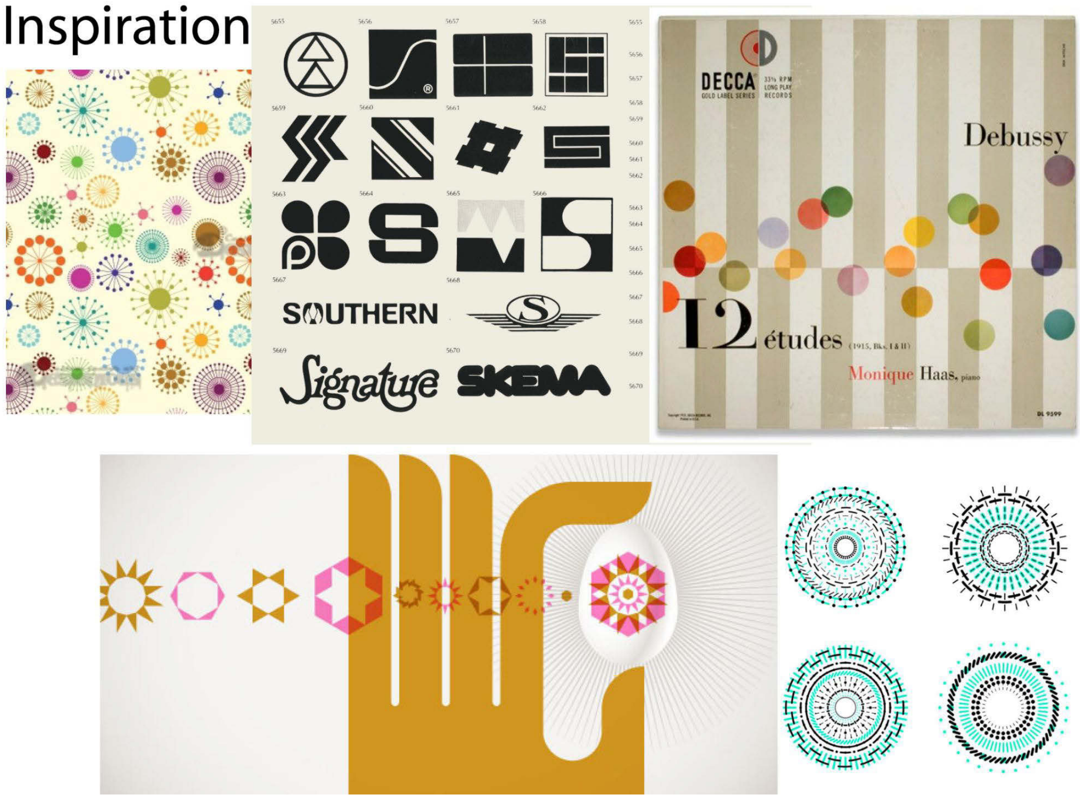


Figure 12: Chapter Icon Inspiration.

The other main question the interaction design needed to address was how the app would have a dynamic, emergent feel?

One way to do this was to have the graphic elements of each chapter icon change according to the keywords it contained and the order of their selection. Each Curate screen ribbon would have its own base graphic shape. These graphic shapes would be individual, but complementary.

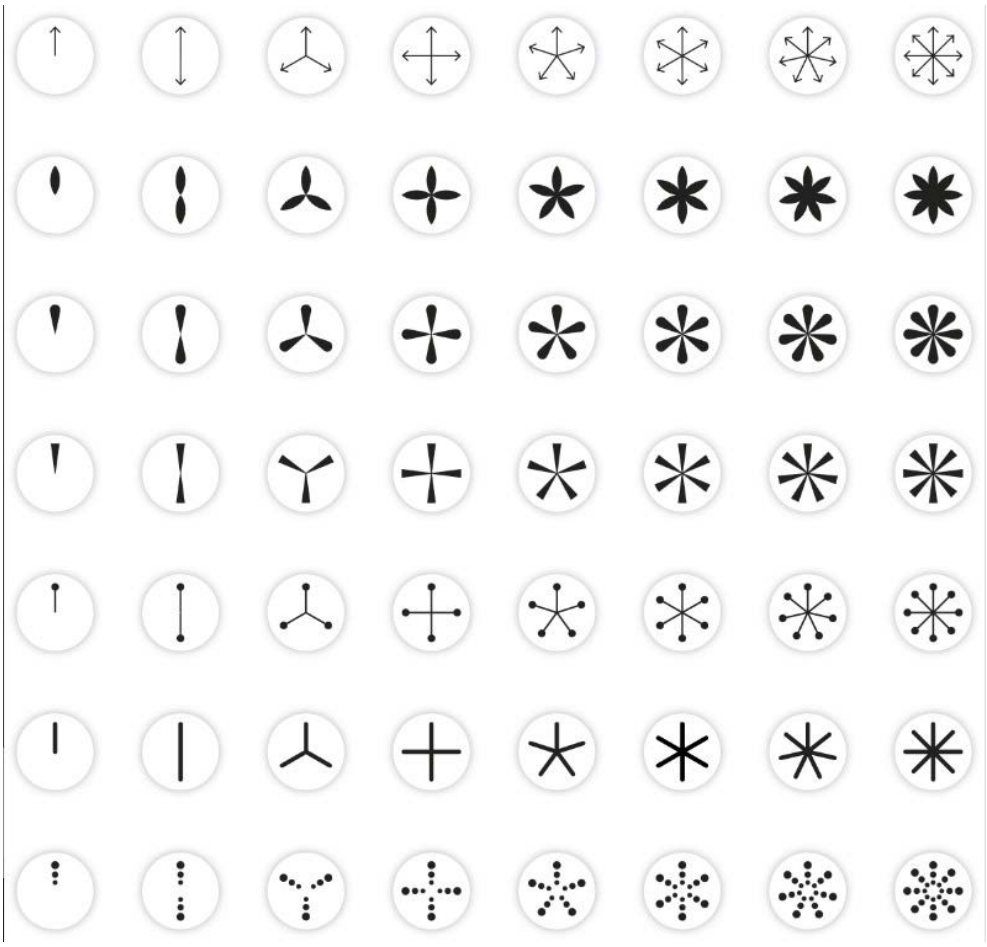


Figure 13: Final Chapter Icon designs.

Whatever ribbon the first keyword was selected from would determine the base shape of the chapter icon. This shape would be multiplied according to the number of overall keywords selected, and the colour determined by the corresponding keyword ribbon colour. It would enable each chapter icon to be unique, and determined according to the user’s keyword selection.

The chapter icons would then be positioned on the underlying map, as if they belonged to a specific location. One option was to have these positions determined programmatically, through the use of a complex mathematical curve (eg a fractal), but the patterns these curves drew were too regular and didn't feel organic enough. Instead I plotted 1000 chapter positions by hand, making a record of these positions with a separate software application designed specially to do this, starting from the centre of the map and emanating outward.

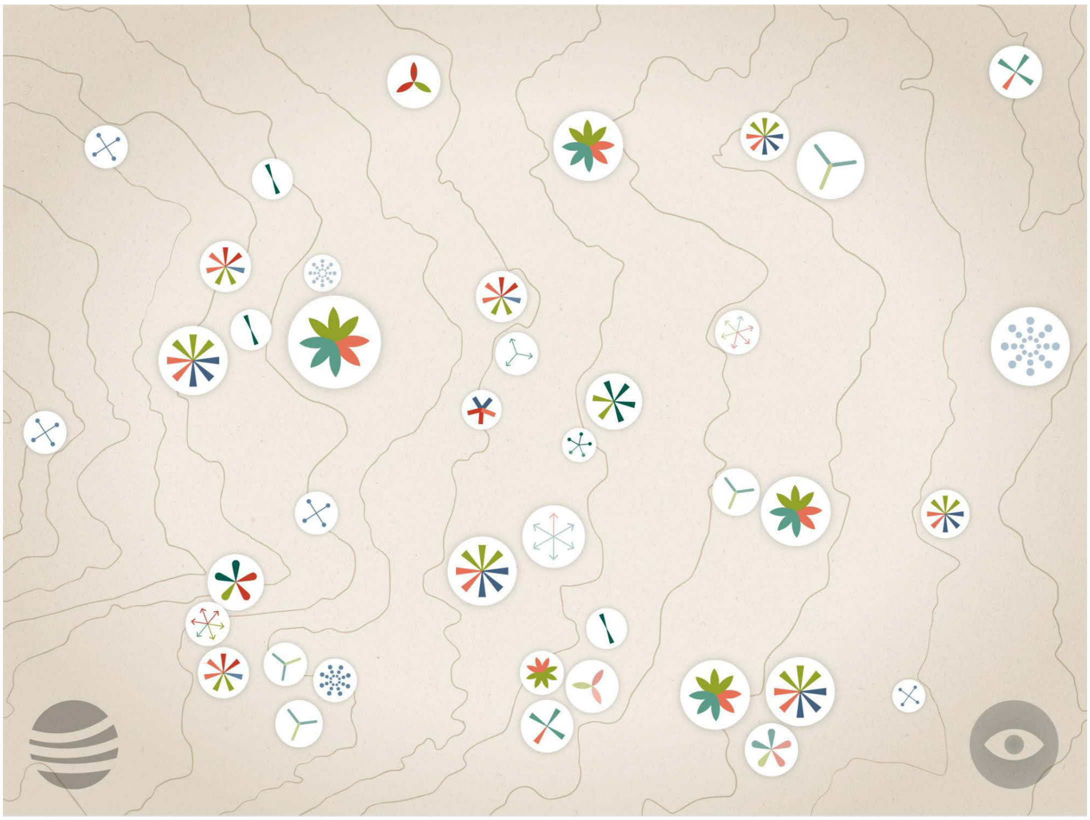
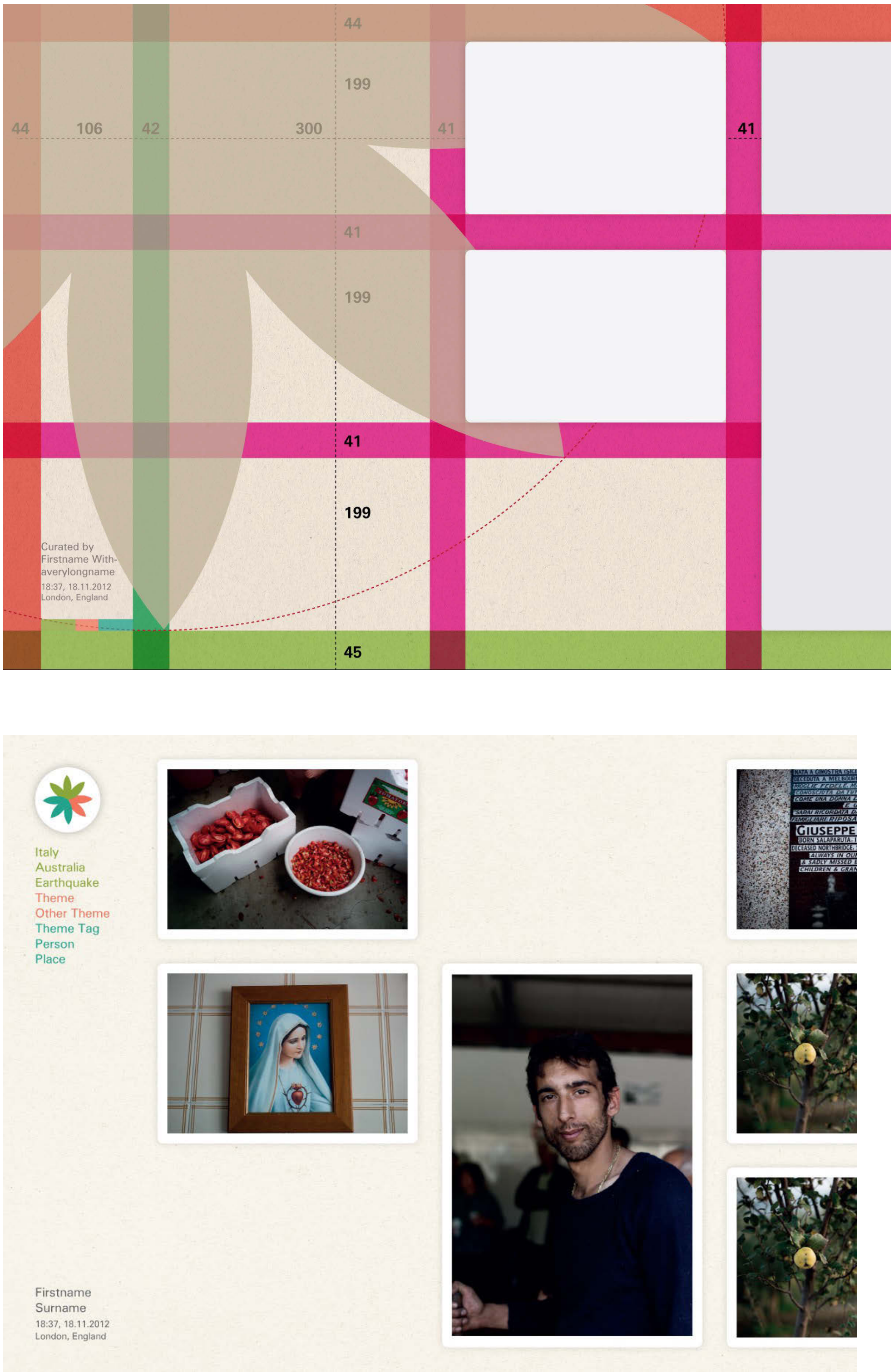


Figure 14: Final Approach for the Explore Screen.

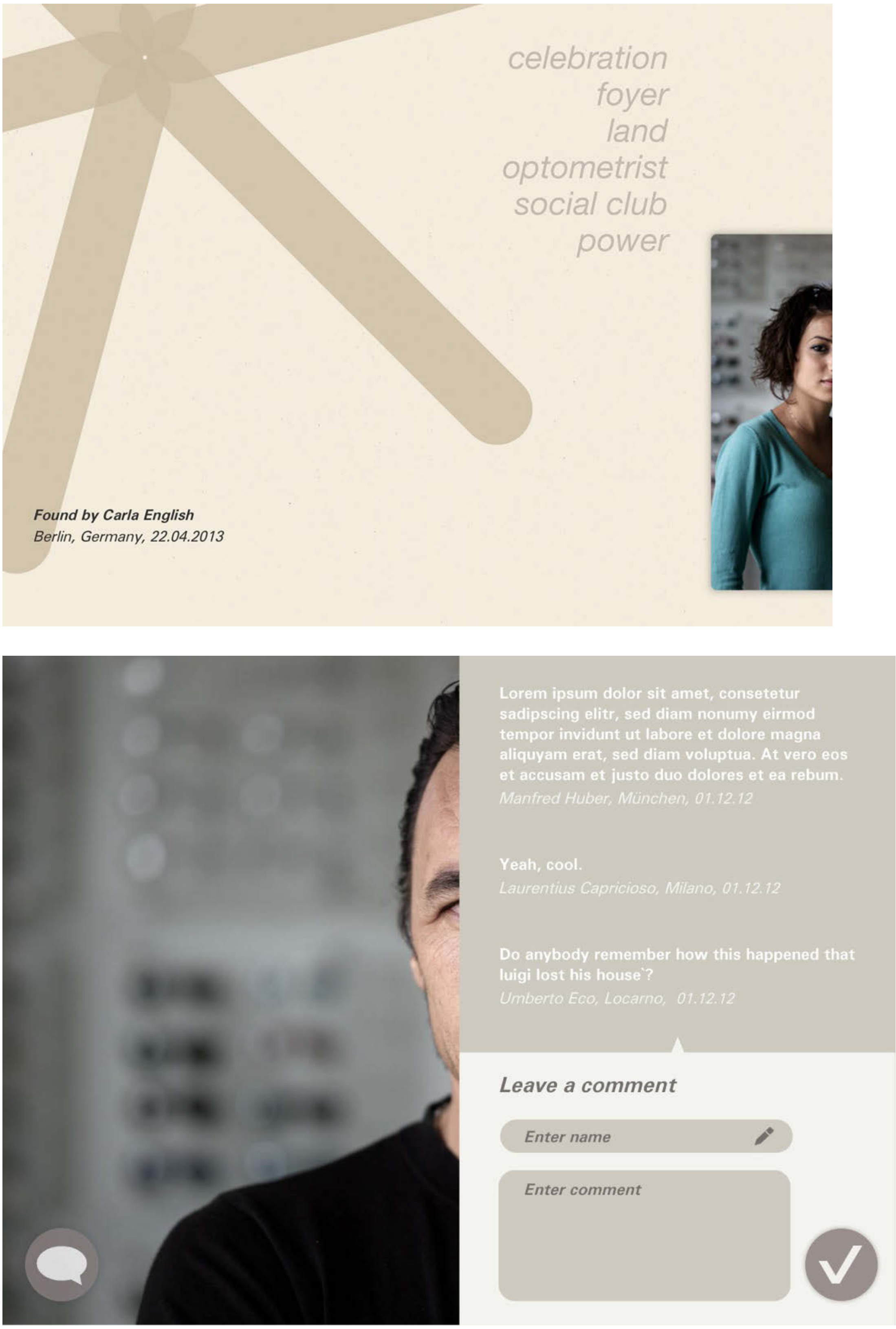
Another method used to enhance the dynamic feel of the app was in the implementation of the chapter thumbnail screen.

When viewing the photo selection I wanted to avoid having the selected photos displayed in the same layout configuration over and over. While there would obviously be a full screen option, where the user would be able to scroll through full-sized images of the photo selection, this wouldn’t allow for a great deal of variation graphically. I did want to include an overall view of the photos in thumbnail form, which the user would see prior to entering full screen mode. This screen offered an opportunity to play with layout variations.

As randomness was an intrinsic philosophical mechanic within the app, it felt right to build some randomness into the way the photos were displayed in their thumbnail view. An underlying grid was created within which photos could be displayed in either portrait or landscape orientation, and where their positions would be determined randomly. This meant it didn’t matter how varied the photo selection was for each chapter (number, portrait, landscape), as the grid would accommodate infinite variations. A graphical watermark of the chapter’s icon graphic would also be included as part of the graphic design of this view. It would help reinforce the Chapter’s individual “identity” and add another variable graphical element to the layout.



Figures 15 & 16: The Chapter View Grid Design



celebration
foyer
land
optometrist
social club
power

Found by Carla English
Berlin, Germany, 22.04.2013

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.
Manfred Huber, München, 01.12.12

Yeah, cool.
Laurentius Capricioso, Milano, 01.12.12

Do anybody remember how this happened that luigi lost his house`?
Umberto Eco, Locarno, 01.12.12

Figures 17 & 18: The Final Approach for the Chapter View and Comments

Development of the application took up a major portion of the research time. Implementing just the minimal set of features saw over 200 issues raised (see Appendix B) and saw over 200 pages of source code written (see Appendix C). The complexities and time involved in developing a digital application cannot be understated, especially when that application is entirely bespoke, without precedents that can be easily copied or repurposed. The primary research value of *Nostalgia Ti Frega*, to my mind, is the contribution it makes to experimentation in application design within the iOS, touch/tablet paradigm.

Release

Unlike an analog artefact, a digital product is not fixed or finished. It may meet an end of life when its development is no longer supported, or the technology it was developed for becomes obsolete. But quite apart from the content that may change as part of the story it tells, its architecture, features, and/or functionality can all be changed, modified, or improved upon. This is in fact often a crucial aspect of many successful digital products: the ability to change and adapt to user needs.

This iteration of *Nostalgia Ti Frega* is merely its first. So long as the iOS framework exists and people continue to use the application, it's conceivable that it will be developed and tweaked further. Likely additions will be functionality that allows users to share information about chapters on other platforms (such as twitter or facebook). It may also prove desirable for users to have the ability to upload and tag their own photographs.

But these questions are ones that will be answered after the application is released. In keeping with the lean agile approach, and also due to resource constraints (development of the project has been self-funded) the functionality of the *Nostalgia Ti Frega* iPad application has been kept to that of a ‘minimum viable product’. This does not mean the project is finished. It will continue to emerge and I look forward to watching its evolution unfold.

APPENDIX A - INTERVIEW QUESTIONS

Nostalgia Ti Frega:

A photographic topography of memory, identity and place

BROAD INTERVIEW THEMES/QUESTIONS

The central questions of the interviews will be:

What do subjects remember of the village?

What does it now mean to them, do they still consider it home?

What do they miss about it?

What things do they use to remind them of it?

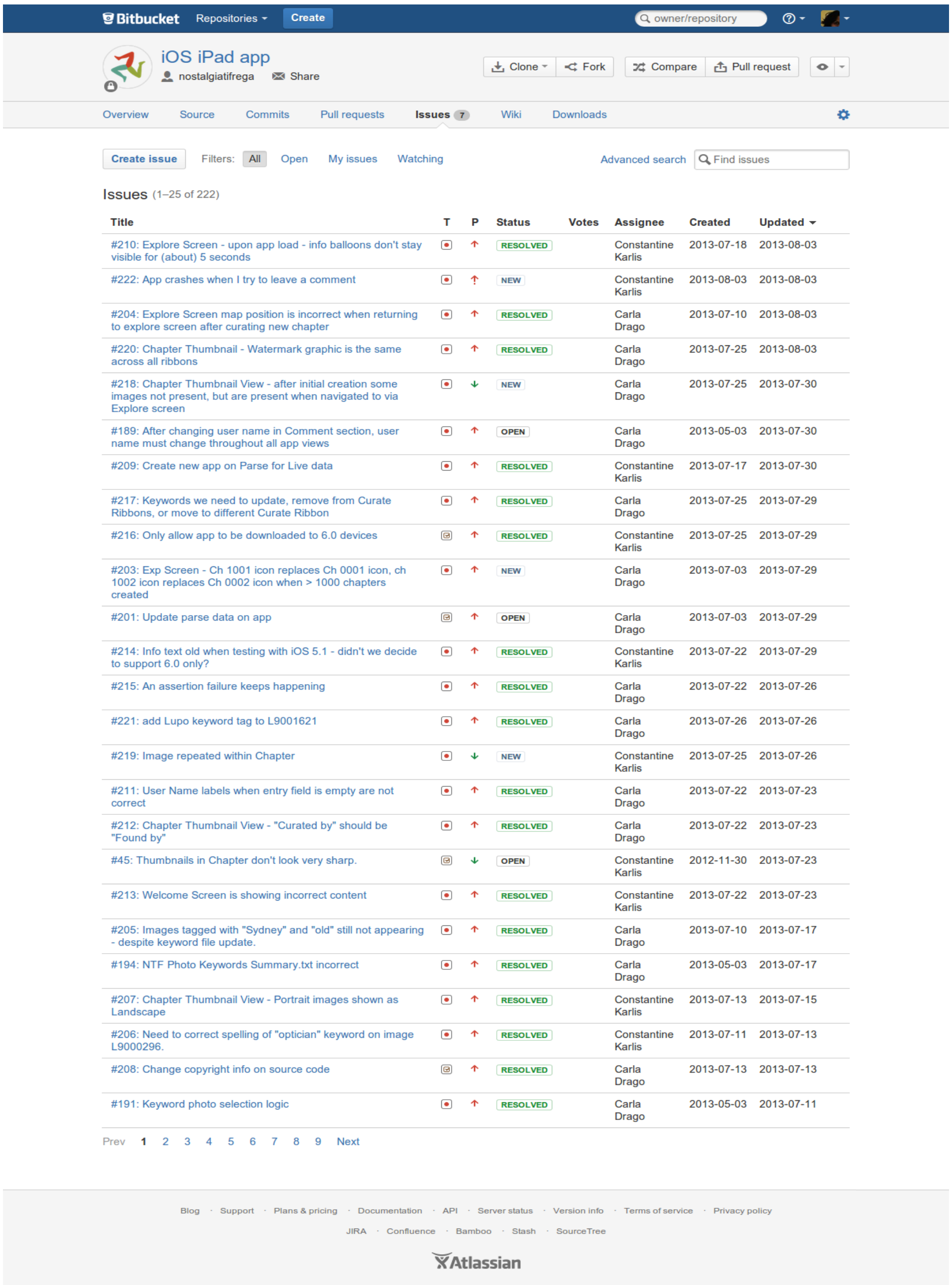
How do they stay connected and communicate with other villagers?

How important are photographs as part of this?

What kinds of media do they engage with to do any of this?

APPENDIX B - APP DEVELOPMENT ISSUE LIST SELECTION

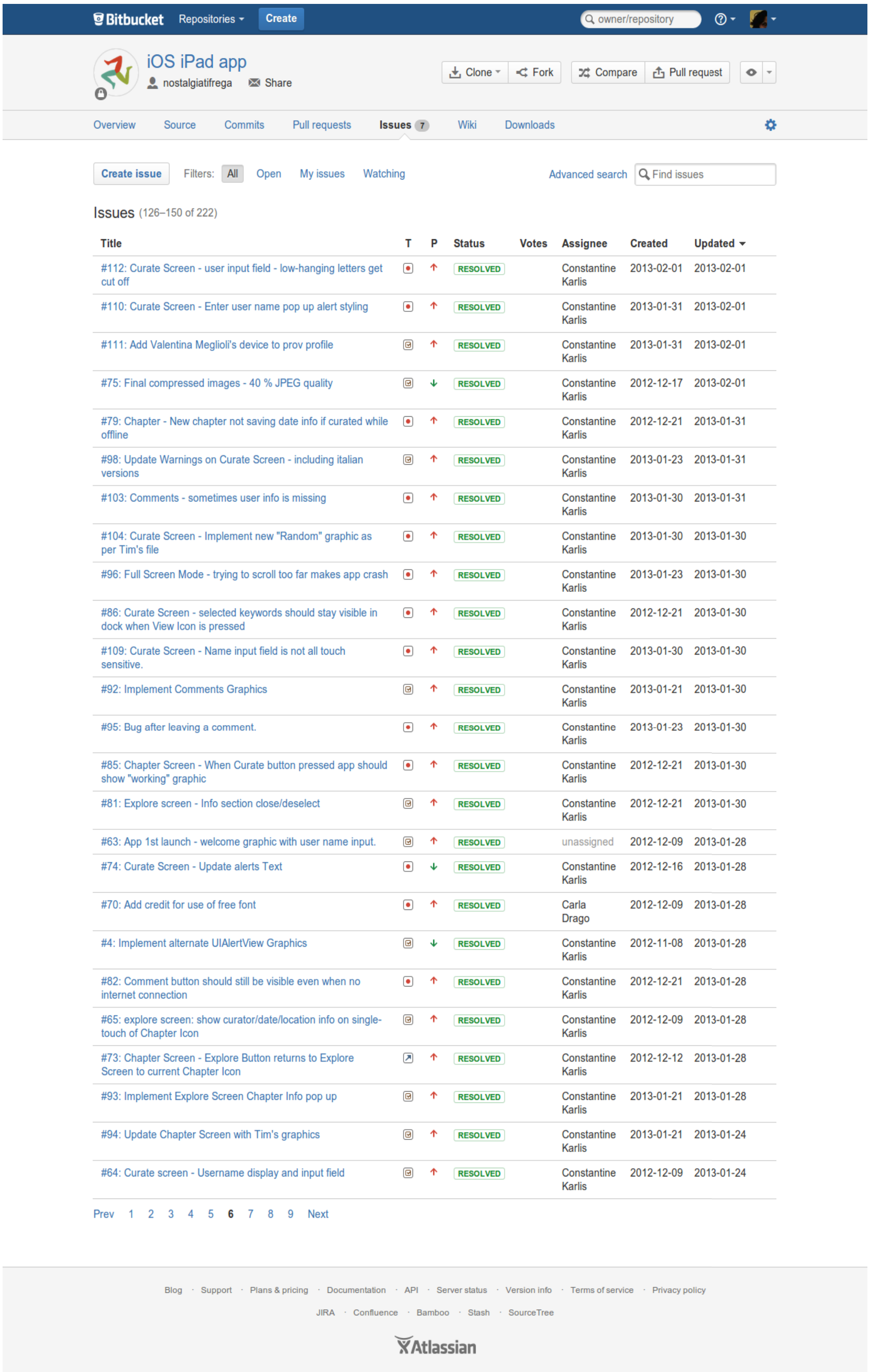
This Appendix includes a selection of screenshots of the issues raised during the development of the the iOS iPad application. It is designed to give a sense of the complexity and extent of the development work done as part of the digital production of *Nostalgia Ti Frega*. For brevity not all the issues have been included.



Issues (1–25 of 222)

Title	T	P	Status	Votes	Assignee	Created	Updated
#210: Explore Screen - upon app load - info balloons don't stay visible for (about) 5 seconds	●	↑	RESOLVED		Constantine Karlis	2013-07-18	2013-08-03
#222: App crashes when I try to leave a comment	●	↑	NEW		Constantine Karlis	2013-08-03	2013-08-03
#204: Explore Screen map position is incorrect when returning to explore screen after curating new chapter	●	↑	RESOLVED		Carla Drago	2013-07-10	2013-08-03
#220: Chapter Thumbnail - Watermark graphic is the same across all ribbons	●	↑	RESOLVED		Carla Drago	2013-07-25	2013-08-03
#218: Chapter Thumbnail View - after initial creation some images not present, but are present when navigated to via Explore screen	●	↓	NEW		Carla Drago	2013-07-25	2013-07-30
#189: After changing user name in Comment section, user name must change throughout all app views	●	↑	OPEN		Carla Drago	2013-05-03	2013-07-30
#209: Create new app on Parse for Live data	●	↑	RESOLVED		Constantine Karlis	2013-07-17	2013-07-30
#217: Keywords we need to update, remove from Curate Ribbons, or move to different Curate Ribbon	●	↑	RESOLVED		Carla Drago	2013-07-25	2013-07-29
#216: Only allow app to be downloaded to 6.0 devices	☑	↑	RESOLVED		Constantine Karlis	2013-07-25	2013-07-29
#203: Exp Screen - Ch 1001 icon replaces Ch 0001 icon, ch 1002 icon replaces Ch 0002 icon when > 1000 chapters created	●	↑	NEW		Carla Drago	2013-07-03	2013-07-29
#201: Update parse data on app	☑	↑	OPEN		Carla Drago	2013-07-03	2013-07-29
#214: Info text old when testing with iOS 5.1 - didn't we decide to support 6.0 only?	●	↑	RESOLVED		Constantine Karlis	2013-07-22	2013-07-29
#215: An assertion failure keeps happening	●	↑	RESOLVED		Carla Drago	2013-07-22	2013-07-26
#221: add Lupo keyword tag to L9001621	●	↑	RESOLVED		Carla Drago	2013-07-26	2013-07-26
#219: Image repeated within Chapter	●	↓	NEW		Constantine Karlis	2013-07-25	2013-07-26
#211: User Name labels when entry field is empty are not correct	●	↑	RESOLVED		Carla Drago	2013-07-22	2013-07-23
#212: Chapter Thumbnail View - "Curated by" should be "Found by"	●	↑	RESOLVED		Carla Drago	2013-07-22	2013-07-23
#45: Thumbnails in Chapter don't look very sharp.	☑	↓	OPEN		Constantine Karlis	2012-11-30	2013-07-23
#213: Welcome Screen is showing incorrect content	●	↑	RESOLVED		Constantine Karlis	2013-07-22	2013-07-23
#205: Images tagged with "Sydney" and "old" still not appearing - despite keyword file update.	●	↑	RESOLVED		Carla Drago	2013-07-10	2013-07-17
#194: NTF Photo Keywords Summary.txt incorrect	●	↑	RESOLVED		Carla Drago	2013-05-03	2013-07-17
#207: Chapter Thumbnail View - Portrait images shown as Landscape	●	↑	RESOLVED		Constantine Karlis	2013-07-13	2013-07-15
#206: Need to correct spelling of "optician" keyword on image L9000296.	●	↑	RESOLVED		Constantine Karlis	2013-07-11	2013-07-13
#208: Change copyright info on source code	☑	↑	RESOLVED		Carla Drago	2013-07-13	2013-07-13
#191: Keyword photo selection logic	●	↑	RESOLVED		Carla Drago	2013-05-03	2013-07-11

Prev 1 2 3 4 5 6 7 8 9 Next
Blog · Support · Plans & pricing · Documentation · API · Server status · Version info · Terms of service · Privacy policy
JIRA · Confluence · Bamboo · Stash · SourceTree
Atlassian



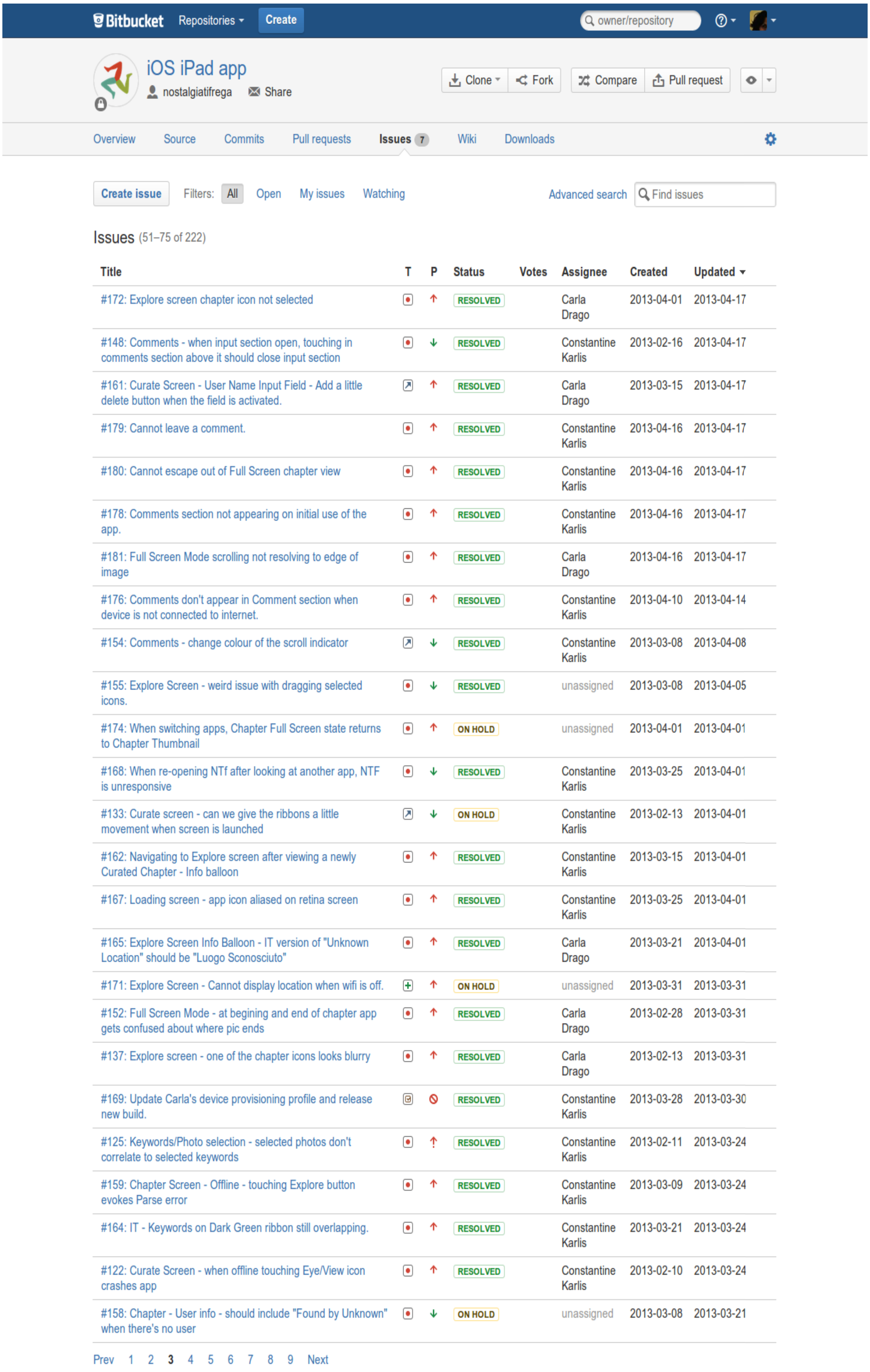
iOS iPad app

nostalgiatifrega   Share

Issues (126–150 of 222)

Title	T	P	Status	Votes	Assignee	Created	Updated ▾
#112: Curate Screen - user input field - low-hanging letters get cut off	◉	↑	RESOLVED		Constantine Karlis	2013-02-01	2013-02-01
#110: Curate Screen - Enter user name pop up alert styling	◉	↑	RESOLVED		Constantine Karlis	2013-01-31	2013-02-01
#111: Add Valentina Meglioli's device to prov profile	☑	↑	RESOLVED		Constantine Karlis	2013-01-31	2013-02-01
#75: Final compressed images - 40 % JPEG quality	☑	↓	RESOLVED		Constantine Karlis	2012-12-17	2013-02-01
#79: Chapter - New chapter not saving date info if curated while offline	◉	↑	RESOLVED		Constantine Karlis	2012-12-21	2013-01-31
#98: Update Warnings on Curate Screen - including italian versions	☑	↑	RESOLVED		Constantine Karlis	2013-01-23	2013-01-31
#103: Comments - sometimes user info is missing	◉	↑	RESOLVED		Constantine Karlis	2013-01-30	2013-01-31
#104: Curate Screen - Implement new "Random" graphic as per Tim's file	◉	↑	RESOLVED		Constantine Karlis	2013-01-30	2013-01-30
#96: Full Screen Mode - trying to scroll too far makes app crash	◉	↑	RESOLVED		Constantine Karlis	2013-01-23	2013-01-30
#86: Curate Screen - selected keywords should stay visible in dock when View Icon is pressed	◉	↑	RESOLVED		Constantine Karlis	2012-12-21	2013-01-30
#109: Curate Screen - Name input field is not all touch sensitive.	◉	↑	RESOLVED		Constantine Karlis	2013-01-30	2013-01-30
#92: Implement Comments Graphics	☑	↑	RESOLVED		Constantine Karlis	2013-01-21	2013-01-30
#95: Bug after leaving a comment.	◉	↑	RESOLVED		Constantine Karlis	2013-01-23	2013-01-30
#85: Chapter Screen - When Curate button pressed app should show "working" graphic	◉	↑	RESOLVED		Constantine Karlis	2012-12-21	2013-01-30
#81: Explore screen - Info section close/deselect	☑	↑	RESOLVED		Constantine Karlis	2012-12-21	2013-01-30
#63: App 1st launch - welcome graphic with user name input.	☑	↑	RESOLVED		unassigned	2012-12-09	2013-01-28
#74: Curate Screen - Update alerts Text	◉	↓	RESOLVED		Constantine Karlis	2012-12-16	2013-01-28
#70: Add credit for use of free font	◉	↑	RESOLVED		Carla Drago	2012-12-09	2013-01-28
#4: Implement alternate UIAlertView Graphics	☑	↓	RESOLVED		Constantine Karlis	2012-11-08	2013-01-28
#82: Comment button should still be visible even when no internet connection	◉	↑	RESOLVED		Constantine Karlis	2012-12-21	2013-01-28
#65: explore screen: show curator/date/location info on single-touch of Chapter Icon	☑	↑	RESOLVED		Constantine Karlis	2012-12-09	2013-01-28
#73: Chapter Screen - Explore Button returns to Explore Screen to current Chapter Icon	↗	↑	RESOLVED		Constantine Karlis	2012-12-12	2013-01-28
#93: Implement Explore Screen Chapter Info pop up	☑	↑	RESOLVED		Constantine Karlis	2013-01-21	2013-01-28
#94: Update Chapter Screen with Tim's graphics	☑	↑	RESOLVED		Constantine Karlis	2013-01-21	2013-01-24
#64: Curate screen - Username display and input field	☑	↑	RESOLVED		Constantine Karlis	2012-12-09	2013-01-24

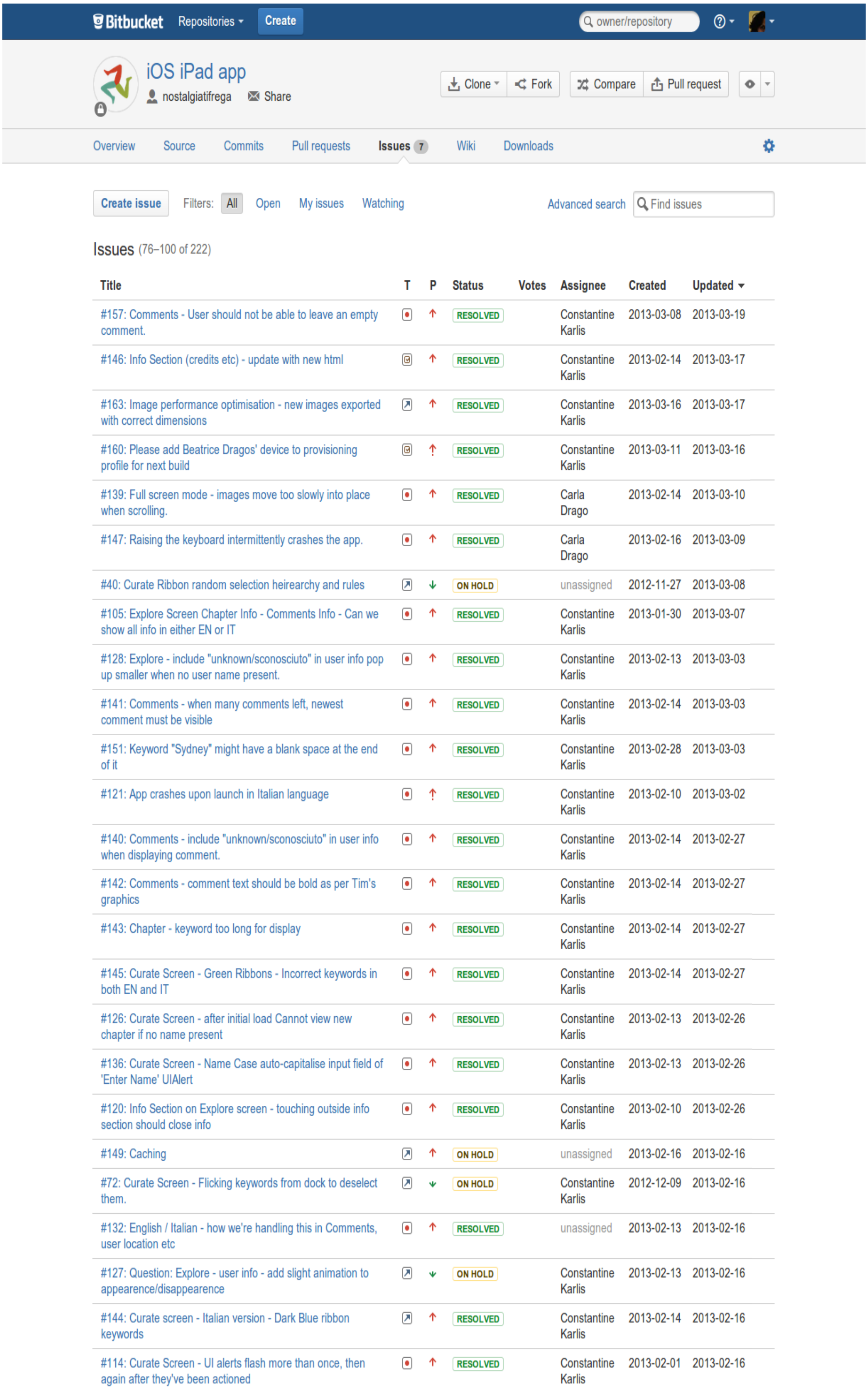
Prev 1 2 3 4 5 6 7 8 9 Next
Blog · Support · Plans & pricing · Documentation · API · Server status · Version info · Terms of service · Privacy policy
JIRA · Confluence · Bamboo · Stash · SourceTree
Atlassian



iOS iPad app

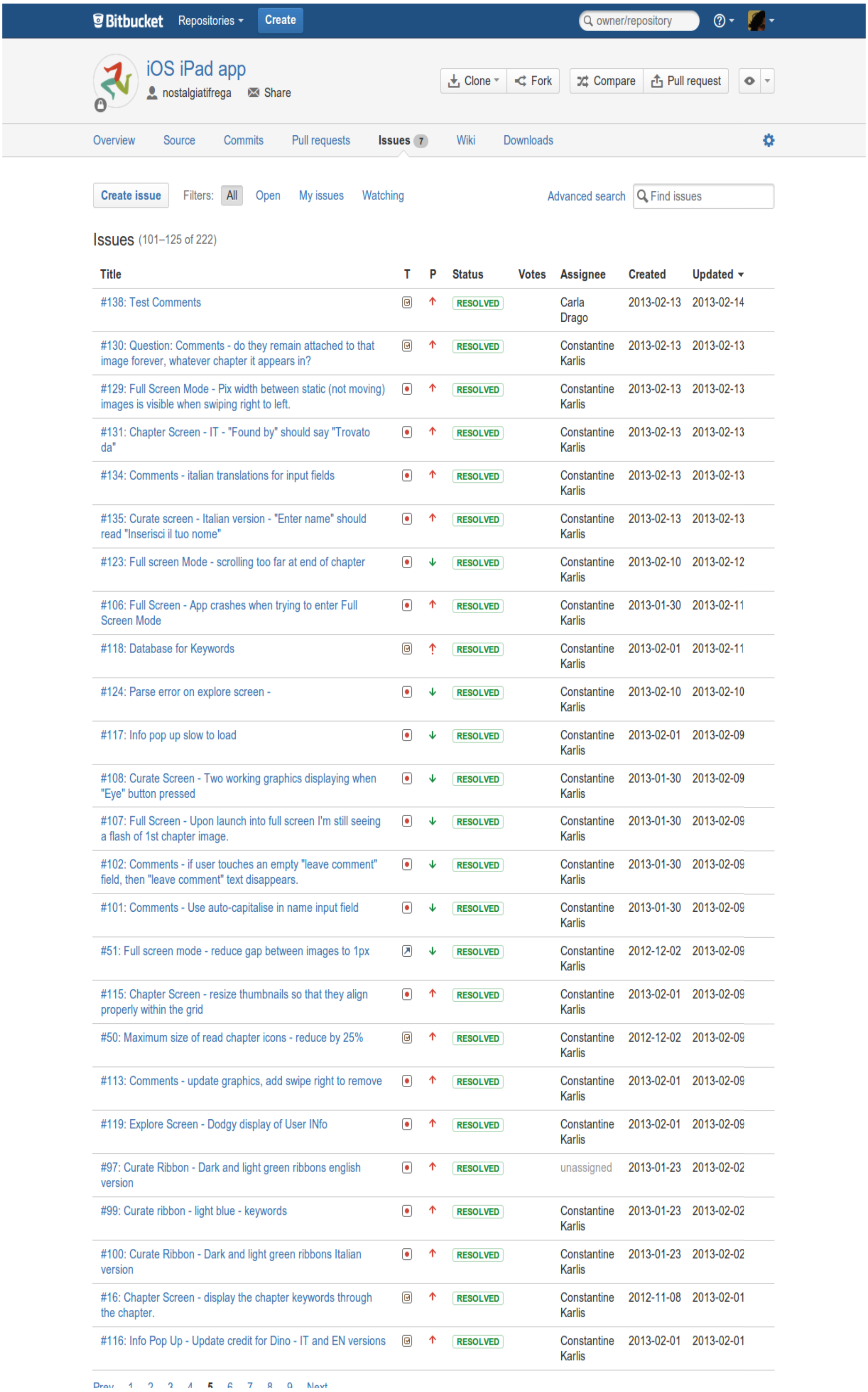
Issues (51–75 of 222)

Title	T	P	Status	Votes	Assignee	Created	Updated ▾
#172: Explore screen chapter icon not selected	●	↑	RESOLVED		Carla Drago	2013-04-01	2013-04-17
#148: Comments - when input section open, touching in comments section above it should close input section	●	↓	RESOLVED		Constantine Karlis	2013-02-16	2013-04-17
#161: Curate Screen - User Name Input Field - Add a little delete button when the field is activated.	↗	↑	RESOLVED		Carla Drago	2013-03-15	2013-04-17
#179: Cannot leave a comment.	●	↑	RESOLVED		Constantine Karlis	2013-04-16	2013-04-17
#180: Cannot escape out of Full Screen chapter view	●	↑	RESOLVED		Constantine Karlis	2013-04-16	2013-04-17
#178: Comments section not appearing on initial use of the app.	●	↑	RESOLVED		Constantine Karlis	2013-04-16	2013-04-17
#181: Full Screen Mode scrolling not resolving to edge of image	●	↑	RESOLVED		Carla Drago	2013-04-16	2013-04-17
#176: Comments don't appear in Comment section when device is not connected to internet.	●	↑	RESOLVED		Constantine Karlis	2013-04-10	2013-04-14
#154: Comments - change colour of the scroll indicator	↗	↓	RESOLVED		Constantine Karlis	2013-03-08	2013-04-08
#155: Explore Screen - weird issue with dragging selected icons.	●	↓	RESOLVED		unassigned	2013-03-08	2013-04-05
#174: When switching apps, Chapter Full Screen state returns to Chapter Thumbnail	●	↑	ON HOLD		unassigned	2013-04-01	2013-04-01
#168: When re-opening NTf after looking at another app, NTF is unresponsive	●	↓	RESOLVED		Constantine Karlis	2013-03-25	2013-04-01
#133: Curate screen - can we give the ribbons a little movement when screen is launched	↗	↓	ON HOLD		Constantine Karlis	2013-02-13	2013-04-01
#162: Navigating to Explore screen after viewing a newly Curated Chapter - Info balloon	●	↑	RESOLVED		Constantine Karlis	2013-03-15	2013-04-01
#167: Loading screen - app icon aliased on retina screen	●	↑	RESOLVED		Constantine Karlis	2013-03-25	2013-04-01
#165: Explore Screen Info Balloon - IT version of "Unknown Location" should be "Luogo Sconosciuto"	●	↑	RESOLVED		Carla Drago	2013-03-21	2013-04-01
#171: Explore Screen - Cannot display location when wifi is off.	+	↑	ON HOLD		unassigned	2013-03-31	2013-03-31
#152: Full Screen Mode - at begining and end of chapter app gets confused about where pic ends	●	↑	RESOLVED		Carla Drago	2013-02-28	2013-03-31
#137: Explore screen - one of the chapter icons looks blurry	●	↑	RESOLVED		Carla Drago	2013-02-13	2013-03-31
#169: Update Carla's device provisioning profile and release new build.	▣	⊘	RESOLVED		Constantine Karlis	2013-03-28	2013-03-30
#125: Keywords/Photo selection - selected photos don't correlate to selected keywords	●	↑	RESOLVED		Constantine Karlis	2013-02-11	2013-03-24
#159: Chapter Screen - Offline - touching Explore button evokes Parse error	●	↑	RESOLVED		Constantine Karlis	2013-03-09	2013-03-24
#164: IT - Keywords on Dark Green ribbon still overlapping.	●	↑	RESOLVED		Constantine Karlis	2013-03-21	2013-03-24
#122: Curate Screen - when offline touching Eye/View icon crashes app	●	↑	RESOLVED		Constantine Karlis	2013-02-10	2013-03-24
#158: Chapter - User info - should include "Found by Unknown" when there's no user	●	↓	ON HOLD		unassigned	2013-03-08	2013-03-21



Issues (76–100 of 222)

Title	T	P	Status	Votes	Assignee	Created	Updated ▾
#157: Comments - User should not be able to leave an empty comment.	●	↑	RESOLVED		Constantine Karlis	2013-03-08	2013-03-19
#146: Info Section (credits etc) - update with new html	☐	↑	RESOLVED		Constantine Karlis	2013-02-14	2013-03-17
#163: Image performance optimisation - new images exported with correct dimensions	↗	↑	RESOLVED		Constantine Karlis	2013-03-16	2013-03-17
#160: Please add Beatrice Dragos' device to provisioning profile for next build	☐	↑	RESOLVED		Constantine Karlis	2013-03-11	2013-03-16
#139: Full screen mode - images move too slowly into place when scrolling.	●	↑	RESOLVED		Carla Drago	2013-02-14	2013-03-10
#147: Raising the keyboard intermittently crashes the app.	●	↑	RESOLVED		Carla Drago	2013-02-16	2013-03-09
#40: Curate Ribbon random selection heirearchy and rules	↗	↓	ON HOLD		unassigned	2012-11-27	2013-03-08
#105: Explore Screen Chapter Info - Comments Info - Can we show all info in either EN or IT	●	↑	RESOLVED		Constantine Karlis	2013-01-30	2013-03-07
#128: Explore - include "unknown/sconosciuto" in user info pop up smaller when no user name present.	●	↑	RESOLVED		Constantine Karlis	2013-02-13	2013-03-03
#141: Comments - when many comments left, newest comment must be visible	●	↑	RESOLVED		Constantine Karlis	2013-02-14	2013-03-03
#151: Keyword "Sydney" might have a blank space at the end of it	●	↑	RESOLVED		Constantine Karlis	2013-02-28	2013-03-03
#121: App crashes upon launch in Italian language	●	↑	RESOLVED		Constantine Karlis	2013-02-10	2013-03-02
#140: Comments - include "unknown/sconosciuto" in user info when displaying comment.	●	↑	RESOLVED		Constantine Karlis	2013-02-14	2013-02-27
#142: Comments - comment text should be bold as per Tim's graphics	●	↑	RESOLVED		Constantine Karlis	2013-02-14	2013-02-27
#143: Chapter - keyword too long for display	●	↑	RESOLVED		Constantine Karlis	2013-02-14	2013-02-27
#145: Curate Screen - Green Ribbons - Incorrect keywords in both EN and IT	●	↑	RESOLVED		Constantine Karlis	2013-02-14	2013-02-27
#126: Curate Screen - after initial load Cannot view new chapter if no name present	●	↑	RESOLVED		Constantine Karlis	2013-02-13	2013-02-26
#136: Curate Screen - Name Case auto-capitalise input field of 'Enter Name' UIAlert	●	↑	RESOLVED		Constantine Karlis	2013-02-13	2013-02-26
#120: Info Section on Explore screen - touching outside info section should close info	●	↑	RESOLVED		Constantine Karlis	2013-02-10	2013-02-26
#149: Caching	↗	↑	ON HOLD		unassigned	2013-02-16	2013-02-16
#72: Curate Screen - Flicking keywords from dock to deselect them.	↗	↓	ON HOLD		Constantine Karlis	2012-12-09	2013-02-16
#132: English / Italian - how we're handling this in Comments, user location etc	●	↑	RESOLVED		unassigned	2013-02-13	2013-02-16
#127: Question: Explore - user info - add slight animation to appearence/disappearence	↗	↓	ON HOLD		Constantine Karlis	2013-02-13	2013-02-16
#144: Curate screen - Italian version - Dark Blue ribbon keywords	↗	↑	RESOLVED		Constantine Karlis	2013-02-14	2013-02-16
#114: Curate Screen - UI alerts flash more than once, then again after they've been actioned	●	↑	RESOLVED		Constantine Karlis	2013-02-01	2013-02-16

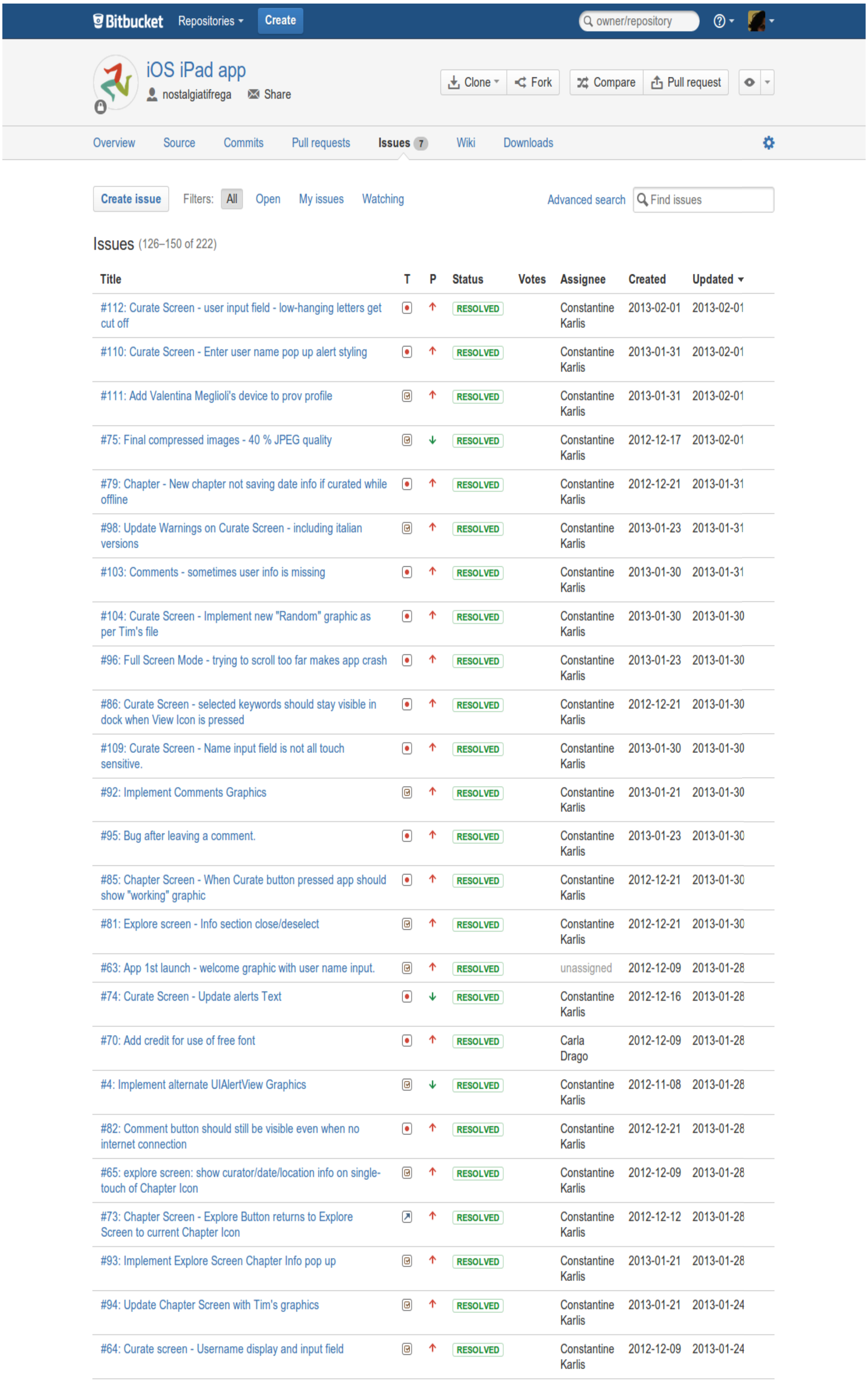


iOS iPad app

nostalgiatifrega Share

Issues (101–125 of 222)

Title	T	P	Status	Votes	Assignee	Created	Updated ▾
#138: Test Comments	☑	↑	RESOLVED		Carla Drago	2013-02-13	2013-02-14
#130: Question: Comments - do they remain attached to that image forever, whatever chapter it appears in?	☑	↑	RESOLVED		Constantine Karlis	2013-02-13	2013-02-13
#129: Full Screen Mode - Pix width between static (not moving) images is visible when swiping right to left.	●	↑	RESOLVED		Constantine Karlis	2013-02-13	2013-02-13
#131: Chapter Screen - IT - "Found by" should say "Trovato da"	●	↑	RESOLVED		Constantine Karlis	2013-02-13	2013-02-13
#134: Comments - italian translations for input fields	●	↑	RESOLVED		Constantine Karlis	2013-02-13	2013-02-13
#135: Curate screen - Italian version - "Enter name" should read "Inserisci il tuo nome"	●	↑	RESOLVED		Constantine Karlis	2013-02-13	2013-02-13
#123: Full screen Mode - scrolling too far at end of chapter	●	↓	RESOLVED		Constantine Karlis	2013-02-10	2013-02-12
#106: Full Screen - App crashes when trying to enter Full Screen Mode	●	↑	RESOLVED		Constantine Karlis	2013-01-30	2013-02-11
#118: Database for Keywords	☑	↑	RESOLVED		Constantine Karlis	2013-02-01	2013-02-11
#124: Parse error on explore screen -	●	↓	RESOLVED		Constantine Karlis	2013-02-10	2013-02-10
#117: Info pop up slow to load	●	↓	RESOLVED		Constantine Karlis	2013-02-01	2013-02-09
#108: Curate Screen - Two working graphics displaying when "Eye" button pressed	●	↓	RESOLVED		Constantine Karlis	2013-01-30	2013-02-09
#107: Full Screen - Upon launch into full screen I'm still seeing a flash of 1st chapter image.	●	↓	RESOLVED		Constantine Karlis	2013-01-30	2013-02-09
#102: Comments - if user touches an empty "leave comment" field, then "leave comment" text disappears.	●	↓	RESOLVED		Constantine Karlis	2013-01-30	2013-02-09
#101: Comments - Use auto-capitalise in name input field	●	↓	RESOLVED		Constantine Karlis	2013-01-30	2013-02-09
#51: Full screen mode - reduce gap between images to 1px	↗	↓	RESOLVED		Constantine Karlis	2012-12-02	2013-02-09
#115: Chapter Screen - resize thumbnails so that they align properly within the grid	●	↑	RESOLVED		Constantine Karlis	2013-02-01	2013-02-09
#50: Maximum size of read chapter icons - reduce by 25%	☑	↑	RESOLVED		Constantine Karlis	2012-12-02	2013-02-09
#113: Comments - update graphics, add swipe right to remove	●	↑	RESOLVED		Constantine Karlis	2013-02-01	2013-02-09
#119: Explore Screen - Dodgy display of User INfo	●	↑	RESOLVED		Constantine Karlis	2013-02-01	2013-02-09
#97: Curate Ribbon - Dark and light green ribbons english version	●	↑	RESOLVED		unassigned	2013-01-23	2013-02-02
#99: Curate ribbon - light blue - keywords	●	↑	RESOLVED		Constantine Karlis	2013-01-23	2013-02-02
#100: Curate Ribbon - Dark and light green ribbons Italian version	●	↑	RESOLVED		Constantine Karlis	2013-01-23	2013-02-02
#16: Chapter Screen - display the chapter keywords through the chapter.	☑	↑	RESOLVED		Constantine Karlis	2012-11-08	2013-02-01
#116: Info Pop Up - Update credit for Dino - IT and EN versions	☑	↑	RESOLVED		Constantine Karlis	2013-02-01	2013-02-01



Issues (126–150 of 222)

Title	T	P	Status	Votes	Assignee	Created	Updated ▾
#112: Curate Screen - user input field - low-hanging letters get cut off	●	↑	RESOLVED		Constantine Karlis	2013-02-01	2013-02-01
#110: Curate Screen - Enter user name pop up alert styling	●	↑	RESOLVED		Constantine Karlis	2013-01-31	2013-02-01
#111: Add Valentina Meglioli's device to prov profile	▣	↑	RESOLVED		Constantine Karlis	2013-01-31	2013-02-01
#75: Final compressed images - 40 % JPEG quality	▣	↓	RESOLVED		Constantine Karlis	2012-12-17	2013-02-01
#79: Chapter - New chapter not saving date info if curated while offline	●	↑	RESOLVED		Constantine Karlis	2012-12-21	2013-01-31
#98: Update Warnings on Curate Screen - including italian versions	▣	↑	RESOLVED		Constantine Karlis	2013-01-23	2013-01-31
#103: Comments - sometimes user info is missing	●	↑	RESOLVED		Constantine Karlis	2013-01-30	2013-01-31
#104: Curate Screen - Implement new "Random" graphic as per Tim's file	●	↑	RESOLVED		Constantine Karlis	2013-01-30	2013-01-30
#96: Full Screen Mode - trying to scroll too far makes app crash	●	↑	RESOLVED		Constantine Karlis	2013-01-23	2013-01-30
#86: Curate Screen - selected keywords should stay visible in dock when View Icon is pressed	●	↑	RESOLVED		Constantine Karlis	2012-12-21	2013-01-30
#109: Curate Screen - Name input field is not all touch sensitive.	●	↑	RESOLVED		Constantine Karlis	2013-01-30	2013-01-30
#92: Implement Comments Graphics	▣	↑	RESOLVED		Constantine Karlis	2013-01-21	2013-01-30
#95: Bug after leaving a comment.	●	↑	RESOLVED		Constantine Karlis	2013-01-23	2013-01-30
#85: Chapter Screen - When Curate button pressed app should show "working" graphic	●	↑	RESOLVED		Constantine Karlis	2012-12-21	2013-01-30
#81: Explore screen - Info section close/deselect	▣	↑	RESOLVED		Constantine Karlis	2012-12-21	2013-01-30
#63: App 1st launch - welcome graphic with user name input.	▣	↑	RESOLVED		unassigned	2012-12-09	2013-01-28
#74: Curate Screen - Update alerts Text	●	↓	RESOLVED		Constantine Karlis	2012-12-16	2013-01-28
#70: Add credit for use of free font	●	↑	RESOLVED		Carla Drago	2012-12-09	2013-01-28
#4: Implement alternate UIAlertView Graphics	▣	↓	RESOLVED		Constantine Karlis	2012-11-08	2013-01-28
#82: Comment button should still be visible even when no internet connection	●	↑	RESOLVED		Constantine Karlis	2012-12-21	2013-01-28
#65: explore screen: show curator/date/location info on single-touch of Chapter Icon	▣	↑	RESOLVED		Constantine Karlis	2012-12-09	2013-01-28
#73: Chapter Screen - Explore Button returns to Explore Screen to current Chapter Icon	↗	↑	RESOLVED		Constantine Karlis	2012-12-12	2013-01-28
#93: Implement Explore Screen Chapter Info pop up	▣	↑	RESOLVED		Constantine Karlis	2013-01-21	2013-01-28
#94: Update Chapter Screen with Tim's graphics	▣	↑	RESOLVED		Constantine Karlis	2013-01-21	2013-01-24
#64: Curate screen - Username display and input field	▣	↑	RESOLVED		Constantine Karlis	2012-12-09	2013-01-24

APPENDIX C - iOS SOURCE CODE SELECTION

This Appendix includes a selection of the source code written for the iOS iPad application. It is designed to give a sense of the complexity and extent of the development work done as part of the digital production of *Nostalgia Ti Frega*. For brevity not all the source code has been included.

```
//
//  AppDelegate.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "NTFAppController.h"

@class MainViewController;

@interface NTFAppDelegate : UIResponder <UIApplicationDelegate>

@property (strong, nonatomic) UIWindow *window;

@property (strong, nonatomic) MainViewController *viewController;

@property (strong, nonatomic) NTFAppController *appController;

@end

//
//  AppDelegate.m
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "NTFAppDelegate.h"
#import "MainViewController.h"
#import "ExploreViewController.h"
#import "NTFLocationManager.h"
#import "NTFServices.h"
#import "DCIntrospect.h"

@implementation NTFAppDelegate
{
    NTFLocationManager *_lm;
}
```

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    [self setWindow:[[UIWindow alloc] initWithFrame:[[UIScreen
mainScreen] bounds]]];

    [self setViewController:[[MainViewController alloc] init]];
    [self.viewController pushViewController:[[ExploreViewController
alloc] init] animated:NO];

    [self setAppController:[[NTFAppController alloc] init]];

    [[NTFServices locationManager] startUpdatingLocation];

    [self.window setRootViewController:self.viewController];
    [self.window makeKeyAndVisible];

    // always call after makeKeyAndDisplay.
#if TARGET_IPHONE_SIMULATOR
    [[DCIntrospect sharedIntrospector] start];
#endif

    return YES;
}

- (NSUInteger)application:(UIApplication *)application
supportedInterfaceOrientationsForWindow:(UIWindow *)window
{
    return UIInterfaceOrientationMaskLandscape;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    /*
     Sent when the application is about to move from active to
inactive state. This can occur for certain types of temporary
interruptions (such as an incoming phone call or SMS message) or
when the user quits the application and it begins the transition to
the background state.
     Use this method to pause ongoing tasks, disable timers, and
throttle down OpenGL ES frame rates. Games should use this method to
pause the game.
     */
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    /*
     Use this method to release shared resources, save user data,
invalidate timers, and store enough application state information to
restore your application to its current state in case it is
terminated later.
     If your application supports background execution, this method
is called instead of applicationWillTerminate: when the user quits.
     */
    [[NSUserDefaults standardUserDefaults] synchronize];
}

- (void)applicationWillEnterForeground:(UIApplication *)application
{
    [self.appController updateDataIfNeeded];
}

- (void)applicationDidBecomeActive:(UIApplication *)application
{
    /*
     Restart any tasks that were paused (or not yet started) while
the application was inactive. If the application was previously in
the background, optionally refresh the user interface.
     */
}

- (void)applicationWillTerminate:(UIApplication *)application
{
    [[NSUserDefaults standardUserDefaults] synchronize];
    /*
     Called when the application is about to terminate.
     Save data if appropriate.
     See also applicationDidEnterBackground:.
     */
}

@end

//
//  notifications.h
//  Curate
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#define BUTTON_CLICK            @"buttonClick"
#define BUTTON_DOWN             @"buttonDown"
#define BUTTON_UP               @"buttonUp"

#define SHUFFLE                 @"shuffle"

#define ITEM_SELECTED           @"itemSelected"
#define ITEM_DESELECTED         @"itemDeselected"
```

```
#define TOO_MANY_KEYWORDS       @"tooManyKeywords"
#define SUBMIT_KEYWORDS         @"submitKeywords"
#define SELECT_RANDOM_KEYWORDS  @"selectRandomKeywords"
#define CLEAR_KEYWORDS          @"clearRandomKeywords"

#define SHOW_EXPLORE            @"showExplore"

#define SAVE_CHAPTER            @"saveChapter"
#define CHAPTER_SAVED           @"chapterSaved"
#define CHAPTER_SAVE_FAILED     @"chapterSaveFailed"
#define CHAPTER_UPDATED         @"chapterUpdated"
#define CHAPTERS_UPDATED        @"chaptersUpdated"
#define UPDATE_CHAPTERS         @"updateChapters"

#define LOAD_CHAPTERS           @"loadChapters"
#define CHAPTERS_LOADED         @"chaptersLoaded"
#define CHAPTER_LOAD_FAILED     @"chapterLoadFailed"

#define LOAD_COMMENTS           @"loadComments"
#define COMMENTS_LOADED         @"commentsLoaded"
#define COMMENTS_LOAD_FAILED    @"commentsLoadFailed"

#define ADD_COMMENT             @"addComment"
#define COMMENT_ADDED           @"commentAdded"
#define COMMENT_ADD_FAILED      @"commentAddFailed"

#define NAVIGATE_TO_CURATE      @"navigateToCurate"
#define NAVIGATE_TO_EXPLORE     @"navigateToExplore"
#define NAVIGATE_TO_CHAPTER     @"navigateToChapter"

#define CHANGE_USERNAME         @"changeUsername"
#define USERNAME_CHANGED         @"usernameChanged"

#define LOCATION_FOUND          @"locationFound"

#define DISPLAY_ERROR           @"displayError"

//
//  constants.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#ifndef NostalgiaTiFrega_constants_h
#define NostalgiaTiFrega_constants_h

#define VIEWED_CHAPTERS     @"viewedChapters"
#define USERNAME            @"username"
```

```
#define IMAGE_VO            @"imageVO"
#define COMMENT_VO          @"commentVO"
#define ERROR               @"error"
#define PETAL_TYPE          @"petalType"
#define VO                  @"vo"
#define USER_WAS_WELCOMED   @"userWasWelcomed"
#define LAST_UPDATE         @"lastUpdate"
#define MAX_CHAPTERS        1000

#endif



//
//  NTFCurateModel.h
//  NTFCreateTest1
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#define MODEL_UPDATE @"modelUpdate"

#import <Foundation/Foundation.h>
#import "KeywordVO.h"

@interface KeywordModel : NSObject

@property (nonatomic , strong , readonly) NSMutableArray
*categories;
@property (nonatomic , strong , readonly) NSMutableArray
*allKeywords;
@property (nonatomic, readonly) NSMutableArray *selectedItems;

+(id)sharedModel;

-(KeywordVO*)getItemById:(int)itemId;
-(NSArray*)keywordsWithIds:(NSArray*)itemIds;
-(void)toggleSelectItemById:(int)itemId;
-(int)numSelectedItems;
-(int)maxSelectedItems;
-(void)clearSelected;
-(int)getCategoryIndexByItemId:(int)itemId;


@end

//
//  NTFCurateModel.m
//  NTFCreateTest1
//
```

```
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//
#import "KeywordModel.h"
#import "notifications.h"

@interface KeywordModel ()

@end


@implementation KeywordModel
{
    NSDictionary *_keywordsById;
}

+(id)sharedModel
{
    static dispatch_once_t pred;
    static id shared = nil;

    dispatch_once(&pred, ^{
        shared = [[[self class] alloc] init];
    });
    return shared;
}

-(id)init
{
    self = [super init];
    if (self)
    {
        NSString *path = [[NSBundle mainBundle]
pathForResource:@"curateItems" ofType:@"json"];
        NSString *unsanitisedString = [NSString
stringWithContentsOfFile:path

encoding:NSUTF8StringEncoding error:NULL];
        NSString *sanitisedString = [unsanitisedString
stringByReplacingOccurrencesOfString:@"null" withString:@"\"\""];
        NSData *data = [sanitisedString
dataUsingEncoding:NSUTF8StringEncoding];
        NSError *error = nil;
        NSArray* sourceOuterArray = [NSJSONSerialization
JSONObjectWithData:data

options:NSJSONReadingAllowFragments

error:&error];
        if (error)
        {
            NSLog(@"Error: %@" , error.localizedDescription);
            abort();
        }

        _categories = [[NSMutableArray alloc] init];
        _allKeywords = [[NSMutableArray alloc] init];
        NSMutableDictionary *tempKeywordsById =
[[NSMutableDictionary alloc] init];
        NSMutableArray* destInnerArray = nil;
        KeywordVO* vo = nil;
        for (NSArray* sourceInnerArray in sourceOuterArray)
        {
            destInnerArray = [[NSMutableArray alloc] init];
            for (NSDictionary* sourceObject in sourceInnerArray)
            {
                vo = [[KeywordVO alloc] initWithId:[[sourceObject
objectForKey:@"id"] intValue] title:[sourceObject
objectForKey:@"title"]];
                [destInnerArray addObject:vo];
                [_allKeywords addObject:vo];
                [tempKeywordsById setObject:vo
forKey:@(vo.objectId)];
            }
            [_categories addObject:[destInnerArray
sortedArrayUsingSelector:@selector(localizedCaseInsensitiveCompare:)
]];
        }
        _keywordsById = [NSMutableDictionary
dictionaryWithDictionary:tempKeywordsById];
    }

    return self;
}

-(void)toggleSelectItemById:(NSInteger)itemId
{
    KeywordVO* itemvo = [self getItemById:itemId];
    if (![itemvo isSelected] && [self numSelectedItems] == [self
maxSelectedItems])
    {
        [[NSNotificationCenter defaultCenter]
postNotificationName:TOO_MANY_KEYWORDS object:nil];
    }
    else
    {
        if (itemvo == nil) return;
        [itemvo setIsSelected:![itemvo isSelected]];
        [[NSNotificationCenter defaultCenter]
postNotificationName:MODEL_UPDATE object:itemvo];
```

```
    }
}

-(KeywordVO*)getItemById:(NSInteger)itemId
{
    return _keywordsById[@(itemId)];
}

-(NSArray*)keywordsWithIds:(NSArray*)itemIds
{
    NSMutableArray *out = [[NSMutableArray alloc] init];
    for (NSNumber *itemId in itemIds)
    {
        KeywordVO *keywordVO = _keywordsById[itemId];
        if (keywordVO)
            [out addObject:_keywordsById[itemId]];
    }
    return out;
}

-(int)numSelectedItems
{
    int out = 0;
    for (KeywordVO* itemvo in _allKeywords)
    {
        if (itemvo.isSelected) out++;
    }
    return out;
}

-(NSMutableArray*)selectedItems
{
    NSMutableArray *a = [[NSMutableArray alloc] init];
    for (KeywordVO* itemvo in _allKeywords)
    {
        if (itemvo.isSelected) [a addObject:itemvo];
    }
    return a;
}

-(int)maxSelectedItems
{
    return 8;
}

-(void)clearSelected
{
    for (KeywordVO* itemvo in _allKeywords)
    {
        if (itemvo.isSelected) [self toggleSelectItemById:[itemvo
objectId]];
    }
}

-(int)getCategoryIndexByItemId:(int)itemId
{
    for (int i=0; i<[_categories count]; i++)
    {
        for (KeywordVO* vo in [_categories objectAtIndex:i])
        {
            if ([vo objectId] == itemId) return i;
        }
    }
    return NSNotFound;
}

@end


//
//  NTFCurateItem.h
//  NTFCreateTest1
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <Foundation/Foundation.h>

@interface KeywordVO : NSObject

@property (nonatomic , readonly) NSUInteger objectId;
@property (nonatomic , strong , readonly) NSString* title;
@property BOOL isSelected;

-(id)initWithId:(NSInteger)objectId title:(NSString*)title;
-(NSComparisonResult)localizedCaseInsensitiveCompare:
(NSString*)aString;

@end


//
//  NTFCurateItem.m
//  NTFCreateTest1
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "KeywordVO.h"
```

```
@implementation KeywordVO

@synthesize objectId = _itemId;
@synthesize isSelected = _isSelected;
@synthesize title = _title;

-(id)initWithId:(int)objectId title:(NSString*)title
{
    self = [super init];
    if (self)
    {
        _itemId = objectId;
        _title = title;
    }
    return self;
}

-(NSString *)description
{
    return [NSString stringWithFormat:@"<NTFCurateItemVO itemId: %i
title: '%@' isSelected: %i>" , _itemId , _title , _isSelected];
}

-(NSComparisonResult)localizedCaseInsensitiveCompare:(KeywordVO*)aVo
{
    return [_title localizedCaseInsensitiveCompare:[aVo title]];
}

@end

//
//  CircleVO.h
//  Curate
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <Foundation/Foundation.h>

@interface CircleVO : NSObject

@property (nonatomic , assign) CGPoint pos;
@property (nonatomic , assign) int radius;
@property (nonatomic , retain) UIColor* color;
@property (nonatomic , retain) UIColor* darkColor;

// designated init
-(id)initWithPos:(CGPoint)pp radius:(int)rr color:(UIColor*)cc
darkColor:(UIColor*)dc;
```

```
@end

//
//  CircleVO.m
//  Curate
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "CircleVO.h"

@implementation CircleVO

@synthesize pos;
@synthesize radius;
@synthesize color;
@synthesize darkColor;

-(id)initWithPos:(CGPoint)pp radius:(int)rr color:(UIColor*)cc
darkColor:(UIColor*)dc
{
    self = [super init];
    if (self)
    {
        [self setPos:pp];
        [self setRadius:rr];
        [self setColor:cc];
        [self setDarkColor:dc];
    }
    return self;
}

- (id)init
{
    return [self initWithPos:CGPointZero radius:0 color:nil
darkColor:nil];
}

@end


//
//  CircleModel.h
//  Curate
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <Foundation/Foundation.h>
#import "CircleVO.h"
```

```
@interface CircleModel : NSObject
{
    NSArray* _items;
}

+(id)sharedInstance;

-(NSArray*)items;

@end

//
//  CircleModel.m
//  Curate
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "CircleModel.h"
#import "UIColor+Color255.h"

@implementation CircleModel

#pragma mark - Singleton method
+(id)sharedInstance
{
    static dispatch_once_t pred;
    static id shared = nil;

    dispatch_once(&pred, ^{
        shared = [[[self class] alloc] init];
    });
    return shared;
}

-(id)init
{
    self = [super init];
    if (self)
    {
        _items = [NSArray arrayWithObjects:
                  [[CircleVO alloc] initWithPos:CGPointMake(172 ,-
376)
                                          radius:509
                                           color:[UIColor
colorWith255Red:47.0 green:69.0 blue:103.0 alpha:191.0]
                                       darkColor:[UIColor
colorWith255Red:47.0 green:69.0 blue:103.0 alpha:255.0]
                   ] ,
                  [[CircleVO alloc] initWithPos:CGPointMake(197 ,-
566)
                                          radius:764
                                           color:[UIColor
colorWith255Red:84.0 green:112.0 blue:145.0 alpha:191.0]
                                       darkColor:[UIColor
colorWith255Red:66.0 green:97.0 blue:133.0 alpha:255.0]
                   ] ,
                  [[CircleVO alloc] initWithPos:CGPointMake(207 ,-
754)
                                          radius:1018
                                           color:[UIColor
colorWith255Red:14.0 green:60.0 blue:42.0 alpha:191.0]
                                       darkColor:[UIColor
colorWith255Red:14.0 green:60.0 blue:42.0 alpha:255.0]
                   ] ,
                  [[CircleVO alloc] initWithPos:CGPointMake(215 ,-
943)
                                          radius:1273
                                           color:[UIColor
colorWith255Red:88.0 green:138.0 blue:115.0 alpha:191.0]
                                       darkColor:[UIColor
colorWith255Red:57.0 green:116.0 blue:88.0 alpha:255.0]
                   ] ,
                  [[CircleVO alloc] initWithPos:CGPointMake(221 ,-
1132)
                                          radius:1527
                                           color:[UIColor
colorWith255Red:136.0 green:143.0 blue:24.0 alpha:191.0]
                                       darkColor:[UIColor
colorWith255Red:121.0 green:131.0 blue:0.0 alpha:255.0]
                   ] ,
                  [[CircleVO alloc] initWithPos:CGPointMake(227 ,-
1321)
                                          radius:1782
                                           color:[UIColor
colorWith255Red:202.0 green:80.0 blue:51.0 alpha:191.0]
                                       darkColor:[UIColor
colorWith255Red:191.0 green:70.0 blue:39.0 alpha:255.0]
                   ] ,
                  [[CircleVO alloc] initWithPos:CGPointMake(232 ,-
1510)
                                          radius:2036
                                           color:[UIColor
colorWith255Red:156.0 green:23.0 blue:0.0 alpha:191.0]
                                       darkColor:[UIColor
colorWith255Red:156.0 green:23.0 blue:0.0 alpha:255.0]
                   ] ,
                  nil];
    }
    return self;
}

-(NSArray*)items
{
    return _items;
}

@end

//
//  ChapterModel.h
//  NTFExplore
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <Foundation/Foundation.h>
#import "ChapterVO.h"
#import "KeywordVO.h"

@interface ChapterModel : NSObject

@property (atomic , strong) NSMutableArray* items;

+(id)sharedInstance;
+(KeywordVO*)getRandomKeywordVO;

-(ChapterVO*)chapterAfter:(ChapterVO*)currentChapter;

@end

//
//  ChapterModel.m
//  NTFExplore
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "ChapterModel.h"
#import "ChapterVO.h"
#import "KeywordModel.h"
#import "notifications.h"

@implementation ChapterModel


@synthesize items = _items;

#pragma mark - Singleton method
```

```
+(id)sharedInstance
{
    static dispatch_once_t pred;
    static id shared = nil;

    dispatch_once(&pred, ^{
        shared = [[[self class] alloc] init];
    });
    return shared;
}

- (id)init
{
    self = [super init];
    if (self)
    {
        _items = [[NSMutableArray alloc] init];
    }
    return self;
}

-(ChapterVO*)chapterAfter:(ChapterVO*)currentChapter
{
    NSInteger currentChapterIndex= [_items
indexOfObject:currentChapter];
    if (currentChapterIndex < _items.count - 1)
    {
        return [_items objectAtIndex:currentChapterIndex + 1];
    }
    return [_items objectAtIndex:0];
}


#pragma mark - Dev methods

+(KeywordVO*)getRandomKeywordVO
{
    NSUInteger randomIndex = arc4random() % [[[KeywordModel
sharedModel] allKeywords] count];
    return [[[KeywordModel sharedModel] allKeywords]
objectAtIndex:randomIndex];
}


#pragma mark - properties

-(void)setItems:(NSMutableArray *)items
{
    _items = items;
    [[NSNotificationCenter defaultCenter]
postNotificationName:CHAPTERS_LOADED object:self];
}

-(NSMutableArray*)items
{
    return _items;
}

@end

//
//  ChapterVO.h
//  NTFExplore
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <Foundation/Foundation.h>

@interface ChapterVO : NSObject

@property (nonatomic , copy) NSString *objectId;
@property (nonatomic) NSDate *createdAt;
@property (nonatomic) NSMutableArray* keywords;
@property (nonatomic) NSMutableArray* imageIds;
@property (nonatomic) NSUInteger numViews;
@property (nonatomic) NSUInteger firstCategory;
@property (nonatomic) NSString *username;
@property (nonatomic) NSArray *layout;
@property (nonatomic) NSString *location_en;
@property (nonatomic) NSString *location_it;
@property (nonatomic, readonly) NSString
*locationForCurrentLanguage;
@property (nonatomic) CGRect rect;
@property (nonatomic) NSUInteger index;

-(id)initWithId:(NSString*)theChapterId imageIds:
(NSMutableArray*)imageIds keywords:(NSMutableArray*)theKeywords
username:(NSString*)theUsername;

@end

//
//  ChapterVO.m
//  NTFExplore
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "ChapterVO.h"
```

```
@implementation ChapterVO

-(id)initWithId:(NSString*)theChapterId imageIds:
(NSMutableArray*)imageIds keywords:(NSMutableArray*)theKeywords
username:(NSString*)theUsername
{
    self = [super init];
    if (self)
    {
        _objectId = theChapterId;
        _imageIds = imageIds;
        _keywords = theKeywords;
        _username = theUsername;
    }
    return self;
}

-(NSString*)locationForCurrentLanguage
{
    if ([[[NSLocale preferredLanguages] objectAtIndex:0]
isEqualToString:@"it"])
    {
        return _location_it;
    }
    return _location_en;
}

@end

//
//  CommentVO.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <Foundation/Foundation.h>

@interface CommentVO : NSObject

@property (nonatomic , copy) NSString *objectId;
@property (nonatomic) NSDate *createdAt;
@property (nonatomic , copy) NSString *text;
@property (nonatomic , copy) NSString *username;
@property (nonatomic, strong) NSString *location_en;
@property (nonatomic, strong) NSString *location_it;
@property (nonatomic, readonly) NSString
*locationForCurrentLanguage;
```

```
@end

//
//  CommentVO.m
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "CommentVO.h"

@implementation CommentVO

-(NSString*)locationForCurrentLanguage
{
    if ([[[NSLocale preferredLanguages] objectAtIndex:0] 
isEqualToString:@"it"])
    {
        return _location_it;
    }
    return _location_en;
}

@end

//
//  ImageModel.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <Foundation/Foundation.h>
#import "ImageVO.h"
#import "ImageVO.h"

@interface ImageModel : NSObject

@property (nonatomic, readonly) NSArray *images;

+(id)sharedModel;

-(NSArray*)getImagesWithKeywords:(NSArray*)keywords;
-(NSArray*)getImagesWithKeywords:(NSArray*)keywords limit:
(NSInteger)limit;
-(ImageVO*)getImageById:(NSNumber*)imageId;
-(NSArray*)getImagesWithIds:(NSArray*)imageIds;

@end
```

```
//
//  ImageModel.m
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "ImageModel.h"
#import "NTFDataClient.h"
#import "KeywordModel.h"

@implementation ImageModel

+(id)sharedModel
{
    static dispatch_once_t pred;
    static id shared = nil;

    dispatch_once(&pred, ^{
        shared = [[[self class] alloc] init];
    });
    return shared;
}

- (id)init
{
    self = [super init];
    if (self)
    {
        NSString *path = [[NSBundle mainBundle]
pathForResource:@"images" ofType:@"json"];
        NSError *error = nil;
        NSArray *objects = [NSJSONSerialization JSONObjectWithData:
[NSData dataWithContentsOfFile:path]

options:NSJSONReadingAllowFragments

error:&error];
        if (error)
        {
            NSLog(@"Error: %@" , error);
            abort();
        }

        NSMutableArray *tempArray = [[NSMutableArray alloc] init];
        for (uint i=0; i<objects.count; i++)
        {
            NSDictionary *d = [objects objectAtIndex:i];
            ImageVO *vo = [[ImageVO alloc] initWithId:[NSNumber
numberWithInt:i]
```

```
                                              filename:d[@"filename"]
                                              keywords:[[KeywordModel
sharedModel] keywordsWithIds:d[@"keywords"]]];
            [tempArray addObject:vo];
        }
        _images = [NSArray arrayWithArray:tempArray];
    }
    return self;
}

- (NSArray*)getImagesWithKeywords:(NSArray*)keywords
{
    __block NSMutableArray *relevantVOs = [[NSMutableArray alloc]
init];
    [_images enumerateObjectsUsingBlock:^(id obj, NSUInteger idx,
BOOL *stop) {
        ImageVO *vo = (ImageVO*)obj;
        if ([vo relevanceToTagSet:keywords] > 0) [relevantVOs
addObject:vo];
    }];

    NSMutableArray *sortedImages = [[relevantVOs
sortedArrayUsingComparator:^NSComparisonResult(id obj1, id obj2) {
        int numViews1 = [(ImageVO*)obj1 numViews];
        int numViews2 = [(ImageVO*)obj2 numViews];

        if (numViews1 > numViews2) {
            return (NSComparisonResult)NSOrderedDescending;
        }

        if (numViews1 < numViews2) {
            return (NSComparisonResult)NSOrderedAscending;
        }
        return (NSComparisonResult)NSOrderedSame;
    }] mutableCopy];

    // grab 1 photo per keyword
    NSMutableArray *keywordsCopy = [keywords mutableCopy];
    NSMutableArray *results = [NSMutableArray array];
    for (ImageVO *imageVO in sortedImages)
    {
        for (KeywordVO *keywordVO in keywordsCopy)
        {
            if ([imageVO.keywords containsObject:keywordVO])
            {
                [results addObject:imageVO];
                [keywordsCopy removeObject:keywordVO];
                break;
            }
        }
        if (keywordsCopy.count == 0) break;
    }
    NSAssert(keywordsCopy.count == 0, @"not all keywords have an
associated image");

    // remove already selected images from sortedImages
    [sortedImages removeObjectsInArray:results];
    // and bung them on the end of the results list
    [results addObjectsFromArray:sortedImages];

    return results;
}

-(NSArray*)getImagesWithKeywords:(NSArray*)keywords limit:
(NSInteger)limit
{
    NSArray *a = [self getImagesWithKeywords:keywords];
    if (a.count <= limit) return a;

    return [a subarrayWithRange:NSMakeRange(0, limit)];
}

-(ImageVO*)getImageById:(NSNumber*)imageId
{
    for (ImageVO *vo in _images)
    {
        if (imageId == vo.objectId) return vo;
    }
    return nil;
}

-(NSArray*)getImagesWithIds:(NSArray*)imageIds
{
    NSMutableArray *out = [[NSMutableArray alloc] init];
    [_images enumerateObjectsUsingBlock:^(id obj, NSUInteger idx,
BOOL *stop) {
        ImageVO *vo = (ImageVO*)obj;
        if ([imageIds containsObject:vo.objectId]) [out
addObject:vo];
    }];
    return out;
}


@end

//
//  ImageVO.h
//  NostalgiaTiFrega
//
```

```
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <Foundation/Foundation.h>

@interface ImageVO : NSObject

@property (readonly) NSNumber *objectId;
@property (nonatomic , copy , readonly) NSString *fileName;
@property (nonatomic , readonly) NSArray *keywords;
@property (nonatomic) NSMutableArray* comments;
@property (nonatomic) BOOL hasComments;
@property NSUInteger numViews;

- (id)initWithId:(NSNumber*)objectId filename:(NSString*)filename
keywords:(NSArray*)keywords;
- (NSInteger)relevanceToTagSet:(NSArray*)tagSet;
- (UIImage*)image;

@end

//
//  ImageVO.m
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "ImageVO.h"
#import "KeywordVO.h"

@implementation ImageVO

- (id)initWithId:(NSNumber*)objectId filename:(NSString*)filename
keywords:(NSArray*)keywords
{
    self = [super init];
    if (self)
    {
        _objectId = objectId;
        _fileName = filename;
        _keywords = keywords;
        _comments = [[NSMutableArray alloc] init];
    }
    return self;
}

-(NSString *)description
{
    return [NSString stringWithFormat:@"ImageVO: objectId=%@
fileName='%@' keywords=%@ numViews=%i" , _objectId , _fileName,
_keywords, _numViews];
}

-(NSInteger)relevanceToTagSet:(NSArray *)tagSet
{
    NSMutableSet *s = [NSMutableSet setWithArray:tagSet];
    [s intersectSet:[NSSet setWithArray:_keywords]];

    return [s count];
}

-(UIImage*)image
{
    return [UIImage imageNamed:_fileName];
}


@end

//
//  MainViewController.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "CKFadingNavigationController.h"

@interface MainViewController : CKFadingNavigationController

@end

//
//  MainViewController.m
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Dragoa. All rights reserved.
//

#import "MainViewController.h"
#import "ExploreViewController.h"
#import "CurateViewController.h"
#import "notifications.h"

@implementation MainViewController

- (void)didReceiveMemoryWarning
{
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Release any cached data, images, etc that aren't in use.
}


#pragma mark - View lifecycle

- (void)viewDidLoad
{
    [super viewDidLoad];
    [self setNavigationBarHidden:YES];
}

- (void)viewDidUnload
{
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}


#pragma mark - rotation

- (BOOL)shouldAutorotateToInterfaceOrientation:
(UIInterfaceOrientation)interfaceOrientation
{
      return
UIInterfaceOrientationIsLandscape(interfaceOrientation);
}

-(NSUInteger)supportedInterfaceOrientations
{
    return UIInterfaceOrientationMaskLandscape;
}

@end

//
//  InfoView.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface InfoView : UIView
```

```
@property (nonatomic, readonly) UIButton *closeBtn;
@property (nonatomic, readonly) UIWebView *webView;

@end

//
//  InfoView.m
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "InfoView.h"
#import <QuartzCore/QuartzCore.h>
#import "UIView+Position.h"
#import "UIFont+NTF.h"
#import "UIColor+Color255.h"
#import "UIButton+NTF.h"

@implementation InfoView

- (id)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];
    if (self)
    {
        [self setBackgroundColor:[UIColor colorWithWhite:0.0
alpha:0.2]];

        _webView = [[UIWebView alloc]
initWithFrame:CGRectMake(182.0f, 0.0f, 668.0f, frame.size.height)];
        NSString *htmlFile = [[NSBundle mainBundle]
pathForResource:@"info" ofType:@"html"];
        NSString* htmlString = [NSString
stringWithContentsOfFile:htmlFile encoding:NSUTF8StringEncoding
error:nil];
        [_webView loadHTMLString:htmlString baseURL:nil];
        UIScrollView *sv = [_webView.subviews objectAtIndex:0];
        [sv setIndicatorStyle:UIScrollViewIndicatorStyleWhite];
        [_webView setBackgroundColor:[UIColor
colorWithHexRGB:0xddd6ca]];
        [_webView setOpaque:NO];

        for(UIView *wview in [[[_webView subviews] objectAtIndex:0]
subviews])
        {
            if([wview isKindOfClass:[UIImageView class]])
            {
                wview.hidden = YES;
            }
        }

        _closeBtn = [UIButton infoButton];
        [_closeBtn setPosition:CGPointMake(27.0f, 657.0f)];

        [self addSubview:_webView];
        [self addSubview:_closeBtn];
        [self setNeedsLayout];
    }
    return self;
}

-(void)layoutSubviews
{

}

@end

//
//  InfoViewController.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "InfoView.h"

@interface InfoViewController : UIViewController <UIWebViewDelegate>

@property (nonatomic, readonly, weak) InfoView *infoView;

@end

//
//  InfoViewController.m
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "InfoViewController.h"
#import "UIImage+ImageWithUIView.h"
#import "UIView+Position.h"
#import "UIScreen+CKAdditions.h"

@interface InfoViewController ()

@end
```

```
@implementation InfoViewController
{
    UITapGestureRecognizer *_tgr;
}

- (id)init
{
    return [super initWithNibName:nil bundle:nil];
}

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle
*)nibBundleOrNil
{
    return [self init];
}

-(void)loadView
{
    [self setView:[[InfoView alloc] initWithFrame:[[UIScreen
mainScreen] rectWithOrientation:[self interfaceOrientation]]]];
    _infoView = (InfoView*)self.view;
    [_infoView.closeBtn addTarget:self
action:@selector(closeBtnTouch:)
forControlEvents:UIControlEventTouchUpInside];
    [_infoView.webView setDelegate:self];

    _tgr = [[UITapGestureRecognizer alloc] initWithTarget:self
action:@selector(viewTapped:)];
    [_infoView addGestureRecognizer:_tgr];
}

- (void)viewDidLoad
{
    [super viewDidLoad];
      // Do any additional setup after loading the view.
}

// iOS <= 5
- (BOOL)shouldAutorotateToInterfaceOrientation:
(UIInterfaceOrientation)interfaceOrientation
{
    return UIInterfaceOrientationIsLandscape(interfaceOrientation);
}
// iOS >= 6
-(NSUInteger)supportedInterfaceOrientations
{
    return UIInterfaceOrientationMaskLandscape;
}
```

```
- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
}


#pragma mark - event handlers

-(void)closeBtnTouch:(id)sender
{
    [UIView animateWithDuration:0.5 delay:0
options:UIViewAnimationOptionCurveEaseOut animations:^{
        self.view.alpha = 0.0;
    } completion:^(BOOL finished) {
        [self.view removeFromSuperview];
        [self removeFromParentViewController];
    }];
}

-(void)viewTapped:(UITapGestureRecognizer*)tgr
{
    NSLog(@"%@: %@" , [self class] , NSStringFromSelector(_cmd));

    CGPoint loc = [tgr locationInView:self.view];

    NSLog(@"loc = %@" , NSStringFromCGPoint(loc));

    if (loc.x < _infoView.webView.x || loc.x >
CGRectGetMaxX(_infoView.webView.frame))
    {
        [UIView animateWithDuration:0.5 delay:0
options:UIViewAnimationOptionCurveEaseOut animations:^{
            self.view.alpha = 0.0;
        } completion:^(BOOL finished) {
            [self.view removeFromSuperview];
            [self removeFromParentViewController];
        }];
    }
}

#pragma mark - UIWebViewDelegate

- (BOOL)webView:(UIWebView *)webView shouldStartLoadWithRequest:
(NSURLRequest *)request navigationType:
(UIWebViewNavigationType)navigationType
{
    if ([[[request URL] scheme] isEqual:@"mailto"])
    {
        [[UIApplication sharedApplication] openURL:[request URL]];
        return NO;
    }
    return YES;
}

@end

//
//  ContentView.h
//  Tiles
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface MapBg : UIView

@end

//
//  ContentView.m
//  Tiles
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "MapBg.h"
#import <QuartzCore/QuartzCore.h>

const CGSize kTileSize = {256.0 , 256.0};

@implementation MapBg

- (id)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];
    if (self)
    {
        self.opaque = YES;
        self.backgroundColor = [UIColor colorWithRed:249.0/255.0 
green:240.0/255.0 blue:232.0/255.0 alpha:1.0];
        [[self tiledLayer] setLevelsOfDetail:4];
        self.layer.opaque = YES;
    }
    return self;
}

-(void)drawRect:(CGRect)rect
{
    CGContextRef c = UIGraphicsGetCurrentContext();
    CGFloat scale = CGContextGetCTM(c).a;
    NSRange r = NSMakeRange(0, 4);
    NSString *s = [[NSString stringWithFormat:@"%f" , scale]
substringWithRange:r];
    int col = (CGRectGetMinX(rect) * scale) / kTileSize.width;
    int row = (CGRectGetMinY(rect) * scale) / kTileSize.height;
    NSString *fileName = [NSString stringWithFormat:@"map_s%@_c%i_r
%i.png" , s , col , row];
    UIImage *im = [UIImage imageNamed:fileName];
    [im drawInRect:rect];
}

- (void)didMoveToWindow
{
    self.contentScaleFactor = 1.0;
}

-(CATiledLayer*)tiledLayer
{
    return (CATiledLayer*)self.layer;
}

+(Class)layerClass
{
    return [CATiledLayer class];
}

@end

//
//  MainViewController.h
//  NTFExplore
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "ChapterIcon.h"
#import "ChapterVO.h"
#import "ExploreView.h"
#import "ChapterIconDelegate.h"
#import "ChapterViewController.h"
#import "WelcomeView.h"
#import "WelcomeViewDelegate.h"
#import "IconInfoView.h"

#define SETTINGS_BUTTON_TOUCH @"settingsButtonTouch"

@interface ExploreViewController : UIViewController
<UIScrollViewDelegate , UITextFieldDelegate , ChapterIconDelegate,
WelcomeViewDelegate>

@property (nonatomic, readonly, weak) ExploreView *exploreView;
@property (nonatomic, readonly) NSArray *allIconPositions;
@property (nonatomic, readonly) ChapterViewController
*chapterViewController;
@property (nonatomic, readonly) NSMutableDictionary *chapterIcons;
@property (nonatomic, readonly, strong) WelcomeView *welcomeView;
@property (nonatomic, strong) IconInfoView *iconInfoView;

-(void)showWelcomeView;

@end

//
//  MainViewController.m
//  NTFExplore
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "ExploreViewController.h"
#import "ChapterModel.h"
#import "UIView+Position.h"
#import "notifications.h"
#import "InfoViewController.h"
#import "UIImage+ImageWithUIView.h"
#import "UIScreen+CKAdditions.h"
#import "MBProgressHUD.h"
#import "constants.h"
#import "IconMaker.h"
#import "NTFServices.h"

@implementation ExploreViewController
{
    UITapGestureRecognizer *_tapGR;
    ChapterVO *_selectedVO;
}

@synthesize chapterViewController = _chapterViewController;
@synthesize chapterIcons = _chapterIcons;


#pragma mark - UIViewController

- (id)init
{
    self = [super initWithNibName:nil bundle:nil];
    if (self)
    {
        NSString *pointsFile = [[NSBundle mainBundle] 
pathForResource:@"NTFChapterPositions" ofType:@"json"];
        NSError *error = nil;
        NSArray *a = [NSJSONSerialization JSONObjectWithData:[NSData 
dataWithContentsOfFile:pointsFile]

options:NSJSONReadingAllowFragments

error:&error];
        if (error)
        {
            NSLog(@"Error: %@" , error);
            abort();
        }

        NSMutableArray *temp = [[NSMutableArray alloc] init];
        NSDictionary *rawPoint = nil;
        for (uint i=0; i<a.count; i++)
        {
            rawPoint = [a objectAtIndex:i];
            CGPoint p = CGPointMake([[rawPoint objectForKey:@"x"] 
floatValue], [[rawPoint objectForKey:@"y"] floatValue]);
            [temp addObject:[NSValue valueWithCGPoint:p]];
        }
        _allIconPositions = [NSArray arrayWithArray:temp];

        // show welcome screen if this is the first launch
        if (![[NSUserDefaults standardUserDefaults] 
objectForKey:USER_WAS_WELCOMED])
        {
            [self showWelcomeView];
            [[NSUserDefaults standardUserDefaults] setObject:@YES 
forKey:USER_WAS_WELCOMED];
            [[NSUserDefaults standardUserDefaults] synchronize];
        }


        [[NSNotificationCenter defaultCenter] addObserver:self 
selector:@selector(updateChapterIcons) name:CHAPTERS_LOADED 
object:nil];
        [[NSNotificationCenter defaultCenter] addObserver:self 
selector:@selector(updateChapterIcons) name:CHAPTERS_UPDATED 
object:nil];
        [[NSNotificationCenter defaultCenter] addObserver:self 
selector:@selector(handleChapterSavedNotification:) 
name:CHAPTER_SAVED object:nil];
        [[NSNotificationCenter defaultCenter] addObserver:self 
selector:@selector(handleChapterUpdatedNotification:) 
name:CHAPTER_UPDATED object:nil];
    }
    return self;
}

-(id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle
*)nibBundleOrNil
{
    return [self init];
}

-(void)loadView
{
    [self setView:[[ExploreView alloc] initWithFrame:[[UIScreen
mainScreen] landscapeRect]]];

    _exploreView = (ExploreView*)self.view;
    [_exploreView.scrollView setDelegate:self];
    [_exploreView.curateBtn addTarget:self
action:@selector(onCurateButtonTouch:)
forControlEvents:UIControlEventTouchUpInside];
    [_exploreView.infoBtn addTarget:self
action:@selector(onInfoButtonTouch:)
forControlEvents:UIControlEventTouchUpInside];

    _tapGR = [[UITapGestureRecognizer alloc] initWithTarget:self
action:@selector(mapTapped:)];
    [_exploreView.mapView addGestureRecognizer:_tapGR];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];

    if (!self.view.window)
    {
        _chapterViewController = nil;
        _chapterIcons = nil;
    }
}


#pragma mark - View lifecycle

- (void)viewDidLoad
{
    [super viewDidLoad];
    if ([[ChapterModel sharedInstance] items] && [[[ChapterModel
sharedInstance] items] count] > 0)
    {
        [self updateChapterIcons];
    }
    else
    {
        [MBProgressHUD showHUDAddedTo:self.view animated:YES];
    }
}

-(void)viewWillAppear:(BOOL)animated
{
    [super viewWillAppear:animated];
    for (NSString *key in self.chapterIcons)
    {
        ChapterIcon *icon = self.chapterIcons[key];
        [icon update];
    }
}

- (void)viewDidUnload
{
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:
(UIInterfaceOrientation)interfaceOrientation
{
      return UIInterfaceOrientationIsPortrait(interfaceOrientation);
}


#pragma mark - ExploreViewController

-(void)handleChapterSavedNotification:(NSNotification*)notification
{
    [MBProgressHUD hideAllHUDsForView:self.view animated:YES];

    __weak ChapterVO *vo = notification.userInfo[VO];
    [self updateIconRects];
    [self centreToVO:vo];
    [self updateIconLayout];

    ChapterIcon *ci = self.chapterIcons[vo.objectId];

    NSAssert(ci != nil, @"ChapterIcon is nil");

    __weak ChapterIcon *weakChapterIcon = ci;
    [IconMaker updateIconImageForVO:vo completion:^(UIImage *image)
{
        weakChapterIcon.vo = vo;
    }];
    [self selectChapterIcon:ci];

}

-(void)handleChapterUpdatedNotification:
(NSNotification*)notification
{
    ChapterVO *vo = notification.userInfo[VO];
    if(vo.numViews < 11)
        [IconMaker updateIconImageForVO:vo completion:NULL];
    [self updateIconRects];
    [self updateIconLayout];
}

-(void)updateChapterIcons
{
    [MBProgressHUD hideAllHUDsForView:self.view animated:YES];

//    _iconInfoView.hidden = YES;

    [self updateIconRects];
    [self updateIconLayout];
}

-(void)updateIconRects
{
    NSArray* chapters = [[ChapterModel sharedInstance] items];
    ChapterVO *vo = nil;
    for (uint i=0; i<chapters.count; i++)
    {
        vo = chapters[i];
        CGSize size = [ChapterIcon sizeForVO:vo];
        CGPoint center = [[_allIconPositions objectAtIndex:vo.index
% MAX_CHAPTERS] CGPointValue];
        vo.rect = CGRectIntegral(CGRectMake(center.x -
size.width/2.0, center.y - size.width/2.0, size.width,
size.height));
    }
}

-(void)updateIconLayout
{
    CGRect f;
    UIScrollView *scrollView = _exploreView.scrollView;
    float zoomScale = scrollView.zoomScale;
    CGPoint p = scrollView.contentOffset;
    f.origin.x = p.x/zoomScale;
    f.origin.y = p.y/zoomScale;
    f.size.width = scrollView.bounds.size.width/zoomScale;
    f.size.height = scrollView.bounds.size.height/zoomScale;
    ChapterIcon *icon = nil;
    NSArray *chapters = [[ChapterModel sharedInstance] items];
    for (ChapterVO *vo in chapters)
    {
        icon = self.chapterIcons[vo.objectId];
        if (CGRectIntersectsRect(vo.rect, f))
        {
            if (!icon)
            {
                // add it
                icon = [_exploreView dequeueChapterIcon];
                if (!icon)
                {
                    icon = [[ChapterIcon alloc] init];
                }
                icon.vo = vo;
                if (_selectedVO == vo) icon.selected = YES;
                icon.frame = vo.rect;
                icon.delegate = self;
                [_exploreView.mapView addSubview:icon];
                self.chapterIcons[vo.objectId] = icon;
            }
            else
            {
                icon.vo = vo;
            }
        }
        else if (icon)
        {
            [self.chapterIcons removeObjectForKey:vo.objectId];
            [icon removeFromSuperview];
            [_exploreView enqueueChapterIcon:icon];
        }
    }
}

-(void)onCurateButtonTouch:(id)sender
{
    [[NSNotificationCenter defaultCenter]
postNotificationName:NAVIGATE_TO_CURATE
                                                         object:self
                                                       userInfo:nil];
}

-(void)onInfoButtonTouch:(id)sender
{
    InfoViewController *ivc = [[InfoViewController alloc] init];
    [self addChildViewController:ivc];
    ivc.view.alpha = 0.0;
    [self.view addSubview:ivc.view];

    [UIView animateWithDuration:0.5 delay:0
options:UIViewAnimationOptionCurveEaseOut animations:^{
        ivc.view.alpha = 1.0;
    } completion:^(BOOL finished) {

    }];
}

-(void)mapTapped:(UITapGestureRecognizer*)gr
{
    ChapterIcon *ci = nil;
    for (NSString *key in self.chapterIcons)
    {
        ci = self.chapterIcons[key];
        if (CGRectContainsPoint(ci.frame, [gr
locationInView:_exploreView.mapView]))
        {
            [self chapterIconTouched:ci];
            return;
        }
    }

    [self deselectAllChapterIcons];
    [self hideInfoView];
}

-(void)hideInfoView
{
    _iconInfoView.hidden = YES;
    _iconInfoView.vo = nil;
}


#pragma mark - UIScrollViewDelegate

-(UIView *)viewForZoomingInScrollView:(UIScrollView *)scrollView
{
    return _exploreView.mapView;
}

-(void)scrollViewDidScroll:(UIScrollView *)scrollView
{
    [self updateIconLayout];
}

#pragma mark - ChapterIconDelegate

-(void)chapterIconTouched:(ChapterIcon *)icon
```

```
{

    if (_iconInfoView && _iconInfoView.vo == icon.vo)
    {
        [[NSNotificationCenter defaultCenter]
postNotificationName:NAVIGATE_TO_CHAPTER

object:self

userInfo:@{VO:icon.vo}];
        return;
    }
    _selectedVO = icon.vo;
    [self selectChapterIcon:icon];
}

-(void)selectChapterIcon:(ChapterIcon *)icon
{
    for (NSString *key in self.chapterIcons)
    {
        ChapterIcon *ci = self.chapterIcons[key];
        ci.selected = (ci == icon);
    }

    if (!_iconInfoView)
    {
        _iconInfoView = [[IconInfoView alloc] init];
    }
    _iconInfoView.vo = icon.vo;
    _iconInfoView.x = icon.x + icon.frame.size.width/2 -
_iconInfoView.frame.size.width/2.0f;
    _iconInfoView.y = icon.y - _iconInfoView.frame.size.height -
10.0f;
    _iconInfoView.hidden = NO;

    [self.exploreView.mapView addSubview:_iconInfoView];
}

-(void)deselectAllChapterIcons
{
    _selectedVO = nil;
    for (NSString *key in self.chapterIcons)
    {
        ChapterIcon *ci = self.chapterIcons[key];
        ci.selected = NO;
    }
}

-(void)centreToVO:(ChapterVO*)vo
{
    CGPoint c = self.view.center;
    float scale = self.exploreView.scrollView.zoomScale;
    CGPoint iconPos = CGPointMake(CGRectGetMidX(vo.rect) * scale,
CGRectGetMidY(vo.rect) * scale);
    CGSize contentSize = self.exploreView.scrollView.contentSize;
    CGSize scrollSize = self.exploreView.scrollView.bounds.size;
    CGPoint maxContentOffset = CGPointMake(contentSize.width/scale -
scrollSize.width, contentSize.height/scale - scrollSize.height);
    CGPoint p;
    p.x = MIN(MAX(iconPos.x - c.x , 0.0) , maxContentOffset.x);
    p.y = MIN(MAX(iconPos.y - c.y , 0.0) , maxContentOffset.y);
    [self.exploreView.scrollView setContentOffset:p animated:NO];

}

-(ChapterViewController*)chapterViewController
{
    if (!_chapterViewController)
    {
        _chapterViewController = [[ChapterViewController alloc]
init];
    }
    return _chapterViewController;
}

-(NSMutableDictionary*)chapterIcons
{
    if (!_chapterIcons)
    {
        _chapterIcons = [[NSMutableDictionary alloc] init];
    }
    return _chapterIcons;
}


#pragma mark - WelcomeView

-(void)showWelcomeView
{
    NSArray *nibObjects = [[NSBundle mainBundle]
loadNibNamed:@"WelcomeView" owner:self options:nil];
    _welcomeView = nibObjects[0];
    _welcomeView.delegate = self;
    [self.view addSubview:_welcomeView];
}


#pragma mark WelcomeViewDelegate

-(void)closeWelcomeView:(UIView *)welcomeView
```

```
{
    [UIView animateWithDuration:0.5 animations:^{
        _welcomeView.alpha = 0.0f;
    } completion:^(BOOL finished) {
        [_welcomeView removeFromSuperview];
        _welcomeView = nil;
    }];
}
@end

//
//  ExploreView.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "ChapterIcon.h"
#import "MapBg.h"

@interface ExploreView : UIView

@property (nonatomic, readonly) UIScrollView *scrollView;
@property (nonatomic, readonly) UIView *mapView;
@property (nonatomic, readonly) UIButton *curateBtn;
@property (nonatomic, readonly) UIButton *infoBtn;

-(void)enqueueChapterIcon:(ChapterIcon*)icon;
-(ChapterIcon*)dequeueChapterIcon;

@end

//
//  ExploreView.m
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "ExploreView.h"
#import <QuartzCore/QuartzCore.h>
#import "UIButton+NTF.h"
#import "UIView+Position.h"

@implementation ExploreView
{
    NSMutableSet *_queue;
}
```

```
- (id)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];
    if (self)
    {
        _queue = [[NSMutableSet alloc] init];

        _mapView = [[MapBg alloc] initWithFrame:CGRectMake(0, 0,
6144, 4608)];
        [_mapView setUserInteractionEnabled:YES];

        _scrollView = [[UIScrollView alloc] initWithFrame:frame];
        [_scrollView setOpaque:YES];
        [_scrollView setBackgroundColor:_mapView.backgroundColor];
        [_scrollView setZoomScale:1];
        [_scrollView setMaximumZoomScale:1.0];
        [_scrollView setMinimumZoomScale:0.33];
        [_scrollView setContentSize:_mapView.frame.size];
        [_scrollView addSubview:_mapView];
        [_scrollView setDelaysContentTouches:NO];
        [_scrollView setBounces:NO];
        [_scrollView setBouncesZoom:NO];

        CGPoint offset = _mapView.center;
        offset.x -= 512.0;
        offset.y -= 384.0;
        [_scrollView setContentOffset:offset];

        _curateBtn = [UIButton curateButton];
        [_curateBtn setPosition:CGPointMake(915.0, 657.0)];

        _infoBtn = [UIButton infoButton];
        [_infoBtn setPosition:CGPointMake(27.0,
_curateBtn.frame.origin.y)];

        [self addSubview:_scrollView];
        [self addSubview:_curateBtn];
        [self addSubview:_infoBtn];
    }
    return self;
}

-(void)enqueueChapterIcon:(ChapterIcon*)icon
{
    [icon prepareForReuse];
    [_queue addObject:icon];
}

-(ChapterIcon *)dequeueChapterIcon
{
    ChapterIcon *ci = [_queue anyObject];

    if (!ci) return nil;

    [_queue removeObject:ci];
    return ci;
}

@end

//
//  ChapterIcon.h
//  NTFExplore
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "ChapterVO.h"
#import "PetalLayer.h"
#import "ChapterIconDelegate.h"

@interface ChapterIcon : UIView

@property (nonatomic, readonly) BOOL visited;
@property (nonatomic) BOOL selected;
@property (nonatomic) id<ChapterIconDelegate> delegate;
@property (nonatomic) ChapterVO *vo;
@property (nonatomic, readonly) CAShapeLayer *bgLayer;
@property (nonatomic, readonly) CALayer *petalLayer;
@property (nonatomic, readonly) NSUInteger petalType;
@property (nonatomic, weak) UIImage *iconImage;

+(CGSize)sizeForVO:(ChapterVO*)vo;

-(id)initWithVO:(ChapterVO*)theVO;
-(UIColor*)getColourByCategoryId:(int)catId;
-(void)update;
-(void)prepareForReuse;

@end
//
//  ChapterIcon.m
//  NTFExplore
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "ChapterIcon.h"
#import "QuartzCore/QuartzCore.h"
```

```
#import "UIView+Rotation.h"
#import "CircleModel.h"
#import "CircleVO.h"
#import "randomFloat.h"
#import "KeywordModel.h"
#import "PetalLayer.h"
#import "constants.h"
#import "IconMaker.h"

static const float kIconDiameter = 101.0f;
static const float kShadowOpacitySelected = 0.75f;
static const float kShadowRadiusSelected = 7.5f;
static const float kShadowOpacityUnselected = 0.2f;
static const float kShadowRadiusUnselected = 5.0f;
static const float kOpacityVisited = 0.75f;
static const float kOpacityNormal = 1.0f;

@interface ChapterIcon ()

@end


@implementation ChapterIcon
{
    UIImageView *_iconImageView;
}

@synthesize selected = _selected;
@synthesize vo = _vo;

+(CGSize)sizeForVO:(ChapterVO*)vo
{
    float scale = 0.5 + (vo.numViews / 20.0);
    if (scale > 1.0) scale = 1.0;
    float diameter = kIconDiameter * scale;
    return CGSizeMake(diameter, diameter);
}

-(id)initWithVO:(ChapterVO*)theVO
{
    self = [self init];
    if (self)
    {
        self.vo = theVO;
    }
    return self;
}

-(id)init
{
    self = [super init];
    if (self)
    {
        [self setup];
    }
    return self;
}

-(void)setup
{
    self.opaque = NO;
//    self.backgroundColor = [UIColor colorWithRed:1 green:0 blue:0
alpha:0.3];
    self.layer.shadowColor = [[UIColor blackColor] CGColor];
    self.layer.shouldRasterize = NO;
    self.selected = NO;

    _iconImageView = [[UIImageView alloc] init];
    [self addSubview:_iconImageView];
}

-(void)prepareForReuse
{
    self.layer.shadowOpacity = 0.0;
    _selected = NO;
    _iconImageView.image = nil;
}

-(CGSize)sizeThatFits:(CGSize)size
{
    return [ChapterIcon sizeForVO:_vo];
}

-(void)sizeToFit
{
    CGPoint c = self.center;
    [super sizeToFit];
    self.center = c;
}

-(UIColor*)getColourByCategoryId:(int)catId
{
    CircleVO* cvo = [[[CircleModel sharedInstance] items]
objectAtIndex:catId];
    return [cvo darkColor];
}


#pragma mark - properties
```

```
-(void)setSelected:(BOOL)selected
{
    _selected = selected;

    if (_selected)
    {
        self.layer.shadowOpacity = kShadowOpacitySelected;
        self.layer.shadowRadius = kShadowRadiusSelected;
    }
    else
    {
        self.layer.shadowOpacity = kShadowOpacityUnselected;
        self.layer.shadowRadius = kShadowRadiusUnselected;
    }
}

-(BOOL)selected
{
    return _selected;
}

-(void)setVisited:(BOOL)value
{
    float o = (value) ? kOpacityVisited : kOpacityNormal;
    [_petalLayer setOpacity:o];
    [self setNeedsLayout];
}

-(BOOL)visited
{
    return _petalLayer.opacity < 1.0;
}

-(void)setVo:(ChapterVO *)vo
{
    _vo = vo;
    _petalType = vo.firstCategory;
    [self update];
}

-(ChapterVO *)vo
{
    return _vo;
}

-(void)setIconImage:(UIImage *)iconImage
{
    if (iconImage == nil)
    {
        self.layer.shadowOpacity = 0.0;
        self.layer.shadowRadius = 0.0;
        return;
    }

    [self sizeToFit];
    _iconImageView.image = iconImage;
    [_iconImageView sizeToFit];

    float radius = iconImage.size.width / 2.0;
    CGPathRef path = CGPathRetain([[UIBezierPath
bezierPathWithArcCenter:CGPointMake(0, 0)

radius:radius

startAngle:0

endAngle:M_PI*2.0

clockwise:YES] CGPath]);
    self.layer.shadowPath = path;
    self.layer.shadowOffset = CGSizeMake(radius, radius);
    self.selected = _selected;
}

-(void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event
{
    [self.delegate chapterIconTouched:self];
}

- (BOOL)pointInside:(CGPoint)point withEvent:(UIEvent *)event
{
    // Return NO if even super returns NO (i.e., if point lies
outside our bounds)
    BOOL superResult = [super pointInside:point withEvent:event];
    if (!superResult) return superResult;

    CGPoint ac = CGPointMake(self.center.x - self.frame.origin.x ,
self.center.y - self.frame.origin.y);
    return pow((point.x - ac.x), 2) + pow((point.y - ac.y), 2) <
pow(self.frame.size.width/2 , 2);
}

-(void)update
{
    NSMutableArray *viewedChapters = [[NSUserDefaults
standardUserDefaults] objectForKey:VIEWED_CHAPTERS];

    if (viewedChapters != nil)
    {
        BOOL wasViewed = [viewedChapters 
indexOfObject:_vo.objectId] != NSNotFound;
        [self setVisited:wasViewed];
    }
    
    __weak ChapterIcon *weakSelf = self;
    [IconMaker getIconImageForVO:_vo completion:^(UIImage *image) {
        weakSelf.iconImage = image;
    }];
}

-(UIImage*)imageWithCALayer:(CALayer*)layer
{
    UIGraphicsBeginImageContext(layer.frame.size);
    
    [layer renderInContext:UIGraphicsGetCurrentContext()];
    UIImage* out = UIGraphicsGetImageFromCurrentImageContext();
    
    UIGraphicsEndImageContext();
    
    return out;
}

@end

//
//  PetalFactory.h
//  SVGTest
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <Foundation/Foundation.h>
#import "QuartzCore/QuartzCore.h"

typedef enum {
    PetalTypeLine ,
    PetalTypeDrumstick ,
    PetalTypeTriangle ,
    PetalTypeArrow,
    PetalTypeLeaf,
    PetalTypeDots,
    PetalTypeTeardrop
} PetalType;

@interface PetalLayer : CALayer

+(PetalLayer*)petalLayerOfType:(PetalType)type withKeywords:
(NSArray*)keywords diameter:(float)diameter;
+(PetalLayer*)monoChromePetalLayerOfType:(PetalType)type
```

```
withKeywords:(NSArray*)keywords;

@end

//
//  PetalFactory.m
//  SVGTest
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "PetalLayer.h"
#import "SvgToBezier.h"
#import "KeywordModel.h"
#import "randomFloat.h"
#import "UIColor+NTF.h"
#import "UIColor+Color255.h"

NSString* const LEAF =
@"M5.499,0C7.813,2.917,8.961,5.833,9.83,8.75c0.82,2.916,1.172,5.833,
1.17,8.75c-0.011,2.916-0.362,5.833-
1.186,8.749C8.943,29.166,7.799,32.082,5.499,35c-2.3-2.918-3.443-
5.834-4.315-8.751C0.361,23.333,0.011,20.416,0,17.5c-0.002-
2.917,0.35-5.834,1.169-8.75C2.038,5.833,3.186,2.917,5.499,0z";
NSString* const LINE =      @"M2.5,37.123c-1.381,0-2.5-1.119-2.5-
2.5V2.377c0-1.381,1.119-2.5,2.5-
2.5S5,0.996,5,2.377v32.246C5,36.004,3.881,37.123,2.5,37.123z";
NSString* const DRUMSTICK =
@"M9,4.393C9,1.967,6.988,0,4.501,0C2.015,0,0,1.967,0,4.393c0,2.085,1
.493,3.823,3.49,4.271v24.349 C3.49,33.559,3.942,34,4.501,34s1.011-
0.441,1.011-0.986V8.665C7.509,8.216,9,6.479,9,4.393z";
NSString* const TEARDROP = @"M12,5.796 C12,8.998,6,34,6,34
S0,8.998,0,5.796 C0,2.595,2.687,0,6,0 S12,2.595,12,5.796z";
NSString* const ARROW =     @"13,6.816 6.499,0 0,6.816 1.164,7.874
5.706,3.109 5.706,35 7.294,35 7.294,3.111 11.834,7.874";
//NSString* const TRIANGLE =  @"4.5,34 4.5,34 0,0 9,0";
NSString* const TRIANGLE =  @"4.5,34 0,0 9,0";

static NSCache* _pathCache;

#pragma mark - Private Methods
@interface PetalLayer ()

+(CALayer*)petalOfType:(PetalType)petalType colour:(UIColor*)colour
inRect:(CGRect)rect;
+(CALayer*)dotsWithColour:(UIColor*)colour referenceSize:
(CGSize)size;
+(CALayer*)layerWithPathAttr:(NSString*)attr colour:(UIColor*)colour
inRect:(CGRect)rect;
```

```
@end


@implementation PetalLayer

+(void)initialize
{
    if (self != [PetalLayer class])
        return;

    _pathCache = [[NSCache alloc] init];
}

+(CALayer*)petalOfType:(PetalType)petalType colour:(UIColor*)colour
inRect:(CGRect)rect
{
    NSString* a = nil;
    switch (petalType) {
        case PetalTypeArrow:
            a = ARROW;
            break;

        case PetalTypeDots:
            return [PetalLayer dotsWithColour:colour
referenceSize:rect.size];
            break;

        case PetalTypeDrumstick:
            a = DRUMSTICK;
            break;

        case PetalTypeLeaf:
            a = LEAF;
            break;

        case PetalTypeLine:
            a = LINE;
            break;

        case PetalTypeTeardrop:
            a = TEARDROP;
            break;

        case PetalTypeTriangle:
            a = TRIANGLE;
            break;

        default:
            break;
    }
    return [PetalLayer layerWithPathAttr:a colour:colour
inRect:rect];
}

+(CALayer*)dotsWithColour:(UIColor*)colour referenceSize:
(CGSize)size
{
//    CGRect bgRect = CGRectMake(0, 0, 9, 35);
    CGRect bgRect = CGRectMake(0, 0, size.width*0.3f,
size.height*1.16667f);
    CAShapeLayer* bgLayer = [CAShapeLayer layer];
    UIBezierPath* bgPath = [UIBezierPath bezierPathWithRect:bgRect];
    [bgLayer setPath:bgPath.CGPath];
    [bgLayer setFillColor:[UIColor clearColor].CGColor];

    UIBezierPath* dotsPath = [UIBezierPath bezierPath];
    [dotsPath appendPath:[UIBezierPath
bezierPathWithOvalInRect:CGRectMake(0.0f, 0.0f, 0.3f*size.width,
0.3f*size.width)]];
    [dotsPath appendPath:[UIBezierPath
bezierPathWithOvalInRect:CGRectMake(0.0333f*size.width,
0.4f*size.width, 0.23f*size.width, 0.23f*size.width)]];
    [dotsPath appendPath:[UIBezierPath
bezierPathWithOvalInRect:CGRectMake(0.0666f*size.width,
0.7333f*size.width, 0.16667f*size.width, 0.16667f*size.width)]];

    CAShapeLayer* dotsLayer = [CAShapeLayer layer];
    [dotsLayer setPath:dotsPath.CGPath];
    [dotsLayer setFillColor:colour.CGColor];

    CALayer* l = [CALayer layer];
    [l addSublayer:bgLayer];
    [l addSublayer:dotsLayer];
    [l setFrame:CGRectMake(0, 0, bgRect.size.width ,
bgRect.size.height)];
    return l;
}

+(CALayer*)layerWithPathAttr:(NSString*)attr colour:(UIColor*)colour
inRect:(CGRect)rect
{
    UIBezierPath *path = nil;//[_pathCache objectForKey:attr];
//    if (!path)
//    {
        SvgToBezier* converter = nil;
        if ([attr hasPrefix:@"M"] || [attr hasPrefix:@"m"])
        {
            converter = [[SvgToBezier alloc]
initFromSVGPathNodeDAttr:attr rect:rect];
```

```
        }
        else
        {
            converter = [[SvgToBezier alloc]
initFromSVGPathNodePointsAttr:attr rect:rect];
        }
        [_pathCache setObject:converter.bezier forKey:attr];
        path = converter.bezier;
//    }

    CAShapeLayer* l = [CAShapeLayer layer];
    [l setPath:path.CGPath];
    [l setFillColor:colour.CGColor];
    [l setFrame:CGRectMake(0, 0, path.bounds.size.width,
path.bounds.size.height)];

    return l;
}

+(PetalLayer*)petalLayerOfType:(PetalType)type withKeywords:
(NSArray*)keywords diameter:(float)diameter
{
    KeywordModel *km = [KeywordModel sharedModel];
    PetalLayer *_petalLayer = [[PetalLayer alloc] init];
    [_petalLayer setFrame:CGRectMake(0.0, 0.0, diameter, diameter)];
    [_petalLayer setOpaque:NO];
    float angle = M_PI*2/keywords.count;
    float offset = angle * randomFloat();
    int catId = -1;
    NSInteger keywordId = 0;
    CALayer* petal = nil;

    for (int i=0; i<keywords.count; i++)
    {
        keywordId = [(KeywordVO*)[keywords objectAtIndex:i]
objectId];
        catId = [km getCategoryIndexByItemId:keywordId];
        if (catId == NSNotFound)
        {
            NSLog(@"catId for keywordId %i not found" , keywordId);
            abort();
        }
        petal = [PetalLayer petalOfType:type colour:[UIColor
colorForCategoryId:catId] inRect:CGRectMake(0.0f, 0.0f,
diameter/3.0f, diameter/3.0f)];
        [petal setAnchorPoint:CGPointMake(0.5, 1.0)];
        [petal
setPosition:CGPointMake(_petalLayer.frame.size.width/2,
_petalLayer.frame.size.height/2)];
        [petal setTransform:CATransform3DMakeRotation(offset +
angle*i, 0.0, 0.0, 1.0)];
        [_petalLayer addSublayer:petal];
    }

//        _petalLayer.backgroundColor = [[UIColor colorWithRed:1.0
green:0 blue:0 alpha:0.5] CGColor];

    return _petalLayer;
}

+(PetalLayer*)monoChromePetalLayerOfType:(PetalType)type
withKeywords:(NSArray*)keywords
{
    PetalLayer *_petalLayer = [[PetalLayer alloc] init];
    CGRect rect = CGRectMake(0.0f, 0.0f, 30.0, 30.0);
    CALayer* petal = [PetalLayer petalOfType:type colour:[UIColor
colorWithHexRGB:0xC8B99D]  inRect:rect];
    [_petalLayer setFrame:CGRectMake(0.0, 0.0,
petal.frame.size.height*2, petal.frame.size.height*2)];
    float angle = M_PI*2/keywords.count;

    for (int i=0; i<keywords.count; i++)
    {
        petal = [PetalLayer petalOfType:type colour:[UIColor
colorWithHexRGB:0xC8B99D] inRect:rect];
        [petal setAnchorPoint:CGPointMake(0.5, 1.0)];
        [petal
setPosition:CGPointMake(_petalLayer.frame.size.width/2,
_petalLayer.frame.size.height/2)];
        [petal setTransform:CATransform3DMakeRotation(angle*i, 0.0,
0.0, 1.0)];
        [_petalLayer addSublayer:petal];
    }

    return _petalLayer;
}

@end

//
//  ChapterIconDelegate.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <Foundation/Foundation.h>

@class ChapterIcon;
```

```
@protocol ChapterIconDelegate <NSObject>

-(void)chapterIconTouched:(ChapterIcon*)icon;

@end

//
//  WelcomeView.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "WelcomeViewDelegate.h"
#import <SSToolkit/SSTextField.h>

@interface WelcomeView : UIView <UITextFieldDelegate>

@property (nonatomic, strong) IBOutlet UIImageView *bg;
@property (nonatomic, strong) IBOutlet UILabel *headerLabel;
@property (nonatomic, strong) IBOutlet UILabel *content1Label;
@property (nonatomic, strong) IBOutlet UILabel *content2Label;
@property (nonatomic, strong) IBOutlet SSTextField *nameTextField;
@property (nonatomic, weak) id<WelcomeViewDelegate> delegate;
@property (nonatomic, strong) IBOutlet UIButton *closeButton;

@end

//
//  WelcomeView.m
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "WelcomeView.h"
#import "UIView+Position.h"
#import "UIFont+NTF.h"
#import "UIColor+Color255.h"
#import "constants.h"
#import "notifications.h"

@implementation WelcomeView

-(void)awakeFromNib
{
    [self.headerLabel setFont:[UIFont
boldApplicationFontOfSize:38.0]];
```

```
    [self.content1Label setFont:[UIFont
applicationFontOfSize:18.0]];
    [self.content1Label setNumberOfLines:0];
    [self.content1Label setLineBreakMode:NSLineBreakByWordWrapping];

    [self.content2Label setFont:[UIFont
italicApplicationFontOfSize:18.0]];
    [self.content2Label setNumberOfLines:0];
    [self.content2Label setLineBreakMode:NSLineBreakByWordWrapping];

    [self.nameTextField setFont:[UIFont
boldItalicApplicationFontOfSize:18.0]];
    [self.nameTextField setTextColor:[UIColor
colorWithHexRGB:0x746E6C]];
    [self.nameTextField setDelegate:self];
    [self.nameTextField setReturnKeyType:UIReturnKeyDone];
    [self.nameTextField setPlaceholderTextColor:[UIColor
colorWithHexRGB:0x746E6C]];

    [self.headerLabel setText:NSLocalizedString(@"welcomeHeader",
nil)];
    [self.content1Label
setText:NSLocalizedString(@"welcomeContent1", nil)];
    [self.content2Label
setText:NSLocalizedString(@"welcomeContent2", nil)];
    [self.nameTextField
setPlaceholder:NSLocalizedString(@"welcomeNamePrompt", nil)];

    [self.content1Label sizeToFit];
    [self.content2Label sizeToFit];

    [self.closeButton addTarget:self action:@selector(close)
forControlEvents:UIControlEventTouchUpInside];

    [[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(moveUp) name:UIKeyboardWillShowNotification
object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(moveDown) name:UIKeyboardWillHideNotification
object:nil];
}

-(void)close
{
    [self.delegate closeWelcomeView:self];
}

-(void)moveUp
{
    [UIView animateWithDuration:0.5 delay:0
options:UIViewAnimationOptionCurveEaseOut animations:^{
        self.y -= 180.0f;
    } completion:^(BOOL finished) {

    }];
}

-(void)moveDown
{
    [UIView animateWithDuration:0.5 delay:0
options:UIViewAnimationOptionCurveEaseOut animations:^{
        self.y = 0.0f;
    } completion:^(BOOL finished) {

    }];
}


-(void)willMoveToSuperview:(UIView *)newSuperview
{
    if (newSuperview == nil)
    {
        [[NSNotificationCenter defaultCenter] removeObserver:self];
    }
}

#pragma mark - UITextFieldDelegate

- (BOOL)textFieldShouldReturn:(UITextField *)textField
{
    if (textField.text.length > 0)
    {
        [[NSNotificationCenter defaultCenter]
postNotificationName:CHANGE_USERNAME

object:self

userInfo:@{USERNAME:textField.text}];
        [self.delegate closeWelcomeView:self];
    }
    [textField resignFirstResponder];

    return YES;
}
@end

//
//  WelcomeViewDelegate.h
//  NostalgiaTiFrega
//
```

```
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <Foundation/Foundation.h>

@protocol WelcomeViewDelegate <NSObject>

-(void)closeWelcomeView:(UIView*)welcomeView;

@end


//
//  IconInfoView.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>

@class ChapterVO;

@interface IconInfoView : UIView

@property (nonatomic, strong, readonly) UIImageView *bg;
@property (nonatomic, strong, readonly) UIImageView *triangle;
@property (nonatomic, strong, readonly) UILabel *nameLabel;
@property (nonatomic, strong, readonly) UILabel *locationDateLabel;
@property (nonatomic, weak) NSString *nameText;
@property (nonatomic, weak) NSString *locationDateText;
@property (nonatomic, weak) ChapterVO *vo;

@end

//
//  IconInfoView.m
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "IconInfoView.h"
#import "UIFont+NTF.h"
#import "UIColor+Color255.h"
#import "UIView+Position.h"
#import "ChapterVO.h"
#import "NSDateFormatter+NTF.h"

@implementation IconInfoView
```

```
- (id)init
{
    UIImage *im = [UIImage imageNamed:@"chapterIconPopupBg"];
    self = [super initWithFrame:CGRectMake(0.0f, 0.0f,
im.size.width, im.size.height)];
    if (self)
    {
        _bg = [[UIImageView alloc] initWithImage:[im
resizableImageWithCapInsets:UIEdgeInsetsMake(0.0f, 40.0f, 0.0f,
40.0f)]];

        _triangle = [[UIImageView alloc] initWithImage:[UIImage
imageNamed:@"chapterIconPopupTriangle"]];
        _triangle.y = _bg.frame.size.height - 6.0f;

        _nameLabel = [[UILabel alloc] initWithFrame:CGRectMake(0.0f,
10.0f, _bg.frame.size.width, 20.0f)];
        _nameLabel.font = [UIFont
boldItalicApplicationFontOfSize:18.0];
        _nameLabel.textColor = [UIColor colorWithHexRGB:0x726C6C];
        _nameLabel.backgroundColor = [UIColor clearColor];
        _nameLabel.textAlignment = NSTextAlignmentCenter;

        _locationDateLabel = [[UILabel alloc]
initWithFrame:CGRectMake(0.0f, 34.0f, _bg.frame.size.width, 20.0f)];
        _locationDateLabel.font = [UIFont
italicApplicationFontOfSize:18.0];
        _locationDateLabel.textColor = _nameLabel.textColor;
        _locationDateLabel.backgroundColor = [UIColor clearColor];
        _locationDateLabel.textAlignment = NSTextAlignmentCenter;

        [self addSubview:_bg];
        [self addSubview:_triangle];
        [self addSubview:_nameLabel];
        [self addSubview:_locationDateLabel];
    }
    return self;
}

- (id)initWithFrame:(CGRect)frame
{
    return [self init];
}

-(void)setNameText:(NSString *)nameText
{
    _nameLabel.text = nameText;
    [_nameLabel sizeToFit];
    [self setNeedsLayout];
```

```
}

-(void)setLocationDateText:(NSString *)locationDateText
{
    _locationDateLabel.text = locationDateText;
    [_locationDateLabel sizeToFit];
    [self setNeedsLayout];
}

-(void)layoutSubviews
{
    [super layoutSubviews];
    UILabel *longLabel = nil;
    if (_nameLabel.frame.size.width >
_locationDateLabel.frame.size.width)
    {
        longLabel = _nameLabel;
    }
    else
    {
        longLabel = _locationDateLabel;
    }

    _bg.frame = CGRectMake(_bg.x, _bg.y,
longLabel.frame.size.width+80.0f, _bg.frame.size.height);
    _nameLabel.frame = CGRectMake(_nameLabel.x, _nameLabel.y,
_bg.frame.size.width, _nameLabel.frame.size.height);
    _locationDateLabel.frame = CGRectMake(_locationDateLabel.x,
_locationDateLabel.y, _bg.frame.size.width,
_locationDateLabel.frame.size.height);
    _triangle.x = (_bg.frame.size.width -
_triangle.frame.size.width)/2.0f;

    self.bounds = CGRectMake(self.bounds.origin.x,
self.bounds.origin.y, _bg.bounds.size.width,
CGRectGetMaxY(_triangle.frame));
}


#pragma mark - getter/setter

-(void)setVo:(ChapterVO *)vo
{
    _vo = vo;
    NSString *name = _vo.username;
    if (!name || name.length == 0) name =
NSLocalizedString(@"Unknown", nil);
    [self setNameText:name];

    NSDateFormatter *df = [NSDateFormatter
dateFormatterWithFormat:@"dd.MM.yyyy"];
    NSString *dateString = [df stringFromDate:_vo.createdAt];
    NSString *locationString = [_vo.locationForCurrentLanguage
copy];
    if (locationString.length > 0)
    {
        [self setLocationDateText:[locationString
stringByAppendingString:[NSString stringWithFormat:@", %@" ,
dateString]]];
    }
    else
    {
        [self setLocationDateText:dateString];
    }
}

@end

//
//  IconManager.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <Foundation/Foundation.h>
#import "ChapterVO.h"

@interface IconMaker : NSObject

+(void)getIconImageForVO:(ChapterVO*)vo completion:(void (^)(UIImage
*image))completion;
+(void)updateIconImageForVO:(ChapterVO *)vo completion:(void (^)
(UIImage *image))completion;

@end

//
//  IconManager.m
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "IconMaker.h"
#import <QuartzCore/QuartzCore.h>
#import "ChapterIcon.h"

static NSCache *__imageCache;
```

```
@implementation IconMaker

+(void)initialize
{
    if (self == [IconMaker class])
    {
        __imageCache = [[NSCache alloc] init];
    }
}

+(void)getIconImageForVO:(ChapterVO*)vo completion:(void (^)(UIImage
*image))completion
{
    dispatch_async(dispatch_get_global_queue(
DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), ^{
        UIImage *im = [__imageCache objectForKey:vo.objectId];

        if (!im)
        {
            NSString *name = [NSString stringWithFormat:@"icon_
%@.png" , vo.objectId];
            NSString *imagePath = [[self cachePath]
stringByAppendingPathComponent:name];
            if ([[NSFileManager defaultManager]
fileExistsAtPath:imagePath])
            {
                NSData *imageData = [[NSData alloc]
initWithContentsOfFile:imagePath];
                im = [UIImage imageWithData:imageData scale:
[[UIScreen mainScreen] scale]];
                [__imageCache setObject:im forKey:vo.objectId];
            }
            else
            {
                im = [self updateIconImageForVO:vo];
            }
        }

        dispatch_async(dispatch_get_main_queue(), ^{
            completion(im);
        });
    });
}

+(void)updateIconImageForVO:(ChapterVO *)vo completion:(void (^)
(UIImage *image))completion
{
    dispatch_async(dispatch_get_global_queue(
DISPATCH_QUEUE_PRIORITY_DEFAULT, 0), ^{
        UIImage *im = [IconMaker updateIconImageForVO:vo];
        dispatch_async(dispatch_get_main_queue(), ^{
            if (completion) completion(im);
        });
    });
}

+(UIImage*)updateIconImageForVO:(ChapterVO *)vo
{
    CGSize size = [ChapterIcon sizeForVO:vo];
    float radius = size.width/2.0;

    CAShapeLayer *bgLayer = [IconMaker createBgWithRadius:radius];
    CALayer *petalLayer = [PetalLayer
petalLayerOfType:vo.firstCategory

withKeywords:vo.keywords

diameter:radius*2.0];

    CALayer *l = [CALayer layer];
    l.frame = (CGRect){{0,0},size};
    [l addSublayer:bgLayer];
    [l addSublayer:petalLayer];

    NSString *name = [NSString stringWithFormat:@"icon_%@.png" ,
vo.objectId];
    UIImage *im = [self writeLayer:l toPNGWithName:name];
    [__imageCache setObject:im forKey:vo.objectId];

    return im;
}

+(UIImage*)writeLayer:(CALayer*)layer toPNGWithName:(NSString*)name
{
    NSString *filePath =  [[self cachePath]
stringByAppendingPathComponent:name];
    UIImage *im = [IconMaker imageWithCALayer:layer];
    [UIImagePNGRepresentation(im) writeToFile:filePath
atomically:YES];

    return im;
}

+(NSString*)cachePath
{
    static NSString *cachePath;
    if (!cachePath)
    {
        NSArray *paths =
NSSearchPathForDirectoriesInDomains(NSCachesDirectory,
NSUserDomainMask, YES);
        cachePath = [paths objectAtIndex:0];
        BOOL isDir = NO;
        NSError *error;
        if (! [[NSFileManager defaultManager]
fileExistsAtPath:cachePath isDirectory:&isDir] && isDir == NO) {
            [[NSFileManager defaultManager]
createDirectoryAtPath:cachePath withIntermediateDirectories:NO
attributes:nil error:&error];
        }
    }
    return cachePath;
}

+(UIImage*)imageWithCALayer:(CALayer*)layer
{
    UIGraphicsBeginImageContextWithOptions(layer.frame.size, NO,
[[UIScreen mainScreen] scale]);

    [layer renderInContext:UIGraphicsGetCurrentContext()];
    UIImage* out = UIGraphicsGetImageFromCurrentImageContext();

    UIGraphicsEndImageContext();

    return out;
}

+(CAShapeLayer*)createBgWithRadius:(float)radius
{
    CGPathRef path = CGPathRetain([[UIBezierPath
bezierPathWithArcCenter:CGPointMake(0, 0)

radius:radius

startAngle:0

endAngle:M_PI*2.0

clockwise:YES] CGPath]);

    CAShapeLayer *bgLayer = [CAShapeLayer layer];
    [bgLayer setFillColor:[[UIColor whiteColor] CGColor]];
    [bgLayer setPath:path];
    [bgLayer setPosition:CGPointMake(0, 0)];
    [bgLayer setFrame:CGRectMake(radius, radius, radius*2.0,
radius*2.0)];

    CGPathRelease(path);
    return bgLayer;
}

@end

//
//  NTFViewController.h
//  Curate
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "CurateView.h"
#import "ButtonView.h"
#import "ChapterViewController.h"
#import <MapKit/MapKit.h>
#import <CoreLocation/CoreLocation.h>

@interface CurateViewController : UIViewController
<UIAlertViewDelegate , UITextFieldDelegate>

@property (nonatomic, weak, readonly) CurateView* curateView;
@property (nonatomic, strong, readonly) UIAlertView *enterNameAlert;
@property (nonatomic, strong, readonly) ChapterViewController
*chapterViewController;

-(void)onModelUpdate:(NSNotification*)notification;
-(void)onTooManyKeywords;
-(void)onSubmitKeywords;

@end

//
//  NTFViewController.m
//  Curate
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "CurateViewController.h"
#import "KeywordModel.h"
#import "notifications.h"
#import "ChapterVO.h"
#import "MBProgressHUD.h"
#import "constants.h"
#import "ImageModel.h"
#import "NPReachability.h"
#import "NTFServices.h"
```

```
@implementation CurateViewController

@synthesize curateView = _curateView;

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    if (self.isViewLoaded && self.view.window)
    {
        _enterNameAlert = nil;
        _chapterViewController = nil;
    }
}


#pragma mark - View lifecycle

-(void)loadView
{
    [[UIApplication sharedApplication] setStatusBarHidden:YES];

    [self setView:[[CurateView alloc] initWithFrame:[[UIScreen
mainScreen] applicationFrame]]];
    _curateView = (CurateView*)self.view;

    [_curateView setItems:[[KeywordModel sharedModel] categories]];
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    _curateView.nameInputField.text = [self username];
    _curateView.nameInputField.delegate = self;

    [[NSNotificationCenter defaultCenter] addObserver:self

selector:@selector(onModelUpdate:)
                                                 name:MODEL_UPDATE
                                               object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self

selector:@selector(onButtonDown:)
                                                 name:BUTTON_DOWN
                                               object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self

selector:@selector(onButtonUp:)
```

```
                                                 name:BUTTON_UP
                                               object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self

selector:@selector(onTooManyKeywords)

name:TOO_MANY_KEYWORDS
                                               object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self

selector:@selector(onSubmitKeywords)

name:SUBMIT_KEYWORDS
                                               object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self

selector:@selector(onShuffleClick:)
                                                 name:SHUFFLE
                                               object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self

selector:@selector(onExploreClick:)
                                                 name:SHOW_EXPLORE
                                               object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self

selector:@selector(onChapterSaved:)
                                                 name:CHAPTER_SAVED
                                               object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self

selector:@selector(onChapterSaveFailed:)

name:CHAPTER_SAVE_FAILED
                                               object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self

selector:@selector(onUsernameChanged:)

name:USERNAME_CHANGED
                                               object:nil];
}

-(void)viewWillAppear:(BOOL)animated
{
    [super viewWillAppear:animated];
    [[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(onKeyboardWillHide:)
```

```
name:UIKeyboardWillHideNotification object:nil];
}

-(void)viewWillDisappear:(BOOL)animated
{
    [super viewWillDisappear:animated];
    [[NSNotificationCenter defaultCenter] removeObserver:self
name:UIKeyboardWillHideNotification object:nil];
}

-(void)viewDidDisappear:(BOOL)animated
{
    [[NSNotificationCenter defaultCenter]
postNotificationName:CLEAR_KEYWORDS object:self];
}

- (void)viewDidUnload
{
    [super viewDidUnload];
    // Release any retained subviews of the main view.
    // e.g. self.myOutlet = nil;
}

- (BOOL)shouldAutorotateToInterfaceOrientation:
(UIInterfaceOrientation)interfaceOrientation
{
    return
NO;//UIInterfaceOrientationIsLandscape(interfaceOrientation);
}


#pragma mark - event handlers

-(void)onModelUpdate:(NSNotification*)notification
{
    NSMutableArray* a = [[NSMutableArray alloc] init];
    for (NSArray* innerArray in [[KeywordModel sharedModel]
categories])
    {
        for (KeywordVO* vo in innerArray)
        {
            if (vo.isSelected)
            {
                [a addObject:vo];
            }
        }
    }

    [_curateView.dockView updateButtons:a];
}
```

```
-(void)onButtonDown:(NSNotification*)notification
{
    int i = [[notification object] intValue];
    [CATransaction begin];
    [CATransaction setDisableActions: YES];
    [[_curateView.circles layerWithTag:i] setOpacity:0.3];
    [CATransaction commit];
}

-(void)onButtonUp:(NSNotification*)notification
{
    int i = [[notification object] intValue];
    [[_curateView.circles layerWithTag:i] setOpacity:1.0];
}

-(void)onTooManyKeywords
{
    UIAlertView* av = [[UIAlertView alloc] initWithTitle:@""

message:NSLocalizedString(@"tooManyKeywords", nil)
                                                delegate:nil

cancelButtonTitle:NSLocalizedString(@"tooManyKeywordsCancel", nil)
                                       otherButtonTitles:nil];
    [av show];
}

-(void)onSubmitKeywords
{
    if ([[KeywordModel sharedModel] numSelectedItems] < 1)
    {
        UIAlertView* av = [[UIAlertView alloc] initWithTitle:@""

message:NSLocalizedString(@"tooFewKeywords", nil)
                                                    delegate:nil

cancelButtonTitle:NSLocalizedString(@"tooFewKeywordsCancel", nil)
                                           otherButtonTitles:nil];
        [av show];
    }
    else if ([self username] == nil || [self username].length == 0)
    {
        _enterNameAlert = [[UIAlertView alloc]
initWithTitle:NSLocalizedString(@"welcomeNamePrompt", nil)

message:NSLocalizedString(@"addYourName", nil)
                                                    delegate:self
cancelButtonTitle:NSLocalizedString(@"OK", nil)
                                          otherButtonTitles:nil];
        [_enterNameAlert
setAlertViewStyle:UIAlertViewStylePlainTextInput];
        [[_enterNameAlert textFieldAtIndex:0]
setAutocapitalizationType:UITextAutocapitalizationTypeWords];
        [[_enterNameAlert textFieldAtIndex:0] setDelegate:self];
        [_enterNameAlert show];
    }
    else
    {
        [self submitKeywordsWithUsername:[self username]];
    }
}

-(void)submitKeywordsWithUsername:(NSString*)username
{
    [self setUsername:username];
    [MBProgressHUD showHUDAddedTo:[[[UIApplication
sharedApplication] delegate] window] animated:YES];

    NSArray *imageVOs = [[ImageModel sharedModel]
getImagesWithKeywords:[[KeywordModel sharedModel] selectedItems]
limit:10];
    NSMutableArray* imageIds = [[NSMutableArray alloc] init];
    [imageVOs enumerateObjectsUsingBlock:^(id obj, NSUInteger idx,
BOOL *stop) {
        ImageVO *ivo = (ImageVO*)obj;
        [imageIds addObject:ivo.objectId];
    }];
    ChapterVO *cvo = [[ChapterVO alloc] initWithId:nil
imageIds:imageIds keywords:[[KeywordModel sharedModel]
selectedItems] username:username];

    [cvo setLocation_en:[[NTFServices locationManager]
locationStrings][@"en"]];
    [cvo setLocation_it:[[NTFServices locationManager]
locationStrings][@"it"]];

    KeywordVO *kvo = [[KeywordModel sharedModel] selectedItems][0];
    [cvo setFirstCategory:[[KeywordModel sharedModel]
getCategoryIndexByItemId:kvo.objectId]];

    [[NSNotificationCenter defaultCenter]
postNotificationName:SAVE_CHAPTER object:self userInfo:@{VO:cvo}];
}

-(void)onShuffleClick:(NSNotification*)notification
{
    [_curateView doShuffleAnimation];
    [self performSelector:@selector(onShuffleAnimationComplete)
withObject:nil afterDelay:1.5];
}

-(void)onExploreClick:(NSNotification*)notification
{
    [[NSNotificationCenter defaultCenter]
postNotificationName:NAVIGATE_TO_EXPLORE object:self userInfo:nil];
}

-(void)onChapterSaved:(NSNotification*)notification
{
    [MBProgressHUD hideHUDForView:[[[UIApplication
sharedApplication] delegate] window] animated:YES];
    [[NSNotificationCenter defaultCenter]
postNotificationName:NAVIGATE_TO_CHAPTER object:self userInfo:
[notification userInfo]];
}

-(void)onChapterSaveFailed:(NSNotification*)notification
{
    [MBProgressHUD hideHUDForView:[[[UIApplication
sharedApplication] delegate] window] animated:YES];
}

-(void)onUsernameChanged:(NSNotification*)notification
{
    _curateView.nameInputField.text =
notification.userInfo[USERNAME];
}

-(void)onShuffleAnimationComplete
{
    [[NSNotificationCenter defaultCenter]
postNotificationName:SELECT_RANDOM_KEYWORDS object:nil];
}

-(void)onKeyboardWillHide:(NSNotification*)notification
{
    if (![_curateView.nameInputField.text isEqualToString:[self
username]])
    {
        _curateView.nameInputField.text = [self username];
    }
}


#pragma mark - UIAlertViewDelegate

-(void)alertView:(UIAlertView *)alertView clickedButtonAtIndex:
```

```
(NSInteger)buttonIndex
{
    if (alertView == _enterNameAlert)
    {
        [self submitKeywordsWithUsername:[[alertView
textFieldAtIndex:0] text]];
    }
}


#pragma mark - UITextFieldDelegate

-(BOOL)textFieldShouldReturn:(UITextField *)textField
{
    if (textField == [_enterNameAlert textFieldAtIndex:0])
    {
        NSString *name = [[_enterNameAlert textFieldAtIndex:0]
text];
        [_enterNameAlert
dismissWithClickedButtonIndex:_enterNameAlert.firstOtherButtonIndex
animated:YES];
        [self submitKeywordsWithUsername:name];
    }
    else if (textField == _curateView.nameInputField)
    {
        [self setUsername:textField.text];
        [textField resignFirstResponder];
    }
    return YES;
}


#pragma mark - getter/setter

-(NSString*)username
{
    return [[NSUserDefaults standardUserDefaults]
objectForKey:USERNAME];
}

-(void)setUsername:(NSString*)name
{
    [[NSNotificationCenter defaultCenter]
postNotificationName:CHANGE_USERNAME
                                                    object:self

userInfo:@{USERNAME:name}];
}

@end
```

```
//
//  MainView.h
//  Curate
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>
#import <QuartzCore/QuartzCore.h>
#import "CircleView.h"
#import "ButtonView.h"
#import "CircleModel.h"
#import "CircleVO.h"
#import "DockView.h"
#import <SSToolkit/SSTextField.h>

@interface CurateView : UIView

@property (nonatomic, strong, readonly) DockView* dockView;
@property (nonatomic, strong, readonly) UIImageView *textFieldBg;
@property (nonatomic, strong, readonly) SSTextField *nameInputField;
@property (nonatomic, strong, readonly) UIButton *shuffleButton;

-(void)setItems:(NSArray*)items;
-(UIView*)buttonViewWithTag:(int)tag;
-(CircleView*)circles;
-(void)doShuffleAnimation;

@end

//
//  MainView.m
//  Curate
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "CurateView.h"
#import "UIScreen+CKAdditions.h"
#import "UIView+Position.h"
#import "UIFont+NTF.h"
#import "UIColor+Color255.h"
#import "UIButton+NTF.h"
#import "notifications.h"

@implementation CurateView
{
    UIImageView* _bg;
    CircleView* _circles;
    NSArray* _buttonViews;
    UITapGestureRecognizer* _nameBgGR;
}

@synthesize dockView = _dockView;

- (id)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];
    if (self)
    {
        _bg = [[UIImageView alloc] initWithImage:[UIImage
imageNamed:@"bg"]];


        CGRect f = [[UIScreen mainScreen]
rectWithOrientation:UIInterfaceOrientationLandscapeLeft];
        _circles = [[CircleView alloc] initWithFrame:CGRectMake(0,
0, f.size.width, 550)];

        CGRect af = [[UIScreen mainScreen] applicationFrame];
        _buttonViews = [[NSArray alloc] initWithObjects:
                        [[ButtonView alloc] initWithFrame:af],
                        [[ButtonView alloc] initWithFrame:af],
                        [[ButtonView alloc] initWithFrame:af],
                        [[ButtonView alloc] initWithFrame:af],
                        [[ButtonView alloc] initWithFrame:af],
                        [[ButtonView alloc] initWithFrame:af],
                        [[ButtonView alloc] initWithFrame:af],
                        nil];

        _dockView = [[DockView alloc] initWithFrame:f];

        [self addSubview:_bg];
        [self addSubview:_dockView];
        [self addSubview:_circles];

        ButtonView* bv = nil;
        CircleVO* vo = nil;
        for (int i=[_buttonViews count]-1; i>-1; i--)
        {
            bv = [_buttonViews objectAtIndex:i];
            [bv setTag:i];
            vo = [[[CircleModel sharedInstance] items]
objectAtIndex:i];
            [bv setColorNormal:[vo color]];
            [bv setColorDark:[vo darkColor]];
            [bv setRadius:[vo radius]];
            [bv setCenter:[vo pos]];
            [self addSubview:bv];
        }

        _nameBgGR = [[UITapGestureRecognizer alloc]
initWithTarget:self action:@selector(nameBgTapped)];
        _textFieldBg = [[UIImageView alloc] initWithImage:[UIImage
imageNamed:@"textInputBg"]];
        _textFieldBg.userInteractionEnabled = YES;
        _textFieldBg.position = CGPointMake(25.0f, 14.0f);
        [_textFieldBg addGestureRecognizer:_nameBgGR];

        _nameInputField = [[SSTextField alloc]
initWithFrame:CGRectMake(52.0f, 19.0f, 266.0f, 30.0f)];
//        _nameInputField.backgroundColor = [UIColor
colorWithRed:1.0 green:0.0 blue:0.0 alpha:0.5];
        _nameInputField.font = [UIFont
boldItalicApplicationFontOfSize:18.0f];
        _nameInputField.clearButtonMode =
UITextFieldViewModeWhileEditing;
        _nameInputField.textColor = [UIColor
colorWithHexRGB:0x746E6C];
        _nameInputField.placeholderTextColor =
_nameInputField.textColor;
        _nameInputField.returnKeyType = UIReturnKeyDone;
        _nameInputField.placeholder =
NSLocalizedString(@"welcomeNamePrompt", nil);
        _nameInputField.autocapitalizationType =
UITextAutocapitalizationTypeWords;
//        _nameInputField.clearButtonMode =
UITextFieldViewModeWhileEditing;

        _shuffleButton = [UIButton shuffleButton];
        [_shuffleButton setPosition:CGPointMake(939.0, 473.0)];
        [_shuffleButton addTarget:self
action:@selector(onShuffleClick:)
forControlEvents:UIControlEventTouchUpInside];

        [self addSubview:_textFieldBg];
        [self addSubview:_nameInputField];
        [self addSubview:_shuffleButton];
    }
    return self;
}

-(void)setItems:(NSArray *)items
{
    for (int i=0; i<[items count]; i++)
    {
        ButtonView* bv = [_buttonViews objectAtIndex:i];
        [bv setButtonVOs:[items objectAtIndex:i]];
    }
```

```
}

-(UIView*)buttonViewWithTag:(int)tag
{
    for (UIView* v in _buttonViews)
    {
        if (v.tag == tag) return v;
    }
    return nil;
}

-(CircleView *)circles
{
    return _circles;
}

-(void)doShuffleAnimation
{
    for (int i=0; i<[_buttonViews count] ; i++) {
        ButtonView* bv = [_buttonViews objectAtIndex:i];
        [bv spin:i%2==0];
    }
}


#pragma mark - event handlers

-(void)nameBgTapped
{
    [_nameInputField becomeFirstResponder];
}

-(void)onShuffleClick:(id)sender
{
    [[NSNotificationCenter defaultCenter]
postNotificationName:SHUFFLE object:nil];
}

@end


//
//  ButtonView.h
//  Curate
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "UIView+Rotation.h"
```

```
@interface ButtonView : UIView

@property (nonatomic) float radius;
@property (nonatomic , strong) NSTimer* timer;
@property (nonatomic , strong) UIColor* colorNormal;
@property (nonatomic , strong) UIColor* colorDark;
@property (nonatomic) BOOL showOutput;
@property (nonatomic , strong) UILabel *outLabel;

-(void)buttonTouchDown:(id)sender forEvent:(UIEvent *)event;
-(void)buttonTouchUp:(id)sender forEvent:(UIEvent *)event;
-(void)createStartAngleWithTouch:(UITouch*)touch;
-(UIButton*)buttonWithTag:(int)tag;
-(void)setButtonVOs:(NSArray *)vos;
-(void)returnToMax;
-(void)returnToMin;
-(void)constrainScroll;
-(BOOL)circleCrossesRightEdge;
-(void)stopTimer;
-(void)startTimer;
-(void)spin:(BOOL)clockwise;
-(void)spinComplete:(BOOL)clockwise;
-(float)scrollPos;

@end

//
//  ButtonView.m
//  Curate
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "notifications.h"
#import "ButtonView.h"
#import "trig.h"
#import "KeywordVO.h"
#import "KeywordModel.h"
#import <QuartzCore/QuartzCore.h>
#import "UIFont+NTF.h"
#import "UIScreen+CKAdditions.h"

static const float DAMPING = 0.95;
static const float MAXIMUM_FRAME_RATE = 30.0;
static const float MINIMUM_FRAME_RATE = 15.0;
static const float UPDATE_INTERVAL = (1.0 / MAXIMUM_FRAME_RATE);
static const float MAX_CYCLES_PER_FRAME = (MAXIMUM_FRAME_RATE /
MINIMUM_FRAME_RATE);
static const float MAX_RV = 0.03;
```

```
@implementation ButtonView
{
    @private
    NSArray*        _buttonVOs;
    NSMutableArray* _buttons;
    float           _radius;
    CGPoint         _startPos;
    float           _touchStartAngle;
    BOOL            _isDragging;
    float           _rv;
    float           _lastRotation;
    double          _lastFrameTime;
    double          _cyclesLeftOver;
    float           _minScroll;
    float           _maxScroll;
    float           _contentArc;
    float           _containerArc;
    BOOL            _cancelButtonTouchUp;
    NSTimer*        _updateTimer;
}

@synthesize colorNormal = _colorNormal;
@synthesize colorDark = _colorDark;
@synthesize timer;

- (id)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];
    if (self)
    {
//        [self setBackgroundColor:[UIColor colorWithRed:1 green:0
blue:0 alpha:1]];
        [[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(onModelUpdate:) name:MODEL_UPDATE object:nil];
        _rv = 0.0;
        _cyclesLeftOver = 0.0;
        _lastFrameTime = 0.0;
    }
    return self;
}

-(void)setButtonVOs:(NSArray *)vos
{
    static const float padding = 20.0;
    _buttons = [[NSMutableArray alloc] init];
    _buttonVOs = vos;
    UIButton* b = nil;
    float angle = M_PI_2;
    CGPoint p;
    float currentAngle = 0.0;
    float previousAngle = 0.0;
    KeywordVO* vo = nil;

    for (int i=0; i<[_buttonVOs count]; i++)
    {
        vo = [_buttonVOs objectAtIndex:i];
        b = [UIButton buttonWithType:UIButtonTypeCustom];
        [b setTag:[vo objectId]];

        [b setTitleColor:[UIColor blackColor]
forState:UIControlStateNormal];
        [b setTitleColor:_colorDark
forState:UIControlStateSelected];
        [b setTitleColor:[UIColor colorWithRed:0.35 green:0.35
blue:0.35 alpha:1.0] forState:UIControlStateHighlighted];
        [b.titleLabel setFont:[UIFont applicationFontOfSize:18.0]];
        [b setTitle:[vo title] forState:UIControlStateNormal];
        [b sizeToFit];
        [b addTarget:self
action:@selector(buttonTouchDown:forEvent:)
forControlEvents:UIControlEventTouchDown];
        [b addTarget:self action:@selector(buttonTouchUp:forEvent:)
forControlEvents:UIControlEventTouchUpInside];
        [b addTarget:self action:@selector(buttonTouchUp:forEvent:)
forControlEvents:UIControlEventTouchUpOutside];
        [_buttons addObject:b];

        currentAngle = atanf((b.frame.size.width/2.0 +
padding)/_radius);
        angle -= currentAngle + previousAngle;

        previousAngle = currentAngle;


        CGAffineTransform transform =
CGAffineTransformMakeRotation(angle - M_PI_2);
        b.transform = transform;

        p = polarToCartesian(radialPointMake(_radius-20.0, angle));
        p.x += self.frame.size.width/2;
        p.y += self.frame.size.height/2;
        [b setCenter:p];

        [self addSubview:b];
    }
    // arc that the button content describes
    currentAngle = atanf(padding/_radius);
    angle -= currentAngle + previousAngle;
    _contentArc = M_PI_2 - angle;
    _maxScroll = asinf(self.center.x/_radius);

    _minScroll = asinf(([[UIScreen mainScreen] currentSize].height -
self.center.x)/_radius) * -1;
    if (![self circleCrossesRightEdge])
    {
        _minScroll = acosf((fabsf(self.center.y) + 20)/_radius) *
-1;
    }
    _containerArc = _maxScroll - _minScroll;

    _minScroll += _contentArc;

    [self setRotation:_maxScroll];

    [self setUserInteractionEnabled:YES];

    UIPanGestureRecognizer *panRecognizer = [[UIPanGestureRecognizer
alloc] initWithTarget:self action:@selector(panDetected:)];
    [panRecognizer setCancelsTouchesInView:NO];
    [panRecognizer setDelaysTouchesEnded:NO];
    [self addGestureRecognizer:panRecognizer];

    [self startTimer];
}

-(BOOL)circleCrossesRightEdge
{
    return sqrtf(powf(1024-self.center.x, 2.0) + powf(self.center.y,
2.0)) < _radius;
}

-(void)buttonTouchDown:(id)sender forEvent:(UIEvent *)event
{
    _isDragging = YES;
    [self createStartAngleWithTouch:[[event touchesForView:sender]
anyObject]];
    [[NSNotificationCenter defaultCenter]
postNotificationName:BUTTON_DOWN object:[NSNumber
numberWithInt:self.tag]];
}

-(void)buttonTouchUp:(id)sender forEvent:(UIEvent *)event
{
    _isDragging = NO;
    UIButton* b = sender;
    if (!_cancelButtonTouchUp)
    {
        [[NSNotificationCenter defaultCenter]
postNotificationName:BUTTON_CLICK object:[NSNumber
numberWithInt:b.tag]];
    }
    [[NSNotificationCenter defaultCenter]
postNotificationName:BUTTON_UP object:[NSNumber
numberWithInt:self.tag]];
//    [self constrainScroll];
    _cancelButtonTouchUp = NO;
}

-(void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
{
    _isDragging = YES;
    [self createStartAngleWithTouch:[touches anyObject]];
    [[NSNotificationCenter defaultCenter]
postNotificationName:BUTTON_DOWN object:[NSNumber
numberWithInt:self.tag]];
}

-(void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event
{
    _isDragging = NO;
    [[NSNotificationCenter defaultCenter]
postNotificationName:BUTTON_UP object:[NSNumber
numberWithInt:self.tag]];
//    [self constrainScroll];
    _cancelButtonTouchUp = NO;
}

-(void)returnToMax
{
    _rv = 0;
    void (^anim) (void) = ^{
        [self setRotation:_maxScroll];
    };

    [UIView animateWithDuration:1.0 delay:0
options:UIViewAnimationOptionCurveEaseInOut animations:anim
completion:nil];
}

-(void)returnToMin
{
    _rv = 0;
    void (^anim) (void) = ^{
        [self setRotation:_minScroll];
    };

    [UIView animateWithDuration:1.0 delay:0
options:UIViewAnimationOptionCurveEaseInOut animations:anim
```

```
completion:nil];
}

-(void)constrainScroll
{
    // if both ends are visible do nothing
    if (self.rotation >= _minScroll && [self rotation] <=
_maxScroll)
    {
        return;
    }

    // if neither end is visible do nothing
    if (self.rotation <= _minScroll && [self rotation] >=
_maxScroll)
    {
        return;
    }

    if (_contentArc > _containerArc)
    {
        // move to one end or t'other
        if (self.rotation > _minScroll)
        {
            [self returnToMin];
        }
        else if (self.rotation < _maxScroll)
        {
            [self returnToMax];
        }
    }
    else
    {
        if (self.rotation > _minScroll)
        {
            [self returnToMax];
        }
        else if (self.rotation < _maxScroll)
        {
            [self returnToMin];
        }
    }
}

-(void)createStartAngleWithTouch:(UITouch*)touch
{
    CGPoint touchPoint = [touch locationInView:self.superview];
    CGPoint ap = CGPointMake(touchPoint.x - self.center.x,
touchPoint.y - self.center.y);
    _touchStartAngle = cartesianToPolar(ap).angle;
    _lastRotation = _touchStartAngle;
}

-(void)panDetected:(UIPanGestureRecognizer *)panRecognizer
{
    _cancelButtonTouchUp = YES;
    CGPoint touchPoint = [panRecognizer
locationInView:self.superview];
    CGPoint ap = CGPointMake(touchPoint.x - self.center.x,
touchPoint.y - self.center.y);
    float touchAngle = cartesianToPolar(ap).angle;
    float angle = touchAngle - _touchStartAngle;
    [self incrementRotationBy:angle];
    _touchStartAngle = touchAngle;
}

-(NSArray*)buttonVOs
{
    return _buttonVOs;
}

-(void)setCenter:(CGPoint)center
{
    [super setCenter:center];
    _startPos = self.frame.origin;
}

-(void)setRadius:(float)radius
{
    [self setFrame:CGRectMake(0, 0, (radius+20)*2, (radius+20)*2)];

    _radius = radius+20;
}

-(float)radius
{
    return _radius;
}

- (BOOL)pointInside:(CGPoint)point withEvent:(UIEvent *)event
{
    // Return NO if even super returns NO (i.e., if point lies
outside our bounds)
    BOOL superResult = [super pointInside:point withEvent:event];
    if (!superResult)
    {
        return superResult;
    }

    CGPoint ac = CGPointMake(self.center.x - _startPos.x ,
self.center.y - _startPos.y);
    return pow((point.x - ac.x), 2) + pow((point.y - ac.y), 2) <
pow(_radius , 2);
}

-(void)onModelUpdate:(NSNotification*)noti
{
    KeywordVO* vo = [noti object];
    UIButton* b = [self buttonWithTag:[vo objectId]];
    [b setSelected:[vo isSelected]];
}

-(UIButton*)buttonWithTag:(int)tag
{
    for (UIButton* b in _buttons)
    {
        if ([b tag] == tag) return b;
    }
    return nil;
}

-(void)startTimer
{
    _updateTimer = [NSTimer timerWithTimeInterval:UPDATE_INTERVAL
target:self selector:@selector(tick:) userInfo:nil repeats:YES];
    [self setTimer:_updateTimer];
    [[NSRunLoop mainRunLoop] addTimer:self.timer
forMode:NSRunLoopCommonModes];
}

-(void)stopTimer
{
    [_updateTimer invalidate];
    _updateTimer = nil;
}

-(void)tick:(NSTimer*)theTimer
{
    double currentTime;
    double updateIterations;

    currentTime = CFAbsoluteTimeGetCurrent();
    updateIterations = ((currentTime - _lastFrameTime) +
_cyclesLeftOver);

    if (updateIterations > (MAX_CYCLES_PER_FRAME * UPDATE_INTERVAL))
    {
        updateIterations = (MAX_CYCLES_PER_FRAME * UPDATE_INTERVAL);
    }
    float currentRot = self.rotation;
    while (updateIterations > UPDATE_INTERVAL)
    {
        updateIterations -= UPDATE_INTERVAL;

        // Update game state a variable number of times
        if (_isDragging == NO)
        {
            currentRot += _rv;
            _rv *= DAMPING;
            [self setRotation:currentRot];
//            [self constrainScroll];
        }
        else
        {
            _rv = currentRot - _lastRotation;
            _rv = MAX(MIN(_rv , MAX_RV), -MAX_RV);
            _lastRotation = currentRot;
        }
    }

    _cyclesLeftOver = updateIterations;
    _lastFrameTime = currentTime;
}

/*
-(void)spin:(BOOL)clockwise
{
    [self stopTimer];
    float pos = (clockwise) ? self.rotation + M_PI : self.rotation -
M_PI;
    void (^anim) (void) = ^{
        [self setRotation:pos];
    };

    void (^complete) (BOOL) = ^(BOOL didComplete){
        [self spinComplete:clockwise];
    };

    [UIView animateWithDuration:0.75 delay:0
options:UIViewAnimationCurveEaseIn animations:anim
completion:complete];
}
*/

-(void)spin:(BOOL)clockwise
{
    _rv = ([self scrollPos] > 0.5) ? -MAX_RV : MAX_RV;
}
```

```
-(void)spinComplete:(BOOL)clockwise
{
    [self stopTimer];
    float pos = (clockwise) ? self.rotation + M_PI : self.rotation -
M_PI;
    void (^anim) (void) = ^{
        [self setRotation:pos];
    };

    void (^complete) (BOOL) = ^(BOOL didComplete){
        [self startTimer];
    };

    [UIView animateWithDuration:0.75 delay:0
options:UIViewAnimationOptionCurveEaseOut animations:anim
completion:complete];
}

-(float)scrollPos
{
    return fabsf([self rotation] / (_containerArc - _contentArc));
}

@end

//
//  CircleView.h
//  Curate
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>
#import <QuartzCore/QuartzCore.h>
#import "CircleModel.h"
#import "CircleVO.h"

@interface CircleView : UIView

-(void)draw;
-(CALayer*)layerWithTag:(int)tag;

@end

//
//  CircleView.m
//  Curate
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
```

```
//

#import "CircleView.h"

@implementation CircleView
{
    NSMutableArray* _circleLayers;
}

-(id)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];
    if (self)
    {
        [self setBackgroundColor:[UIColor colorWithWhite:1.0
alpha:0.0]];
        [self draw];
    }
    return self;
}

- (void)draw
{
    _circleLayers = [[NSMutableArray alloc] init];
    CAShapeLayer* arcLayer = nil;
    UIBezierPath *arc = nil;

    for (CircleVO* d in [[CircleModel sharedInstance] items])
    {
/*        arc = [UIBezierPath bezierPathWithArcCenter:[d pos]
                                              radius:[d radius]
                                          startAngle:0.0
                                            endAngle:M_PI
                                           clockwise:YES];*/
        arc = [UIBezierPath bezierPathWithOvalInRect:CGRectMake([d
pos].x - [d radius], [d pos].y - [d radius], [d radius]*2, [d
radius]*2)];

        arcLayer = [CAShapeLayer layer];
        arcLayer.path = arc.CGPath;

        arcLayer.strokeColor = [[d color] CGColor];

        arcLayer.fillColor = [[UIColor colorWithWhite:1.0 alpha:0.0]
CGColor];
        arcLayer.lineWidth = 50;
        [_circleLayers addObject:arcLayer];
        [self.layer addSublayer:arcLayer];
    }
}
```

```
-(CALayer*)layerWithTag:(int)tag
{
    return [_circleLayers objectAtIndex:tag];
}

@end

//
//  DockView.h
//  Curate
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "KeywordVO.h"

@interface DockView : UIView

@property (nonatomic, readonly) UIButton *viewButton;

-(UIButton*)createButtonWithVO:(KeywordVO*)vo;
-(UIColor*)colourForVO:(KeywordVO*)vo;
-(UIColor*)darkColourForVO:(KeywordVO*)vo;
-(void)updateButtons:(NSArray*)vos;
-(void)onViewClick:(id)sender;
-(void)onShuffleClick:(id)sender;
-(void)showButton:(UIButton*)button delay:(NSTimeInterval)delay;
-(void)showButtonSettle:(UIButton*)button;
-(void)hideButton:(UIButton*)button;
-(void)postClickNotificationFromButton:(UIButton*)button;

@end

//
//  DockView.m
//  Curate
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "DockView.h"
#import "notifications.h"
#import "CircleModel.h"
#import "KeywordModel.h"
#import <QuartzCore/QuartzCore.h>
#import "UIFont+NTF.h"
#import "UIButton+NTF.h"
#import "UIView+Position.h"
```

```
static const float removeButtonTime = 0.1;

@implementation DockView
{
    UIButton* _shuffleButton;
    UIButton* _exploreButton;
    UIImageView* _dockBg;
    NSMutableArray* _buttons;
    NSMutableArray* _buttonPositions;
}

- (id)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];
    if (self)
    {
        _dockBg = [[UIImageView alloc] initWithImage:[UIImage
imageNamed:@"dock_bg"]];
        [_dockBg setFrame:CGRectMake(0.0, 632.0, 1024.0, 136.0)];
        [_dockBg.layer setShadowColor:[UIColor blackColor].CGColor];
        [_dockBg.layer setShadowOpacity:0.2];
        [_dockBg.layer setShadowRadius:5];
        [_dockBg.layer setShadowOffset:CGSizeMake(0.0f, 0.0f)];

        _viewButton = [UIButton viewButton];
        [_viewButton setPosition:CGPointMake(915.0, 658.0)];
        [_viewButton addTarget:self action:@selector(onViewClick:)
forControlEvents:UIControlEventTouchUpInside];

        _shuffleButton = [UIButton shuffleButton];
        [_shuffleButton setPosition:CGPointMake(939.0, 573.0)];
        [_shuffleButton addTarget:self
action:@selector(onShuffleClick:)
forControlEvents:UIControlEventTouchUpInside];

        _exploreButton = [UIButton exploreButton];
        [_exploreButton setPosition:CGPointMake(25.0, 658.0)];
        [_exploreButton addTarget:self
action:@selector(onExploreClick:)
forControlEvents:UIControlEventTouchUpInside];

        float topRow = 673.0;
        float bottomRow = 728.0;
        float startX = 220.0;
        float endX = 800.0;
        float padding = (endX - startX) / 7;
        _buttonPositions = [[NSMutableArray alloc] init];
        for (int i = 0; i<8; i++)
        {
            float rowY = (i%2 == 0) ? topRow : bottomRow;
            [_buttonPositions addObject:[NSValue
valueWithCGPoint:CGPointMake(startX + padding * i, rowY)]];
        }

        [self addSubview:_dockBg];
        [self addSubview:_viewButton];
//        [self addSubview:_shuffleButton];
        [self addSubview:_exploreButton];
    }
    return self;
}

-(UIButton*)createButtonWithVO:(KeywordVO*)vo
{
    UIButton* b= [UIButton buttonWithType:UIButtonTypeCustom];
    [b setTitleColor:[self colourForVO:vo]
forState:UIControlStateNormal];
    [b setTitleColor:[self darkColourForVO:vo]
forState:UIControlStateHighlighted];
    [b.titleLabel setFont:[UIFont boldApplicationFontOfSize:18.0]];
    [b setTitle:[vo title] forState:UIControlStateNormal];
    [b setTag:[vo objectId]];
    [b sizeToFit];
    if (b.frame.size.width > 160)
    {
        [b.titleLabel setNumberOfLines:0];
        [b.titleLabel setLineBreakMode:UILineBreakModeWordWrap];
        [b.titleLabel setTextAlignment:UITextAlignmentCenter];
        [b setFrame:CGRectMake(0.0, 0.0, 160, 40)];
    }
    [b setAlpha:0.0];
    CGAffineTransform tr = CGAffineTransformScale(b.transform, 0.5,
0.5);
    [b setTransform:tr];
    [b addTarget:self action:@selector(buttonClicked:)
forControlEvents:UIControlEventTouchUpInside];
    [_buttons addObject:b];
    return b;
}

-(void)updateButtons:(NSArray*)vos
{
    for (UIButton* b in _buttons)
    {
        [b removeTarget:self action:@selector(buttonClicked:)
forControlEvents:UIControlEventTouchUpInside];
        [b removeFromSuperview];
    }
    _buttons = nil;
    _buttons = [[NSMutableArray alloc] init];

    UIButton* thisButton = nil;
    KeywordVO* vo = nil;
    for (int i=0; i<[vos count]; i++)
    {
        vo = [vos objectAtIndex:i];
        thisButton = [self createButtonWithVO:vo];
        CGPoint p = [[_buttonPositions objectAtIndex:i]
CGPointValue];
        int perturbX = -10 + arc4random() % 21;
        int perturbY = -10 + arc4random() % 21;
        [thisButton setCenter:CGPointMake(p.x + perturbX, p.y +
perturbY)];
        [self addSubview:thisButton];
        [self showButton:thisButton delay:i/50.0];
    }
}

-(void)showButton:(UIButton*)button delay:(NSTimeInterval)delay
{
    void (^anim) (void) = ^{
        [button setAlpha:1];

        [button
setTransform:CGAffineTransformScale(button.transform, 2.5, 2.5)];
    };

    void (^complete) (BOOL) = ^(BOOL didComplete){

        [self showButtonSettle:button];
    };

    [UIView animateWithDuration:0.10 delay:delay
options:UIViewAnimationOptionCurveEaseIn animations:anim
completion:complete];
}

-(void)showButtonSettle:(UIButton *)button
{
    void (^anim) (void) = ^{
        [button setTransform:CGAffineTransformMakeTranslation(1.0,
1.0)];
    };

    [UIView animateWithDuration:0.015 delay:0
options:UIViewAnimationOptionCurveEaseIn animations:anim
completion:nil];
}
```

```
-(void)hideButton:(UIButton*)button
{
    void (^anim) (void) = ^{
        [button setAlpha:0];
        [button
setTransform:CGAffineTransformScale(button.transform, 0.5, 0.5)];
    };

    void (^complete) (BOOL) = ^(BOOL didComplete){

    };

    [UIView animateWithDuration:removeButtonTime delay:0
options:UIViewAnimationOptionCurveEaseIn animations:anim
completion:complete];
}

-(UIColor*)colourForVO:(KeywordVO*)vo
{
    NSArray* a = [[KeywordModel sharedModel] categories];
    for (int i = 0; i<[a count]; i++)
    {
        for (KeywordVO* itemVO in [a objectAtIndex:i])
        {
            if (itemVO == vo)
            {
                CircleVO* cvo = [[[CircleModel sharedInstance]
items] objectAtIndex:i];
                return cvo.color;
            }
        }
    }
    return [UIColor blackColor];
}

-(UIColor*)darkColourForVO:(KeywordVO*)vo
{
    NSArray* a = [[KeywordModel sharedModel] categories];
    for (int i = 0; i<[a count]; i++)
    {
        for (KeywordVO* itemVO in [a objectAtIndex:i])
        {
            if (itemVO == vo)
            {
                CircleVO* cvo = [[[CircleModel sharedInstance]
items] objectAtIndex:i];
                return cvo.darkColor;
            }
        }
    }
    return [UIColor blackColor];
}

-(void)buttonClicked:(id)sender
{
    UIButton* b = (UIButton*)sender;
    [self hideButton:b];
    [self
performSelector:@selector(postClickNotificationFromButton:)
withObject:b afterDelay:removeButtonTime];
}

-(void)postClickNotificationFromButton:(UIButton*)button
{
    [[NSNotificationCenter defaultCenter]
postNotificationName:BUTTON_CLICK object:[NSNumber numberWithInt:
[button tag]]];
}

-(void)onViewClick:(id)sender
{
    [[NSNotificationCenter defaultCenter]
postNotificationName:SUBMIT_KEYWORDS object:nil];
}

-(void)onShuffleClick:(id)sender
{
    [[NSNotificationCenter defaultCenter]
postNotificationName:SHUFFLE object:nil];
}

-(void)onExploreClick:(id)sender
{
    [[NSNotificationCenter defaultCenter]
postNotificationName:SHOW_EXPLORE object:nil];
}


@end

//
//  ChapterViewContoller.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "ChapterView.h"
```

```
#import "ChapterVO.h"
#import "PetalLayer.h"
#import "ChapterThumbnailDelegate.h"
#import "ChapterDetailDelegate.h"

@interface ChapterViewController : UIViewController
<ChapterThumbnailDelegate, UIScrollViewDelegate,
ChapterDetailDelegate>

@property (nonatomic, readonly, weak) ChapterView *chapterView;
@property (nonatomic, strong) ChapterVO *vo;
@property (nonatomic) PetalType petalType;

//-(id)initWithChapterVO:(ChapterVO*)vo;

@end

//
//  ChapterViewContoller.m
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "ChapterViewController.h"
#import "ImageModel.h"
#import "UIImage+AspectRatio.h"
#import "UIView+Position.h"
#import "randomFloat.h"
#import <QuartzCore/QuartzCore.h>
#import "NTFServices.h"
#import "ChapterModel.h"
#import "ChapterThumbnailView.h"
#import "ChapterDetailViewController.h"
#import "CurateViewController.h"
#import "notifications.h"
#import "UIScreen+CKAdditions.h"
#import "UIColor+Color255.h"
#import "NSDateFormatter+NTF.h"
@interface ChapterViewController ()

@end

@implementation ChapterViewController
{
    BOOL _shouldRepopulateView;
}

- (id)init
{
    return [super initWithNibName:nil bundle:nil];
}

- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle
*)nibBundleOrNil
{
    return [self init];
}

- (id)initWithCoder:(NSCoder *)aDecoder
{
    return [self init];
}

-(void)loadView
{
    [self setView:[[ChapterView alloc] initWithFrame:[[UIScreen
mainScreen] landscapeRect]]];
    _chapterView = (ChapterView*)self.view;
    [_chapterView.exploreButton addTarget:self
action:@selector(exploreBtnTouch:)
forControlEvents:UIControlEventTouchUpInside];
    [_chapterView.curateButton addTarget:self
action:@selector(curateBtnTouch:)
forControlEvents:UIControlEventTouchUpInside];
    [_chapterView.nextButton addTarget:self
action:@selector(nextBtnTouch:)
forControlEvents:UIControlEventTouchUpInside];
    [_chapterView.scrollView setDelegate:self];
}

- (void)viewDidLoad
{
    [super viewDidLoad];
      // Do any additional setup after loading the view.
}

-(void)viewWillAppear:(BOOL)animated
{
    [super viewWillAppear:animated];

    if (!_shouldRepopulateView) return;

    [self populateView];
    _shouldRepopulateView = NO;
}

-(void)viewDidDisappear:(BOOL)animated
{
    [super viewDidDisappear:animated];
```

```
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

-(void)exploreBtnTouch:(id)sender
{
    [[NSNotificationCenter defaultCenter]
postNotificationName:NAVIGATE_TO_EXPLORE object:self];
}

-(void)curateBtnTouch:(id)sender
{
    [[NSNotificationCenter defaultCenter]
postNotificationName:NAVIGATE_TO_CURATE object:self];
}

-(void)nextBtnTouch:(id)sender
{
    [self setVo:[[ChapterModel sharedInstance] chapterAfter:_vo]];
    [self populateView];
}

-(void)setVo:(ChapterVO *)vo
{
    if (_vo != vo) _shouldRepopulateView = YES;
    _vo = vo;
}

-(void)populateView
{
    if (!_vo) return;

    [_chapterView removeImages];

    [[NTFServices dataService] chapterWasViewed:_vo];

    NSString *nameString = @"";
    if (_vo.username.length > 0)
    {
        nameString = [NSString
stringWithFormat:NSLocalizedString(@"curatedBy", nil) ,
_vo.username];
    }

    [_chapterView.nameLabel setText:nameString];
    [_chapterView.nameLabel sizeToFit];
    NSDateFormatter *df = [NSDateFormatter
dateFormatterWithFormat:@"dd.MM.yyyy"];
    NSString *dateString = [df stringFromDate:_vo.createdAt];
    NSString *locationString = [_vo.locationForCurrentLanguage
copy];
    if (locationString.length > 0)
    {
        _chapterView.dateLabel.text = [locationString
stringByAppendingString:[NSString stringWithFormat:@", %@" ,
dateString]];
    }
    else
    {
        _chapterView.dateLabel.text = dateString;
    }
    [_chapterView.dateLabel sizeToFit];
    [_chapterView setNeedsLayout];

    NSArray *imageVOs = [[ImageModel sharedModel]
getImagesWithIds:_vo.imageIds];
    [[NTFServices dataService] incrementImageViews:imageVOs];

    if (!_vo.layout || _vo.layout.count == 0)
    {
        [_chapterView layoutImages:imageVOs];
        [_vo setLayout:[_chapterView getThumbLayout]];
        [[NTFServices dataService] updateChapter:_vo];
    }
    else
    {
        [_chapterView layoutImages:imageVOs withRects:_vo.layout];
    }

    [_chapterView layoutKeywords:[_vo.keywords
valueForKey:@"title"]];

    for (ChapterThumbnailView *ctv in _chapterView.thumbnailViews)
    {
        [ctv setDelegate:self];
    }
    [_chapterView setPetalLayer:[PetalLayer
monoChromePetalLayerOfType:_petalType withKeywords:_vo.keywords]];
}


#pragma mark - ChapterThumbnailDelegate

-(void)thumbnailWasClicked:(id)thumb
{
    ChapterDetailViewController *cdv = [[ChapterDetailViewController
alloc] initWithVO:_vo];
    [cdv setDelegate:self];
    [cdv
setModalTransitionStyle:UIModalTransitionStyleCrossDissolve];
    NSUInteger i = [_chapterView.thumbnailViews
indexOfObject:thumb];
    cdv.selectedIndex = i;
    [self presentViewController:cdv animated:YES completion:NULL];
}


#pragma mark - UIScrollViewDelegate

-(void)scrollViewDidScroll:(UIScrollView *)scrollView
{
    if (scrollView.contentOffset.x <= -200.0f)
    {
        [self.navigationController popViewControllerAnimated:YES];
    }
}


#pragma mark - ChapterDetailDelegate

-(void)closeDetailViewAtIndex:(int)index
{
    UIView *v = _chapterView.thumbnailViews[index];
    float preferredXOffset = v.x - (self.view.frame.size.width -
v.frame.size.width)/2.0f;
    if (preferredXOffset < 0.0f) preferredXOffset = 0.0f;
    if (preferredXOffset > _chapterView.scrollView.contentSize.width
- self.view.frame.size.width)
    {
        preferredXOffset = _chapterView.scrollView.contentSize.width
- self.view.frame.size.width;
    }
    [_chapterView.scrollView
setContentOffset:CGPointMake(preferredXOffset, 0.0f)];
    [self dismissViewControllerAnimated:YES completion:nil];
}

@end

//
//  ChapterView.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//
```

```
#import <UIKit/UIKit.h>
#import "PetalLayer.h"

@interface ChapterView : UIView

@property (nonatomic, readonly) UIScrollView *scrollView;
@property (nonatomic, readonly) UIView *contentView;
@property (nonatomic, readonly) UIButton *exploreButton;
@property (nonatomic, readonly) UIButton *curateButton;
@property (nonatomic, readonly) UIButton *nextButton;
@property (nonatomic, readonly) NSMutableArray *thumbnailViews;
@property (nonatomic, readonly) UILabel *nameLabel;
@property (nonatomic, readonly) UILabel *dateLabel;
@property (nonatomic) PetalLayer *petalLayer;

-(void)layoutImages:(NSArray*)images;
-(void)layoutImages:(NSArray*)images withRects:
(NSArray*)layoutFrames;
-(void)layoutKeywords:(NSArray*)keywords;
-(void)removeImages;
-(NSArray*)getThumbLayout;

@end


//
//  ChapterView.m
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "ChapterView.h"
#import "UIButton+NTF.h"
#import "UIView+Position.h"
#import "UIFont+NTF.h"
#import "UIColor+Color255.h"
#import "ImageVO.h"
#import "UIImage+AspectRatio.h"
#import "UIImage+ProportionalFill.h"
#import "randomFloat.h"
#import <QuartzCore/QuartzCore.h>
#import "ChapterThumbnailView.h"

static const float startX = 533.0f;
static const float colWidth = 341.0f;
static const float row1 = 44.0f;
static const float row2 = 284.0f;
static const float row3 = 524.0f;
static const float row4 = 724.0f;
static const float padding = 41.0f;
static const float smallWidth = 300.0f;
static const float largeWidth = 641.0f;
static const float smallHeight = 199.0f;
static const float largeHeight = 439.0f;
static const float buttonPadding = 200.0f;

static const CGSize landscapeSm = {smallWidth , smallHeight};
static const CGSize landscapeLg = {largeWidth , largeHeight};
static const CGSize portrait    = {smallWidth , largeHeight};

@implementation ChapterView
{
    UIImageView *_previousImageView;
}

- (id)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];
    if (self)
    {
        UIColor *col = [UIColor colorWithPatternImage:[UIImage
imageNamed:@"chapter_bg"]];
        [self setBackgroundColor:col];
        [self setUserInteractionEnabled:YES];

        CGRect f = frame;
        f.size.width *= 3.0;
        _contentView = [[UIView alloc] initWithFrame:f];
        [_contentView setBackgroundColor:col];

        _scrollView = [[UIScrollView alloc] initWithFrame:frame];
        [_scrollView addSubview:_contentView];
        [_scrollView setContentSize:_contentView.frame.size];
        [_scrollView setScrollEnabled:YES];
        [_scrollView setDelaysContentTouches:NO];
        [_scrollView setUserInteractionEnabled:YES];

        _exploreButton = [UIButton exploreButton];
        _curateButton = [UIButton curateButton];
        _nextButton = [UIButton nextButton];

        _nameLabel = [[UILabel alloc]
initWithFrame:CGRectMake(39.0f, 617.0f, 400.0f, 88.0f)];
        [_nameLabel setBackgroundColor:[UIColor clearColor]];
        [_nameLabel setFont:[UIFont
boldItalicApplicationFontOfSize:18.0f]];
        [_nameLabel setTextColor:[UIColor
colorWithHexRGB:0x333333]];
```

```
        [_nameLabel setNumberOfLines:1];

        _dateLabel = [[UILabel alloc]
initWithFrame:CGRectMake(39.0f, 639.0f, 400.0f, 88.0f)];
        [_dateLabel setBackgroundColor:[UIColor clearColor]];
        [_dateLabel setFont:[UIFont
italicApplicationFontOfSize:18.0]];
        [_dateLabel setTextColor:[UIColor
colorWithHexRGB:0x333333]];

        [_contentView addSubview:_nameLabel];
        [_contentView addSubview:_dateLabel];
        [_contentView addSubview:_exploreButton];
        [_contentView addSubview:_curateButton];

        [self addSubview:_scrollView];
    }
    return self;
}

-(void)layoutSubviews
{
    if (_nameLabel.text.length > 0)
    {
        [_dateLabel setY:_nameLabel.bottom];
    }
    else
    {
        [_dateLabel setY:_nameLabel.y];
    }
}

-(void)layoutImages:(NSArray*)images
{
    [self removeImages];
    _thumbnailViews = [[NSMutableArray alloc] init];
    _previousImageView = nil;

    CGSize size = CGSizeZero;
    UIImage *im = nil;
    ChapterThumbnailView *imv = nil;
    int numLarge = 0;
    for (ImageVO *vo in images)
    {
        im = vo.image;
        if ([im isLandscape])
        {
            if (randomFloat() > 0.5 && numLarge < 2)
            {
                size = landscapeLg;
                numLarge++;
            }
            else
            {
                size = landscapeSm;
            }
        }
        else
        {
            size = portrait;
        }

        imv = [[ChapterThumbnailView alloc] initWithImage:[im
imageCroppedToFitSize:size]];

        [imv setPosition:[self positionForImage:imv
withPreviousImage:_previousImageView]];
        [_thumbnailViews addObject:imv];
        [_contentView addSubview:imv];

        _previousImageView = imv;
    }

    [_exploreButton setPosition:CGPointMake(_previousImageView.right
+ padding + buttonPadding, 534.0)];
    [_curateButton setPosition:CGPointMake(_exploreButton.x,
639.0)];
    [_contentView setFrame:CGRectMake(0.0, 0.0, _exploreButton.right
+ padding, _contentView.frame.size.height)];
    [_scrollView setContentSize:_contentView.frame.size];
}

-(void)layoutImages:(NSArray*)images withRects:
(NSArray*)layoutFrames
{
    [self removeImages];

    _thumbnailViews = [[NSMutableArray alloc] init];

    ChapterThumbnailView *imv = nil;
    UIImage *im = nil;
    NSString *s = nil;
    CGRect r;
    for (uint i=0; i<images.count; i++)
    {
        s = [layoutFrames objectAtIndex:i];
        r = CGRectFromString(s);
        im = [(ImageVO*)[images objectAtIndex:i] image];
        imv = [[ChapterThumbnailView alloc] initWithImage:[im
imageToFitSize:r.size method:MGImageResizeCrop]];
```

```
        [imv setPosition:r.origin];
        [_thumbnailViews addObject:imv];
        [_contentView addSubview:imv];
    }

    [_exploreButton setPosition:CGPointMake(imv.right + padding +
buttonPadding, 534.0)];
    [_curateButton setPosition:CGPointMake(_exploreButton.x,
639.0)];
    [_contentView setFrame:CGRectMake(0.0, 0.0, _exploreButton.right
+ padding, _contentView.frame.size.height)];
    [_scrollView setContentSize:_contentView.frame.size];
}

-(void)layoutKeywords:(NSArray*)keywords
{
    UILabel *l = nil;
    UILabel *previousLabel = nil;
    for (NSString *s in keywords)
    {
        l = [[UILabel alloc] initWithFrame:CGRectMake(startX + 300.0
- 500.0f, row1, 500.0f, 45.0f)];
        [l setFont:[UIFont italicSystemFontOfSize:40.0f]];
        [l setBackgroundColor:[UIColor clearColor]];
        [l setTextColor:[UIColor colorWithWhite:0.0f alpha:0.2f]];
        // add a space because the letters are getting cut off on
the right
        [l setText:[s stringByAppendingString:@" "]];
        [l setTextAlignment:NSTextAlignmentRight];
        l.y = (previousLabel) ? CGRectGetMaxY(previousLabel.frame) :
row1;
        previousLabel = l;
        [_contentView addSubview:l];
    }
}

-(void)removeImages
{
    if (!_thumbnailViews) return;

    for (UIView *iv in _thumbnailViews)
    {
        [iv removeFromSuperview];
    }
    _thumbnailViews = nil;
}

-(CGPoint)positionForImage:(UIImageView*)image withPreviousImage:
(UIView*)previousImage
```

```
{
    CGPoint out = CGPointMake(startX, row1);
    if (!previousImage)
    {
        previousImage = [[UIView alloc]
initWithFrame:CGRectMake(startX, row1, 300.0f, 439.0f)];
    }

    if (previousImage.bottom + padding + image.frame.size.height >=
row4 || previousImage.frame.size.width > 300.0 ||
image.frame.size.width > 300.0)
    {
        out.x = previousImage.right + padding;
        if (image.frame.size.height > 199.0 && randomFloat() > 0.5)
        {
            out.y = row2;
        }
    }
    else
    {
        out.x = previousImage.x;
        out.y = previousImage.bottom + padding;
    }

    // if the image is small landscape theres a 25% chance of a gap
    if (image.frame.size.height == 199.0 && out.y <= row2 &&
randomFloat() < 0.25)
    {
        out.y += 199.0 + padding;
    }

    return out;
}

-(void)setPetalLayer:(PetalLayer *)petalLayer
{
    if (_petalLayer)
    {
        [_petalLayer removeFromSuperlayer];
        _petalLayer = nil;
    }

    _petalLayer = petalLayer;
    [_petalLayer setPosition:CGPointMake(176.0, 66.0)];

    float scale = 1311.0/_petalLayer.frame.size.width;
    CGAffineTransform t1 = CGAffineTransformMakeScale(scale, scale);
    CGAffineTransform t2 = CGAffineTransformMakeRotation(M_PI*0.75);
    [_petalLayer setAffineTransform:CGAffineTransformConcat(t1,
t2)];
```

```
    [_petalLayer setOpacity:0.75];
    [self.contentView.layer insertSublayer:_petalLayer atIndex:0];
}

-(NSArray*)getThumbLayout
{
    NSMutableArray *a = [[NSMutableArray alloc] init];
    for (ChapterThumbnailView *th in _thumbnailViews)
    {
        [a addObject:NSStringFromCGRect(th.frame)];
    }
    return a;
}

@end

//
//  ChapterThumbnailView.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "ChapterThumbnailDelegate.h"

@interface ChapterThumbnailView : UIImageView

@property (nonatomic , weak) id<ChapterThumbnailDelegate> delegate;

@end

//
//  ChapterThumbnailView.m
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "ChapterThumbnailView.h"
#import <QuartzCore/QuartzCore.h>
#import "UIImage+RoundedCorner.h"

@implementation ChapterThumbnailView

- (id)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];
    if (self) {
        // Initialization code
    }
    return self;
}

-(id)initWithImage:(UIImage *)image
{
    self = [super initWithImage:[image roundedCornerImage:7
borderSize:0]];
    if (self)
    {
        self.layer.shadowOffset = CGSizeMake(0.0, 0.0);
        self.layer.shadowColor = [UIColor blackColor].CGColor;
        self.layer.shadowRadius = 5.0;
        self.layer.shadowOpacity = 0.35;
        self.layer.shadowPath = [[UIBezierPath
bezierPathWithRect:self.layer.bounds] CGPath];
        self.layer.shouldRasterize = YES;
        self.layer.cornerRadius = 7.0f;

        [self setUserInteractionEnabled:YES];
    }

    return self;
}

-(void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event
{
    [_delegate thumbnailWasClicked:self];
}

/*
// Only override drawRect: if you perform custom drawing.
// An empty implementation adversely affects performance during
animation.
- (void)drawRect:(CGRect)rect
{
    // Drawing code
}
*/

@end




//
//  ChapterThumbnailDelegate.h
//  NostalgiaTiFrega
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
```

```
//

#import <Foundation/Foundation.h>

@protocol ChapterThumbnailDelegate <NSObject>

-(void)thumbnailWasClicked:(id)thumb;

@end

//
//  CKViewController.h
//  NTFChapterTest
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "ChapterDetailView.h"
#import "ChapterVO.h"
#import "ChapterDetailDelegate.h"
#import "PSTCollectionView.h"

@interface ChapterDetailViewController : UIViewController
<UITextViewDelegate, PSUICollectionViewDataSource ,
PSUICollectionViewDelegate , PSUICollectionViewDelegateFlowLayout ,
UIScrollViewDelegate>

@property (nonatomic, weak, readonly) ChapterDetailView *detailView;
@property (nonatomic, strong, readonly) ChapterVO *vo;
@property (nonatomic) NSUInteger selectedIndex;
@property (nonatomic, weak) id<ChapterDetailDelegate> delegate;

-(id)initWithVO:(ChapterVO*)vo;

@end

//
//  CKViewController.m
//  NTFChapterTest
//
//  Copyright (c) 2013 Carla Drago. All rights reserved.
//

#import "ChapterDetailViewController.h"
#import "UIScreen+CKAdditions.h"
#import "UIView+Position.h"
#import "UIImage+MetaData.h"
#import <ImageIO/ImageIO.h>
#import "ImageModel.h"
```

```
#import "NTFServices.h"
#import "NSObject+DispatchError.h"
#import "NTFServices.h"
#import "constants.h"
#import "notifications.h"
#import "NPReachability.h"
#import "ChapterDetailCell.h"

static NSString *cellID = @"cellID";

@interface ChapterDetailViewController ()

@end

@implementation ChapterDetailViewController
{
    NSArray *_imageVOs;
    UIPinchGestureRecognizer *_pinchRecognizer;
    UITapGestureRecognizer *_doubleTapGestureRecognizer;
    UITapGestureRecognizer *_tapGestureRecognizer;
    UISwipeGestureRecognizer *_swipeGestureRecognizer;
    NSMutableArray *_sizes;
}

- (id)initWithVO:(ChapterVO *)vo
{
    self =  [super initWithNibName:nil bundle:nil];
    if (self)
    {
        _vo = vo;
        _imageVOs = [[ImageModel sharedModel]
getImagesWithIds:_vo.imageIds];
        _pinchRecognizer = [[UIPinchGestureRecognizer alloc]
initWithTarget:self action:@selector(handlePinchOrDoubleTap:)];
        _doubleTapGestureRecognizer = [[UITapGestureRecognizer
alloc] initWithTarget:self
action:@selector(handlePinchOrDoubleTap:)];
        _doubleTapGestureRecognizer.numberOfTapsRequired = 2;
        _tapGestureRecognizer = [[UITapGestureRecognizer alloc]
initWithTarget:self action:@selector(handleImageTapped:)];
        _swipeGestureRecognizer = [[UISwipeGestureRecognizer alloc]
initWithTarget:self action:@selector(handleImageTapped:)];
        _swipeGestureRecognizer.direction =
UISwipeGestureRecognizerDirectionRight;

        [self calculateImageSizes];
    }
    return self;
}
```

```
-(id)init
{
    return [self initWithVO:nil];
}

-(id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle
*)nibBundleOrNil
{
    return [self initWithVO:nil];
}

-(void)loadView
{
    [self setView:[[ChapterDetailView alloc] initWithFrame:
[[UIScreen mainScreen] currentRect]]];
    _detailView = (ChapterDetailView*)self.view;
}

-(void)viewDidUnload
{
    [self.view removeGestureRecognizer:_pinchRecognizer];
    [self.view removeGestureRecognizer:_doubleTapGestureRecognizer];
    [self.view removeGestureRecognizer:_swipeGestureRecognizer];
}

- (void)viewDidLoad
{
    [super viewDidLoad];

    [_detailView addGestureRecognizer:_pinchRecognizer];
    [_detailView addGestureRecognizer:_doubleTapGestureRecognizer];
    [_detailView addGestureRecognizer:_swipeGestureRecognizer];

    [_detailView.commentsBtn addTarget:self
action:@selector(commentsBtnTouch:)
forControlEvents:UIControlEventTouchUpInside];

    [_detailView.commentsView.commentTextView setDelegate:self];
    [_detailView.commentsView.addCommentBtn addTarget:self
action:@selector(addCommentButtonTapped:)
forControlEvents:UIControlEventTouchUpInside];
    [_detailView.commentsView.nameTextField setText:[[NSUserDefaults
standardUserDefaults] objectForKey:USERNAME]];

    [_detailView.collectionView
addGestureRecognizer:_tapGestureRecognizer];
    [_detailView.collectionView setDelegate:self];
    [_detailView.collectionView setDataSource:self];
    [_detailView.collectionView registerClass:[ChapterDetailCell
class] forCellWithReuseIdentifier:cellID];
```

```
}

-(void)viewWillAppear:(BOOL)animated
{
    if (_selectedIndex > 0)
    {
        NSIndexPath *ip = [NSIndexPath
indexPathForRow:_selectedIndex inSection:0];
        [_detailView.collectionView scrollToItemAtIndexPath:ip

atScrollPosition:PSTCollectionViewScrollPositionLeft
                                                   animated:NO];
    }

    [self
updateCommentsButtonWithImageVO:_imageVOs[_selectedIndex]];

    [[NSNotificationCenter defaultCenter] addObserver:self

selector:@selector(commentsLoaded:)

name:COMMENTS_LOADED
                                               object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self

selector:@selector(commentsLoadFailed:)

name:COMMENTS_LOAD_FAILED
                                               object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self

selector:@selector(commentsLoaded:)
                                                 name:COMMENT_ADDED
                                               object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self

selector:@selector(commentsLoadFailed:)

name:COMMENT_ADD_FAILED
                                               object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self

selector:@selector(commentsLoadFailed:)

name:COMMENT_ADD_FAILED
                                               object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(handleUsernameChanged:)

name:USERNAME_CHANGED
                                                     object:nil];
}

-(void)viewWillDisappear:(BOOL)animated
{
    [[NSNotificationCenter defaultCenter] removeObserver:self];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];

    if(self.isViewLoaded && self.view.window)
    {
        self.view = nil;
    }
}

-(void)dealloc
{
    _imageVOs = nil;
    _pinchRecognizer = nil;
    _doubleTapGestureRecognizer = nil;
    _tapGestureRecognizer = nil;
    _swipeGestureRecognizer = nil;
    [[NSNotificationCenter defaultCenter] removeObserver:self];
}

#pragma mark - rotation

-(BOOL)shouldAutorotateToInterfaceOrientation:
(UIInterfaceOrientation)toInterfaceOrientation
{
    return
UIInterfaceOrientationIsLandscape(toInterfaceOrientation);
}

// iOS >= 6
-(NSUInteger)supportedInterfaceOrientations
{
    return UIInterfaceOrientationMaskLandscape;
}

-(BOOL)shouldAutomaticallyForwardRotationMethods
{
    return YES;
}
```

```
#pragma mark - event handlers

-(void)commentsBtnTouch:(id)sender
{
    if ([[NPReachability sharedInstance] isCurrentlyReachable] ==
NO)
    {
        UIAlertView *av = [[UIAlertView alloc] initWithTitle:nil

message:NSLocalizedString(@"Please connect to the internet", nil)
                                                     delegate:self
                                            cancelButtonTitle:@"OK"
                                            otherButtonTitles:nil];
        [av show];

        return;
    }


    if (_detailView.commentsViewIsVisible)
    {
        [_detailView hideCommentsView];
    }
    else
    {
        [_detailView showCommentsView];
        NSDictionary *ui = @{IMAGE_VO:[self currentImageVO]};
        [[NSNotificationCenter defaultCenter]
postNotificationName:LOAD_COMMENTS object:nil userInfo:ui];
    }
}


#pragma mark - helpers

-(void)calculateImageSizes
{
    NSString *filePath = nil;
    NSArray *a = nil;
    float w , h;
    NSDictionary *metadata = nil;
    _sizes = [[NSMutableArray alloc]
initWithCapacity:_imageVOs.count];
    for (ImageVO *vo in _imageVOs)
    {
        a = [vo.fileName componentsSeparatedByString:@"."];
        filePath = [[NSBundle mainBundle] pathForResource:[a
objectAtIndex:0] ofType:[a objectAtIndex:1]];
```

```
        metadata = [UIImage metaDataForImageWithFilePath:filePath];
        w = [[metadata objectForKey:kPixelWidthKey] floatValue];
        h = [[metadata objectForKey:kPixelHeightKey] floatValue];
        [_sizes addObject:[NSValue valueWithCGSize:CGSizeMake(w,
h)]];
    }
}

-(UIView*)viewAtLeftEdge
{
    NSIndexPath *ip = [_detailView.collectionView
indexPathForItemAtPoint:_detailView.collectionView.contentOffset];
    return [self collectionView:_detailView.collectionView
cellForItemAtIndexPath:ip];
}

-(float)maxScroll
{
    static float ms = 0.0;
    if (ms == 0.0) ms = _detailView.collectionView.contentSize.width
- _detailView.collectionView.bounds.size.width;

    return ms;
}


#pragma mark - Comments

-(void)addCommentButtonTapped:(id)sender
{
    NSString *t = _detailView.commentsView.commentTextView.text;
    NSString *prompt = NSLocalizedString(@"Enter comment", nil);
    if (!t || t.length == 0 || [t isEqualToString:prompt])
    {
        UIAlertView *av = [[UIAlertView alloc] initWithTitle:nil

message:NSLocalizedString(@"Please enter your comment", nil)
                                                      delegate:self
                                             cancelButtonTitle:@"OK"
                                             otherButtonTitles:nil];
        [av show];
        return;
    }

    CommentVO *vo = [[CommentVO alloc] init];
    [vo setText:t];
    [vo setUsername:_detailView.commentsView.nameTextField.text];
    NSDictionary *ui = @{COMMENT_VO:vo , IMAGE_VO:[self
currentImageVO]};
    [[NSNotificationCenter defaultCenter]
```

```
postNotificationName:ADD_COMMENT object:nil userInfo:ui];

    [_detailView.commentsView.commentTextView setText:@""];


    [[NSNotificationCenter defaultCenter]
postNotificationName:CHANGE_USERNAME
                                                         object:self

userInfo:@{USERNAME:_detailView.commentsView.nameTextField.text}];


    [self.view endEditing:YES];
}

-(void)commentsLoaded:(NSNotification*)notification
{
    ImageVO *ivo = [notification.userInfo objectForKey:IMAGE_VO];

    if ([self currentImageVO] != ivo) return;
    [_detailView.commentsView hideCommentFields];
    [_detailView.commentsView setComments:ivo.comments];

    [self updateCommentsButtonWithCurrentImage];
}

-(void)commentsLoadFailed:(NSNotification*)notification
{
    [self dispatchError:[notification.userInfo objectForKey:ERROR]];
}

-(void)handleUsernameChanged:(NSNotification*)notification
{
    [_detailView.commentsView.nameTextField
setText:notification.userInfo[USERNAME]];
}

-(ImageVO*)currentImageVO
{
    NSIndexPath *ip = [_detailView.collectionView
indexPathForItemAtPoint:_detailView.collectionView.contentOffset];
    return [_imageVOs objectAtIndex:ip.row];
}

-(void)handlePinchOrDoubleTap:(UIGestureRecognizer*)recognizer
{
    // only close if double tap is off comments interface
    NSIndexPath *ip = [_detailView.collectionView
indexPathForItemAtPoint:_detailView.collectionView.contentOffset];
    [self.delegate closeDetailViewAtIndex:ip.row];
```

```
}

-(void)handleImageTapped:(UIGestureRecognizer*)recognizer
{
    if (_detailView.commentsViewIsVisible)
    {
        [_detailView hideCommentsView];
    }
}

-(void)updateCommentsButtonWithCurrentImage
{
    CGPoint offset = _detailView.collectionView.contentOffset;
    NSUInteger index = [[_detailView.collectionView
indexPathForItemAtPoint:offset] row];
    [self updateCommentsButtonWithImageVO:_imageVOs[index]];
}

-(void)updateCommentsButtonWithImageVO:(ImageVO*)vo
{
    if (vo.hasComments)
    {
        [_detailView.commentsBtn setImage:[UIImage
imageNamed:@"commentLinesBtnN"] forState:UIControlStateNormal];
        [_detailView.commentsBtn setImage:[UIImage
imageNamed:@"commentLinesBtnT"] forState:UIControlStateHighlighted];
    }
    else
    {
        [_detailView.commentsBtn setImage:[UIImage
imageNamed:@"commentBtnN"] forState:UIControlStateNormal];
        [_detailView.commentsBtn setImage:[UIImage
imageNamed:@"commentBtnT"] forState:UIControlStateHighlighted];
    }
}


#pragma mark - UITextFieldDelegate

-(void)textViewDidBeginEditing:(UITextView *)textView
{
    textView.text = @"";
}


#pragma mark - UICollectionView Datasource and Delegate

-(ChapterDetailCell *)collectionView:(PSTCollectionView
*)collectionView cellForItemAtIndexPath:(NSIndexPath *)indexPath
{
    ChapterDetailCell *cell = (ChapterDetailCell *)[collectionView
dequeueReusableCellWithReuseIdentifier:cellID

forIndexPath:indexPath];
    ImageVO *vo = _imageVOs[indexPath.row];
    cell.image = vo.image;

    return cell;
}

-(NSInteger)collectionView:(PSTCollectionView *)collectionView
numberOfItemsInSection:(NSInteger)section
{
    return _imageVOs.count;
}


#pragma mark - UICollectionViewDelegateFlowLayout

-(CGSize)collectionView:(UICollectionView *)collectionView layout:
(UICollectionViewLayout *)collectionViewLayout
sizeForItemAtIndexPath:(NSIndexPath *)indexPath
{
    CGSize s = [_sizes[indexPath.row] CGSizeValue];
    CGSize currentSize = self.view.bounds.size;
    if (s.width > s.height)
    {
        return currentSize;
    }
    return CGSizeMake(currentSize.width/2.0, currentSize.height);
}


#pragma mark - UIScrollViewDelegate

-(CGPoint)snap
{
    UIView *currentView = [self viewAtLeftEdge];
    float midX = CGRectGetMinX(currentView.frame) +
currentView.frame.size.width/2.0f;

    float targetX = 0.0f;
    if (_detailView.collectionView.contentOffset.x >= midX)
    {
        targetX = CGRectGetMaxX(currentView.frame) + 1.0;
    }
    else
    {
        targetX = CGRectGetMinX(currentView.frame);
    }
    return CGPointMake(targetX, 0.0f);
}

-(void)scrollViewWillEndDragging:(UIScrollView *)scrollView
withVelocity:(CGPoint)velocity targetContentOffset:(inout CGPoint
*)targetContentOffset
{
    ChapterDetailCell *currentView = (ChapterDetailCell*)[self
viewAtLeftEdge];
    if (velocity.x > 0.0)
    {
        targetContentOffset->x =
MIN((CGRectGetMaxX(currentView.frame) + 1.0) , [self maxScroll]);
    }
    else if (velocity.x < 0.0)
    {
        targetContentOffset->x = CGRectGetMinX(currentView.frame);
    }
    else
    {
        *targetContentOffset = [self snap];
    }
}

-(void)scrollViewDidEndDecelerating:(UIScrollView *)scrollView
{
    [self updateCommentsButtonWithCurrentImage];
}

@end
```

BIBLIOGRAPHY

Adams, R. Baltz, L. Becher, H. Deal, J. Gohlke, F. Nixon, N. Schott, J. Shore S. Wessel, H. 2009, *New Topographies*, Steidl & Centre for Creative Photography, Arizona

Alexenberg, M. 2011, *The Future of Art in the Post Digital Age*, Intellect Books/University of Chicago Press, Bristol and Chicago.

Anderson, R. and Russotti, P. 2009, *Digital Photography Best Practices and Workflow Handbook*, Focal Press, London

Ang, I. 1985, *Watching Dallas: Soap Opera and the Melodramatic Imagination*, Routledge, Oxford

Badger, G. 2007, *The Genius of Photography: How Photography Has Changed Our Lives*, Quadrille Publishing, UK

Baldassar, L. 2001, *Visits Home: Migration Experiences Between Italy and Australia*, Melbourne: Melbourne University Press.

Baldassar, L. 2007, "Transnational Families and the Provision of Emotional Support: The Relationship Between Truth and Distance," *Identities: Global Studies in Culture and Power, Vol. 14*

Baldassare, G. Fr. 1992*, Salaparuta Ieri e Oggi*, Comitato Communale per l'Emigrazione di Salaparuta, Salaparuta

Barabasi, A. 2003, *Linked: How Everything is Connected to Everything Else*. Plume.

Barthes, R. 1999, *Camera Lucida,* Farr Straus & Giroux Inc,NY

Batchen, G. 1999, *Burning With Desire: The Conception of Photography*, MIT Press, Massachusetts

BBC Online website, 2005, viewed 14 September, <http://news.bbc.co.uk/1/hi/entertainment/tv_and_radio/3020275.stm>

Beck, K. et al. 2001, "Manifesto for Agile Software Development". Agile Alliance <http://agilemanifesto.org/> Retrieved 14 June 2010

Bell, G. and Gemmel, J. 2009, *Total Recall: How the E-Memory Revolution*

Will Change Everything, Penguin, New York

Benjamin, W. 2006, *Berlin Childhood Around 1900*, Belknap Press, Boston

Big Brother website, 2005, viewed 16 August, 2005,
<http://bigbrother.3mobile.com.au/>

Boccafucci, T. 1975, *The Italian Joke Book,* Belmont Tower Books

Boot, C. 2004, *Magnum Stories,* Phaidon Press, London

Boyle, T. 1997, *Design for Multimedia Learning*. London: Prentice Hall.

Boym, S. 2001, *The Future of Nostalgia,* Basic Books, New York

Brand, S. 1995, *How Buildings Learn: What Happens After They're Built*, Penguin, London

BroadBananas website, 2005, viewed 8 September, 2005
<http://www.broadbandbananas.com>

Buxton, B. 2007, *Sketching User Experiences*, Morgan Kaufmann, San Francisco

Campbell, J. 1991, *The Power of Myth,* Anchor, New York

Canaletto (ongoing) *Reproduction of Canaletto's Camera Obscura*, Precinema Museum, Padua

Cartier-Bresson, H. 2007, *Scrapbook,* Thames & Hudson, London

Church, J. 2005, *Migration per l'Australia: The Story of Australian Migration,* Italian Historical Society & Miegunyah Press

Coalson, N. 2010, *Adobe Lightroom 3: Streamlining Your Digital Process,* Wiley Publishing, Indianapolis

Connerton, P. 2009, *How Modernity Forgets*, Cambridge University Press, Cambridge

Cooper, A. Cronin, D. and Reimann, R. 2007, *About Face: The Essentials of Interaction Design*, John Wiley & Sons, San Fansisco

Cranny-Francis, A. 2013, *Technology and Culture: the biopolitics of emerging technologies*, Palgrave, London

Creet, J. Kitzmann, A. Eds. 2011, *Memory and Migration: Multidisciplinary Approaches to Memory Studies,* University of Toronto.

Curran, S. 2003, *Convergence Design*. Glouscester, Rockport.

Danielewski, M. Z. 2000, *House of Leaves*. Doubleday, Bath.

de Saint Exupéry, A. 1948, *Citadelle*, Gallimard, Paris

De Landa, M. 1997, *A Thousand Years of Non-Linear History*. New York: Zone Books.

Dawes, B. 2006, *Analogue In, Digital Out*, New Riders, Berkeley

Delahaye, L. 2003, *Winterreise*, Phaidon Press, London

Deleuze, G. & Guattari, F. 1987, *A Thousand Plateaus: Capitalism and Schizophrenia*, Continuum Press Edition 2004, London.

Denne, M and Cleind-Huang, J. 2003, *Software by Numbers*, Prentice Hall, New Jersey

Dewhirst, C. 2014, "The 'Southern Question' in Australia: The 1925 Royal Commission's Radicalisation of Southern Italians", *Queensland History Journal*, vol.22, no.4

Dunne, A. 2005, *Hertzian Tales,* MIT Press, Cambridge

Dyer, G. 2006, *The Ongoing Moment*, Vintage, New York

Eggleston, W. 2010, *For Now*, (Almereyda Ed), Twin Palms Publishers, Santa Fe

Eggleston, W. 1976, *William Eggleston's Guide,* Hatje Cantz Verlag, Berlin

Ellis, C. Adams, T.E, & Bochner, A. P, 2011, "Autoethnography: An Overview", *Forum: Qualitative Research*, Vol 12, No 1.

Field, S. 2005, *Screenplay: The Foundations of Screenwriting,* Delta, New

York

Forlizzi, J. and Battarbee, K. 2004, *Understanding Experience in Interactive Systems*, Cambridge, MA: Association for Computing Machinery

Foster, H. Ed. 2002, The Anti-Aesthetic: Essays on Post-Modern Culture, The New Press, New York

Frank, R. 1998, *The Americans*, Scalo Publishers, Zurich

Freund, A. & Thomson, A (Eds), 2012, *Oral History and Photography*, Palgrace Macmillian, New York.

Fritz, R. 1989, *The Path of Least Resistance*. New York, Fawcett Columbine.

Gaetano, R. & Turcotte, G. (Eds) 2008, *Literary and Social Diasporas: An Italian Australian Perspective,* Peter Lang International, Brussels

Gefter, P. 2009, *Photography After Frank,* Aperture, NY

Georgiou, G. 2010, *Fault Lines/Turkey/East/*West, Schilt Publishing, Amsterdam

Gilbert, D. 2007, *Stumbling on Happiness*, Vintage, London

Graham, P. 2009, *A Shimmer of Possibility,* SteidlMack, Gottingen

Greenough, S. 2009, *Looking In: Robert Frank's The Americans,* Steidl and Museum of Washington, Gottingen

Guler, A. 2009, *Istanbul,* Les Editions Du Pacifique, Paris

Guynn, W. Ed. 2010, *The Routledge Companion to Film History*, Routledge, London

Igoe, T. 2007, *Making Things Talk: Practical Methods for Connecting Physical Objects,* O'Reilly Press, Sebastopol

Jameson, F. 1997 "Postmodernism, or the Cultural Logic of Late Capitalism" in *Rethinking Architecture: a reader in cultural theory*, by Neil Leach ed Routledge

Jameson, F. 1985, “Postmodernism and Consumer Society” in H.Forstered *Postmodern Culture*

Jacobs, J. 1961, *The Death and Life of Great American Cities,* New York: Random House.

Jenkins, H. 2006, *Convergence Culture: Where Old and New Media Collide*, New York University Press, New York

Jensen, J. F. 2005, *Interactive Television: New Genres, New Format, New Content*.

Johnson, B. S. 2007, *The Unfortunates,* New Directions, New York

Johnson, B. 2004, *Photography Speaks: 150 Photographers on their Art,* Aperture, New York

Johnson, S. 2002, *Emergence: The Connected Lives of Ants, Brains, Cities, and Software*.

Jones, M. 2012, “ Media-bodies and Photoshop”, in *Controversial Images: Media Representations on the Edge*, Palgrave McMillian, London

Juniper, A. 2003, *WabiSabi: The Japanese Art of Impermanence*, Tuttle Publishing, Tokyo

Kaneko, R. & Vartanian, I. 2009, *Japanese Photobooks of the1960s and 70s,* Aperture, New York

Karanfil, G. 2009 “Psuedo Exiles and Reluctant Trans Nationals” *Media Culture and* Society, SAGE, London

Kawauchi, R. 2005, *Cui Cui* FOIL, Fondation Cartier, ACT sud

Knuth, D. 1997, *The Art of Computer Programming*, *Volume One, Fundamental Algorithims*, Addison-Wesley Professional, London

Kramer, R. 2010, *Lost Track*, Neo Lux, Amsterdam

Kuhn, A. 2002, *Family Secrets: Acts of Memory & Imagination*, Verso Books, London

Kuhn, A. & McAllister, K. E. Eds. 2006, *Locating Memory - Photographic Acts*, Oxford, Berghan Books

Kwint, M. Breward, C, & Aynsle, J. (1999) *Material Memories: Design and Evocation,* Berg, Oxford

Kwon, M. 2004, *One Place After Another: Site Specific Art and Locational Identity,* MIT Press, Cambridge

Lawson, H. 2001, *Closure*. Routledge, London.

Leach, N. Ed. 1997, *Rethinking Architecture: a reader in cultural theory*, Routledge, New York

Leung, L. Ed. 2009, *Digital Experience Design*, Intellect Books, UK

Lippy, T. Ed *Esopus Magazine,* Esopus Foundation,New York

Lowgren, J. and Stolterman, E. 2005, *Thoughtful Interaction Design: A Design Perspective on Information Technology*, MIT Press, Massachusetts

Lu, K. 2005, *Principles of Interaction Design for iTV*. Part of a thesis presented to Georgia Institute of Technology, <http://idt.gatech.edu/ms_projects/klu/lu_karyn_y_200505_mast.pdf>

Lynch, D. 2006, *Catching the Big Fish*, Penguin, New York

Malcolm, D. 2012, *Globalizing Cricket: Englishness, Empire and Identity*, Bloomsbury, London

Mayer-Schonberger, V, 2009, *Delete: The Virtue of Forgetting in the Digital Age,*Princeton University Press, Princeton

McKee, R. 1997, *Story: Substance, structure, style, and the principles of screenwriting*. Methuen, Chatham, UK.

McLuhan, M. & Powers, B. 1992, *The Global Village: Transformations in World Life and Media in the 21st Century*, Oxford University Press, Oxford.

Miller, D. 2008, *The Comfort of Things,* Polity Press, Cambridge

Moggridge, B. 2007, *Designing Interactions*, MIT Press, Massachusetts

Motte, W. 1986, *Oulipo: A Primer of Potential* Literature, University of

Nebraska Press

Noble, G. 2004, “Accumulating Being” *International Journal of Cultural Studies,* SAGE, London

Noble, G. 2002, “Comfortable and Relaxed: furnishing the home and nation” *Continuum Journal of Media and Cultural* Studies, Carfax Publishing

Norman, D. A. 1998, *The Invisible Computer*. Cambridge, MIT Press

Norman, D. A. 2005, *Emotional Design,* Basic Books, NY

O’Grady, J. 1957, *They’re a Weird Mob: A Nove*l, Ure Smith, Sydney

Parr, M. 2002, *Last Resort: Photographs of New Brighton*, Dewi Lewis Publishing, UK

Patt, L. Ed. 2007, *Searching for Sebald: Photography After W. G. Sebald,* Institute of Cultural Enquiry, Los Angeles

Pettersson, M. 2011, “Depictive Traces: On the Phenomenology of Photography” *Journal of Aesthetics and Art Criticism* 69 (2)

Pilcher, R. 2010, *Agile Product Management with Scrum: Creating Products that Customers Love*, Addison-Wesley, London

Reis, E. 2011, *The Lean Startup*, Crown Business Publishing, New York

Rinde, S. 2005, "Tree structures, are we hard-wired?", *Thingamy Blog*, <http://thingamy.typepad.com/sigs_blog/2005/07/treestructures_.html>

Ricucci, A. 2008, *La Storia Siamo Noi: Belice ’68*, RAI Educational Television, Rome.

Romano, H. Ed. 2009 – 2010, *Unless You Will,* <http://www.unlessyouwill.com>

Saffer, D. 2009, *Designing Gestural Interfaces*, O’Reilly Media Inc, Sebastopol

Sagmeister, S. 2008, *Things I Have Learnt In My Life So Far*, Abrams Books, New York

Sebald, W.G. 1999, *Vertigo,* New Directions, New York

Sebald, W.G. 2001, *Austerlitz*. Penguin, London.

Shore, S. 2005, *American Surfaces,* Phaidon Press, London

Shore, S. 2005, *Uncommon Places*, Aperture, NY

Scianna, F. 2002, *Quelli Di Bagheria*, Peliti Associati, Italy

Sobchack, V. 1991, *The Address of the Eye; a Phonomenology of Film Experience*, Princeton University Press, Princeton

Soth, A. 2010, *From Here to There: Alec Soth's America*, (Engberg Ed), Walker Art Center Minneapolis

Spoerri, D. 1995, *An Anecdoted Topography of Chance*. London: Atlas Press.

Sofia, Z. 2000, "Container Technologies" *Hypatia Vol 15 No 2*

Sontag, S. 2001, *On Photography,* Picador, New York

Stewart, K. 2007, *Ordinary Affects*, Duke University Press, Durham

Sure, M. and F. 2001, *Supporting Users Creativity*. London. Asean Academic Press.

Tagg, J. 2009, *The Disciplinary Frame: Photographic Truths and the Capture of Meaning,* University of Minnesota Press, Minneapolis

Tanizaki, Junichiro. 2001 *In Praise of Shadows,* Vintage Books, London

Toledano, P. 2006 – ongoing, *Days With My Father,* website, <http://www.dayswithmyfather.com>

Tomasula, S. and Farrel, S. 2002, *VAS: An Opera in Flatland,* Chicago University Press, Chicago

Troika collective. 2008, *Digital by Design,* Thames and Hudson, London

Turkle, S. Ed. 2007, *Evocative Objects: Things We Think With,* MIT Press, Cambridge

Turkle, S. Ed. 2008, *Inner History of Devices,* MIT Press, Cambridge

Turkle, S. 2009, *Simulation and Its Discontents,* MIT Press, Cambridge

Wurman, R, S. 1989, *Information Anxiety,* Doubleday, New York.

Wurman, R, S. 2001, *Information Anxiety 2*, Cue, Indianapolis.
