Semester: LiD 4

Title:

An Exploration of Interactive Light Art, Architecture and Emotional Response

Project Period:
June - September 2018

Semester Theme:
Master Thesis - 30 ECTS

Supervisor(s):
Georgios Triantafyllidis
George Palamas

Project Group no:
N/A

Members:
Simon Davis

Copies:
1

Pg.s:
75

Finished:
20/09/18

Aalborg University Copenhagen

Frederikskaj 12,

DK-2450 Copenhagen SV

Semester Coordinator:
Georgios Triantafyllidis

Secretary: Lisbeth Nykjær

Abstract:

This thesis explores how a building's facade can be transformed into an interactive digital canvas.

Facades have adapted to change by modifying their uses according to the demands of architects, business, advertising and the public, as their needs change over time. I have shown how technology can be used to provide new experiences which capture the attention of observers.

This has lead to the formulation of the following research question: How may we provoke different emotional responses in observers through the use of Interactive Light Art?

Through the use of Flocking Boids interacting with a facade I will demonstrate how changes to the behaviour of the Boids can alter how installations are perceived emotionally, and also determine their effectiveness as a tool for Placemaking.

Acknowledgements

Firstly I would like to thank my supervisors George Palamas and Georgios Triantafyllidis for their guidance and support over the course of this project. This thesis has taken many twists and turns and would not have been completed without their help.

I would also like to thank my friend Vlad Limbean for his patience while helping me get to grips with some of the unfamiliar coding elements of this project.

Finally I would like to thank my parents for their love and support over the course of this MSc.

Contents

List of Figures

An Exploration of Interactive Light Art, Architecture and Emotional Response

1 Introduction

1.1 A Brief History of Media Facades

The facade of a building has long been considered to be one of the most important elements of an architectural plan as it is usually the first thing a visitor will see on approaching it. Communication has been an important element of facades, from inscriptions carved into stone in Roman and Greek times, to the inclusion of primitive sundials in old architecture and clocks on modern buildings, to the use of imagery to tell a story, especially in religious architecture: The basilica of the Sagrada Familia in Barcelona has three monumental facades, each one representing one of the three crucial events of Christ's existence: his birth; his Passion, Death and Resurrection; and his present and future Glory.

Fig. 1 Nativity Scene, Sagrada Familia
http://www.sagradafamilia.org/en/structure-and-form/

In the UK the industrial revolution saw a mass movement of people into towns and cities, which saw a corresponding growth in products, and small businesses to sell them. In many cases the use of their buildings' facade was the logical solution to promote the business.

Fig. 2 A Shop of Miscellany, Black Country Living Museum, UK

The mid 19th century USA saw a huge growth in advertising in the form of bill posting, where facades and other surfaces were covered in posters. Eventually this was regulated by issuing legal rights to certain surfaces and rented out to advertisers, which continues to this day. These surfaces could be stand-alone, or attached to a building.

The incandescent light bulb was introduced in the 1870s in the USA and was soon applied to advertising. Huhtamo (2010) describes how the illuminated advertisements and shop windows on Broadway in New York during this era “turned the street into a luminous attraction after dark.” In a further development, animations could be created by switching off sections of the bulbs rhythmically, vastly increasing the dynamism of the facade and cementing the area’s reputation for being a fun and exciting place to see, and be seen in, drawing more people into the district.

Fig. 3 Broadway, early 20th century

1.2 Genres of Urban Media

At this point it is helpful to understand the difference between light architecture, media architecture and media facades. Light architecture is the artificial light (or daylight) which illuminates a building. The use of different colours and brightness creates contrast, which in turn creates atmosphere, as I will discuss in chapter 2. This can also be the case with media architecture, although the main difference is that media architecture has an important dynamic aspect: any form of spatial movement of dynamic graphics or text distinguishes it from light architecture. If we define media as “communication in the visual form of a dynamic text, graphic or image, then the inclusion of this type of media onto one exterior side of a building constitutes a communicative element”. Therefore, a media facade is a facade with communicative elements embedded into it, while media architecture “describes the cultural, social and economic implications of these facades for the immediate environment.” Haeusler et.al. (2009)

So far I have discussed media architecture in terms of advertising, and basic communication, but this is only one of several uses. Fritsch et al. (2008) have summarised the genres of urban media in the following diagram:

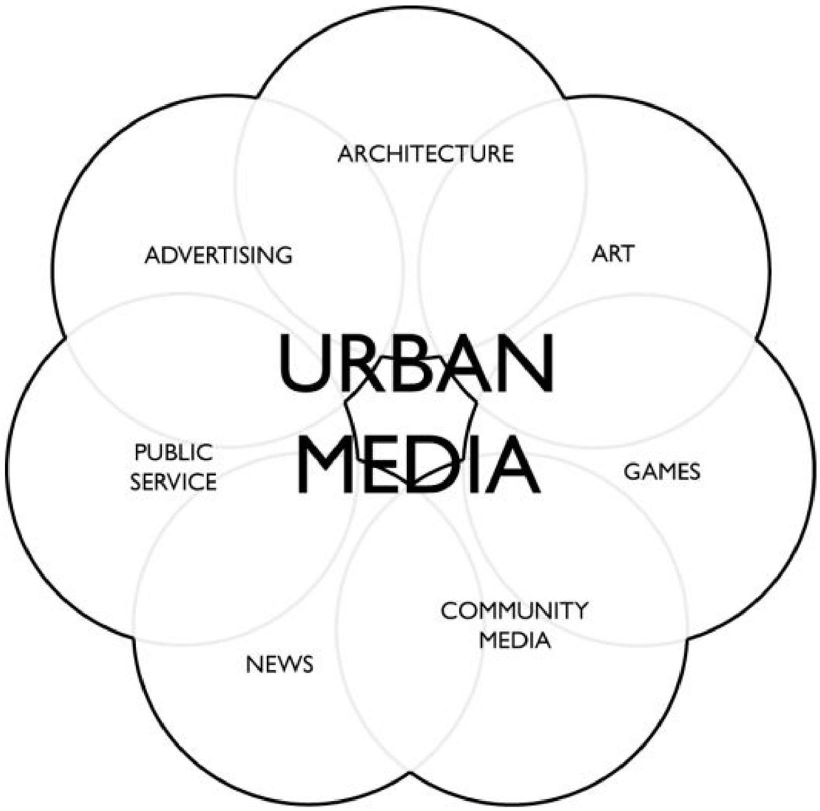

Fig. 4 Genres of Urban Media, Fritsch, et. al., 2008

Whilst there are plenty of examples of all of these genres, it is apparent that the majority of urban media is dedicated to advertising, followed by public service information. The photo below of Copenhagen's Rådhuspladsen shows two facades plastered with illuminated product placement, framing a large illuminated thermometer which spans the height of the building.

However, it can be argued that now in the 21st century our public spaces have been saturated with advertising, which we must not forget is the capture of public space by private interests. Whilst we will never get the advertising genie back in the bottle, it is worth exploring how we can use urban media in a different way, to return a sense of ownership of shared space to the public. In the case of this thesis I will be focusing on the combination of art, architecture and technology to create an installation which will temporarily reclaim a private space for the benefit of the public.

Fig. 5 Rådhuspladsen, Copenhagen

1.3 Mediatecture

According to Singer (2010), the word ´Mediatecture´ was coined in 1993 by the group ag4 as a portmanteau of media and architecture, and is described by Huhtamo (2010) as "the hybridization of architecture, urban design, media and new technology".

Brynskov et al. (2013) expand this definition to state that "the concept that covers the design of physical spaces at architectural scale incorporating materials with dynamic properties that allow for dynamic, reactive or interactive behavior....Light is the most frequently used modality, so light-emitting materials and building elements are central to the majority of buildings and spaces that would be characterized as media architecture."

Raheja (2007) starts from the position that if architecture is defined as the art of space, then media is an art of time. "This realization challenges the idea of separation between media and architecture and prompts the conception of a new "media architecture" as a space-time construct in which media and architecture come together.

Stojsic (2017) notes that "it was Robert Venturi who introduced the paradigm of information into architecture. Advising us to learn from vernacular and commercial culture, Venturi saw electronic display not as an optional addition secondary to physical form, but as a building brick of architecture for the information age we live in. … The only difference is that information conveyed through architecture is no longer a monologue set in stone or brick, but an ever-changing digital image that offers the possibility of dialogue through the possibility of real-time interaction.... Instead of conventional space with the addition of media surface, we operate with architecture that holds a potential to become media infrastructure, integrating physical and information space."

Although a relatively new discipline, it can be seen that there are as many interpretations of the word as there are mediatects. However, the common denominator is that it involves the combination of buildings or structures with technology to produce visual content that a viewer may interact with in their own way, or not at all. In other words, architecture itself has become a new media interface, which according to Singer (2010) , is actually addressing " a very old architectural subject using contemporary methods...mediatecture does not only use media but is also a medium itself - a mediator between worlds of built and physical realities on the one hand and of imagined identities and visions on the other."

1.4 Placemaking

The Project for Public Spaces use the following definition: "Placemaking refers to a collaborative process by which we can shape our public realm in order to maximize shared value. More than just promoting better urban design, Placemaking facilitates creative patterns of use, paying particular attention to the physical, cultural, and social identities that define a place and support its ongoing evolution."

The key to successful placemaking lies in the involvement of the community and should be viewed holistically. Many factors are involved in the creation of a space which works for people, revolving around sociability, uses and activities, access and linkages and comfort and image, which ideally result in a place that improves people's health, well-being and quality of life.

Clearly, one light installation on its own can only be considered to be a small element of the overall plan for community involvement, but as technology has become more accessible, digital urban installations have become more common worldwide. In the Media Architecture Compendium, Hespanhol (2017) states that "they are digital solutions that facilitate direct human interactions in shared urban spaces, encouraging social digital encounters and allowing citizens political or creative expression.

The benefit of my installation will be to help create the space as a night-time destination, it is dynamic, inclusive, collaborative and will provide opportunities for socializing among people of all ages and backgrounds. Gehring (2014) discusses the importance of "shared encounters", or making a connection between the environment and its inhabitants when creating interactive installations in urban spaces. The key is to create engagement by bringing together architecture, urbanism, social sciences, anthropology and computer science.

1.5 Architectural Facades and Projection Mapping

While the method of projecting images onto a flat surface dates back to the 1840s (Jones 2012), the art of projection onto 3D surfaces is much more recent. The first public display of 3D projection mapping can be traced to the Haunted Mansion ride at Disneyland in 1969: this involved 5 busts (the “Grim Grinning Ghosts”) singing the ride’s theme song and was created by filming the singers then projecting their images onto the busts.

Fig. 6 “Grim Grinning Ghosts”, Disneyland

However it was not until around 2005 that the the concept of projection mapping as an urban installation took off, as this was when “the first video projectors with a high light output of 10.000 Ansi-Lumen became more affordable”. (Schielke, 2013)

Arcara (2016) defines the process of projection mapping as “drafting in a digitally created virtual reality, then projecting the digital copy back into real space, thereby creating a new reality.”

As the software used for mapping, such as MadMapper or TouchDesigner, has become more accessible and the projector technology involved has decreased in price, the growth of projection mapped light installations has become widespread worldwide due to their effectiveness in communicating a message and their popularity with the general public. This is in part because the technology is still so new that there is a novelty value for many people who are unaware of the medium’s capabilities and thus still have the capacity to be amazed by animations created by a skilled programmer.

In her thesis, “Mapping the Mundane”, Arcara (2016) explores how projection mapping can be used “to create a spectacle that influences how spectators perceive existing space.” For her, the importance of projection mapping is in its “potential to transform aspects of physical space” with the potential to adapt the form to a variety of indoor or outdoor environments, responding to the different needs of the the environment or the viewer.

Not only can we use projection mapping to create a responsive, dynamic environment in architecture, but the medium can be used as a narrative form over time. An example of using projection mapping to tell a story is the Guggenheim museum in Bilbao. From October 11-14th 2017, a 20-minute-long multisensory production, “Reflections”, combined music, light and projection to create a show on the building’s north-facing titanium facades that told the story of the museum’s genesis and design. (Lynch, 2017) The image below shows blooming flowers in reference to Jeff Koon’s iconic “Puppy”, situated outside the museum.

Fig. 7 Guggenheim Museum, Bilbao, mapped by 59 Productions, 2017

2 Phenomenology

2.1 Definition

“Phenomenology is the study of essences; and according to it, all problems amount to finding definitions of essences...But Phenomenology is also a philosophy that puts essences back into existence” according to the Merleau-Ponty’s Phenomenology of Perception (1962). “It tries to give a direct description of our experience as it is” offering “an account of space, time and the world as we ‘live’ them”, or in other words, the ‘Phenomenology of Perception’. Merleau Ponty was a philosopher who never wrote about light or architecture, so at first glance it may not appear relevant to this thesis.

However, Hale (2013) argues in Merleau-Ponty for Architects that “Merleau-Ponty should, in fact, be seen as a ‘proto- posthumanist’ thinker: someone who believes in a fluid definition of the individual self, both dependent on and inseparable from its surroundings.” Hale explains this by stating that “sense is actually inherent in bodily experience, and how the body acts as the vital pivot between the inner world of the individual and the outer world of social and cultural forces.”

It is this connection to the senses which is relevant to this project. By creating a unique bodily experience for my observers I would like to communicate the essence, or intrinsic nature of the the design, which will determine the character of the experience. Nevertheless, the construction of the display is only part of the experience. We must also consider how the wider context in which the design is shown also influences the experience from the perspective of the viewer.

2.2 Experience and Atmosphere

Hale (2013) states that Merleau-Ponty described how “our initial understanding of a space is based on its practical possibilities – we grasp it as a structured arena for action, inviting us to use it in a particular way. This idea of experience as an ongoing interplay between perception and action has vital implications for how we think about space in architecture today, and more importantly, for how we set about designing places that people find engaging, stimulating and meaningful.”

The key point here is how our perceptions of a space affect our actions and how in turn this forms our overall experience. We are at the point now where technology has advanced to the extent that we can now take Merleau-Ponty literally and enable people to take control of creating their own stimulating experiences through the creation of a stage, or canvas for them to use. By this definition there is nothing more engaging than literally interacting with an architectural feature.

Furthermore, if we consider our facade to be the stage for our ‘performance’, then we must consider the work of Gernot Böhme. In “The art of the stage set as a paradigm for an aesthetics of atmospheres" (2013) he starts by identifying the paradox of trying to create something intangible, and how "atmospheres can be divided into moods, phenomena of synaesthesia, suggestions for motions, communicative and social-conventional atmospheres...the character of an atmosphere is the way in which it communicates a feeling to us as participating subjects". Therefore a person's mood can be altered, for better or for worse, by changing the atmosphere according to the intention of the designer. However, atmosphere is not a thing - one can only create an atmosphere by creating the conditions for an atmosphere to appear, hence his use of theatrical sets as humanity's oldest and most typical example of atmospheric creation. However, he goes on to say, "In general, it can be said that atmospheres are involved wherever something is being staged, wherever design is a factor - and that now means: almost everywhere."

Böhme then goes on to reference Plato's theory of mimesis, specifically 'phantastike techne', which in contrast to 'eikastike techne', does not have to "strictly imitate a model".
He states that "the artist does not see his actual goal in the production of an object or work of art, but in the imaginative idea the observer receives through the object. The manipulation of objects serves only to establish conditions in which these phenomena can emerge. But that is not achieved without the active contribution of the subject, the onlooker." This is particularly interesting to me as I would like my design to spark ideas and build connections with my audience, whereas the design itself is figurative, but solely representative.

Arcara (2016) defines phenomenology as "the analysis of appearances studied through human experience". She then goes on to relate phenomenology to projection mapping by stating that "the relationship between concept and form is based on the mind's interpretation of what the brain perceives. Architecture that emphasizes a phenomenological paradigm is often filled with stimulant elements immersing the occupant in the entirety of the architecture."

Projection mapping can be effective in evoking these sensory stimuli. Arcara applies Mirror Image theory developed by Jacques Lacan, that infants can identify themselves in a mirror despite having underdeveloped vision until the age of 24 months. It follows that "as we age our sense of motion and spatial awareness is inherent within our subconscious, it is logical to believe that we are drawn to high contrast imagery with strong geometric lines because of our underdeveloped sense of sight at a young age. Even as we age and develop, we are still drawn to works rudimentary in form with a strong sense of geometry and a monochromatic palette, such as the architectural works of Le Corbusier and the art of Sol LeWitt."

The relevance of Arcara's theory to this thesis is that I have chosen a building with distinctive geometric forms incorporated into the facade. In order to create the most effective stimuli it should be ensured that the design is incorporated holistically into the facade, using high contrast monochromatic elements for maximum effect.

3 State of the Art

3.1 Interactive Light Art

In the past, public art had tended to made with durability and longevity in mind for resistance against the elements: static and designed for eternity, in the words of Lieser (2010). However, the widespread adoption of LEDs and the growth of computer processing power from the 1990s to the present day has made digital public art a viable medium for many artists. Lieser has identified a paradigm shift in this field as the lines between art, craft and design are blurred. He notes that since Andy Warhol's Factory in the 1960s, contemporary art has moved away from convention that artists should produce their own original art with their own personal signature and that artists such as Jeff Koons and Damien Hirst run their studios like a production company - they provide the creative vision while their employees create the art. “The absence of a personal signature, which in the beginning was declared the flaw of digital art, is now regarded as an especially honest form of artistic creation.”

Werner (2007) observes that the emergence of digital interactivity in the late 20th century has allowed both consumer society and the critical art world to investigate (and manipulate) the possibilities of creativity through it, using technological advances “to perceive the world and interact with others from new perspectives.”

A further difference from traditional art in that digital art is not considered permanent, but is treated more like a performance. The ability of digital art to respond to or interact with the surrounding environment and its observers demonstrates an agility which often mirrors our own transient lifestyles, which implies that the medium is an apt reflection of the times we live in.

Werner (2007) proceeds to explain how modern art investigates notions of alternative perceptual understandings, and concludes that “an underlying inclination of the artwork in this study is the representation of space.” The rest of this chapter discusses modern digital interactive artworks, showing how their audiences “become the main focus of the work by either being immersed into a space or by becoming interactive with the piece. Rather than being viewers of the art, they become participants in it; thus, they are forced to interact with the work in order for it to be accessible.”

3.2 Examples of Interactive Light Art

The following examples show the range and capabilities of interactive light art and how the installations can create shared encounters in urban environments:

3.2.1 Spider, by Kollision

This installation is particularly relevant to this thesis as it successfully demonstrates how flocking boids can be used on a 2D surface to create emotion in the observer, in this case fear. It is also a participatory experience for the viewer according to Pihlajaniemi's definition of the stages of interactivity (2017), which will be explored in greater detail in the following chapter. It is best described using Kollision's own explanation of the installation as found on their website:

"In 2010, Kollision was commissioned to create an interactive floor installation in Faarup Sommerland. The installation was to be a part of a new attraction, The Fox Hole, based on the idea that after closing time the animals of the surrounding forest invades the theme park. The Fox Hole is staged as an underground world where the visitor meets a variety of fun, challenging and scary obstacles while in search of the fox.

The interactive floor created by Kollision is situated in a dark narrow alley in the Fox Hole. Using projectors we created the illusion of an old, wooden floor projected onto a concrete corridor. When a visitor approaches, spiders emerge from the cracks in between the floorboards, and when the visitor steps into the corridor, hundreds of small and large spiders suddenly appear from underneath the walls and start chasing the visitors brave enough to cross the floor.

The installation is running on a tracking library, developed by Kollision, that analyses a video stream from an IR-camera mounted in the ceiling along with an IR-emitter. In this way a computer placed with a projector above the ceiling calculates the positions of people on the floor and spawns spiders according to the activity on the floor. The spiders behave according to basic flocking behaviour similar to schools of fish and flocks of birds, creating a scary and realistic swarm of spiders."

Fig. 8 Spider, by Kollision, 2010
https://kollision.dk/en/spider

3.2.2 Carsten Nicolai . α (alpha) pulse (3.2.2)

This installation, in contrast, turns an entire Hong Kong skyscraper into a media facade. His concept is to reimagine the building as a lighthouse, sending pulses of light and sound in a synchronised frequency throughout the city, a work of art which involved millions of people (whether they wanted to or not). A further level of interaction was included in the installation via the use of an app which provided the audio link to the user and also allowed the user to respond to the light display.

This work has been included because it shows how powerful even a simple idea can be, and that it also shows a different use of technology in the form of phone apps, and how they can be used to connect people to the installation, providing a communal experience on a huge scale.

Fig. 9 a (Alpha) Pulse, Carsten Nicolai, 2014

https://vimeo.com/105547177

3.2.3 Behaviour Morphe by Zaha Hadid Architects

This project is of interest because firstly, it uses generative biomimetic forms to create ever-changing organic shapes which contrast with the artificiality of the building. Secondly, this is an interesting example of another potential use of interaction: while there is no scope for human observers to interact with the work, the digital life forms are programmed to interact with each other. A concept such as this could potentially be applied to my own project, and would certainly be effective in creating a memorable experience for the public.

In the article , Marco Rinaldi states that the installation “showcases the latest digital spatial simulation tools the practice applies in the comprehensive analysis and planning of its architecture, and looks further ahead to concepts of spatial design of the future. The projection on the castle’s façade reveals its interiors as digital laboratories of human behavioural simulation, demonstrating the circulation and congregation of digital actors programmed with artificial intelligence that interact with the spaces of the castle and each other.”

Fig. 10 Behaviour Morphe by Zaha Hadid Architects
https://www.archdaily.com/876660/zaha-hadid-architects-to-project-augmented-reality-light-show-onto-karlsruhe-castle-at-2017-schlosslichtspiele-festival

3.2.4 WAKE UP by Iregular

This is an interesting installation as it combines old technology (fluorescent tubes) with modern proximity sensors. “WAKE UP is a reactive light sculpture made of vertically organized fluorescent bulbs that react to the presence of a person. As the viewer approaches the sculpture random light patterns are produced. As he gets closer the patterns get more intense to finally deliver the message. The main interest of this work is to generate a reflection on how much the dream of a society without discrimination has come true, and how much must one still need to act on it to achieve it.” This installation demonstrates the importance of having a human-centric approach, having been presented at MLK50, an exhibition commemorating the 50 years of Martin Luther King's speech "I have a dream".

Fig. 11 WAKE UP by Iregular, 2014
https://vimeo.com/63244130

3.2.5 'P-Cube' by Marcos Zotes

Whilst this installation is not interactive, it has been included because it is a good example of how generative art combined with projection mapping can work together to create a space for people to congregate around and inside. In other words, it is a fantastic example of placemaking in action.

“P-Cube uses the industrial everyman material of scaffolding and combines it with semi-transparent fabric to create a temporary installation at Moscow's Polytechnic Museum. By day the structure has a spectral presence and at night geometric projections make it pulse gently with digital life. The public can walk inside the near 30-feet by 30-feet temporary pavilion, walking up a stairway to a viewing platform where they can look over the site and are immersed in the grid-like morphological projections mapped onto its skin, along with the interplay of color and shadows.”

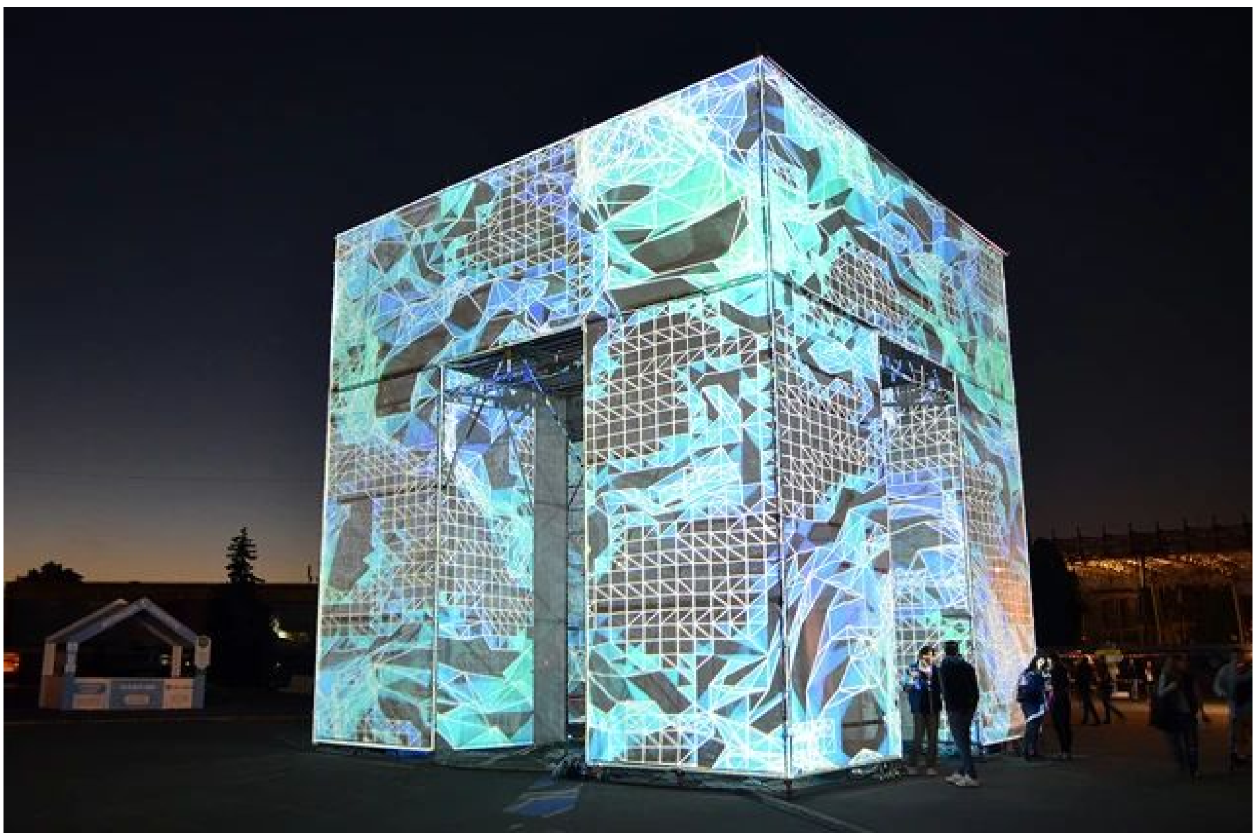

Fig. 12 P-Cube by Marcos Zotes, 2015
https://creators.vice.com/en_us/article/qkwx93/projection-mapped-cube-pulses-with-digital-life

3.2.6 An Interactive Projection Mapped Graffiti Wall by Michael Marner and Lachlan Tetlow-Stewart

This project differs from the previous projection mapping techniques discussed earlier as the content is produced for a specific projector viewpoint and mapped in 2D. The objective of the installation was to obtain the phone numbers of the audience of a show, which they did by encouraging them to send SMS messages to the organisers, which would then be projected onto a facade as an animated graffiti tag.

Fig. 13 An Interactive Projection Mapped Graffiti Wall by Michael Marner and Lachlan Tetlow-Stewart, 2016
https://www.20papercups.net/software-projects/an-interactive-projection-mapped-graffiti-wall/

3.2.7 Velcom HQ Opening Show: Projection Mapping and Interactive Tablet On-Building Painting Combined, by Gluk Media, 2012

As part of a show arranged for Velcom, Gluk Media created a tablet-based interactive installation. Audience members had the unique experience of playing a large building like a giant musical instrument. Using the touch screen tablets, they lit up the building’s windows, generated melodies, and made bright graphics appear and move on a grand scale. The opening gala gained attention from both the media as well as ordinary participants and can be considered a good example of cooperation between media, business and the public.

Fig. 14 Velcom HQ Opening Show: Projection Mapping by Gluk Media, 2012

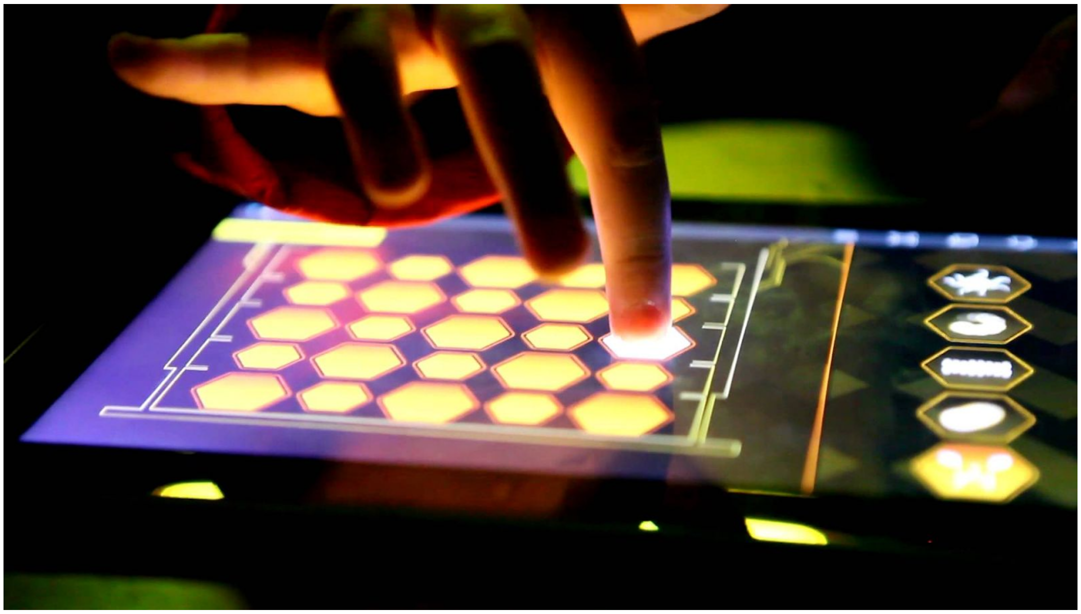

Fig. 15 Velcom HQ Opening Show: Interactive Tablet used to control On-Building Painting
http://glukmedia.com/portfolio/velcom/

3.2.8 B/SIDE by João Martinho Moura, 2012

This installation shows combines 2 elements: the background shows Flocking Boids connected to, and interacting with each other. We can also see a Kinect type camera capturing the movements of a member of the audience. Their arm is raised and as a consequence the boids have identified it as a predator and swarmed away from it, much like a school of fish. Of all the examples I have shown, this one captures the immense potential of using Flocking Boids to create a public light art experience. Combined with music, Moura has created a true multisensory experience for the people of Braga, Portugal.

Fig. 16 B/SIDE by João Martinho Moura, 2012
https://vimeo.com/51272346

4 Levels of Interactivity

In her Doctoral dissertation “Designing and Experiencing Adaptive Lighting”, Pihlajaniemi (2017) defines different stages of interactivity with regard to how “lighting adapts to information about the environment and its users or to other information relevant to intended lighting behaviour”.

Starting with dynamic lighting, which simply changes, we move through Adaptive lighting explained as “dynamic lighting that adapts to some form of reference data”, Intelligent lighting which “is a form of adaptive lighting that adapts in real-time to data”, Interactive lighting “when there is explicit user interaction as part of the adaptation process” and Participatory lighting, where adaptive and interactive lighting allows users “[to create] something artistic or meaningful”.

These inter-relations are summarised in the following model:

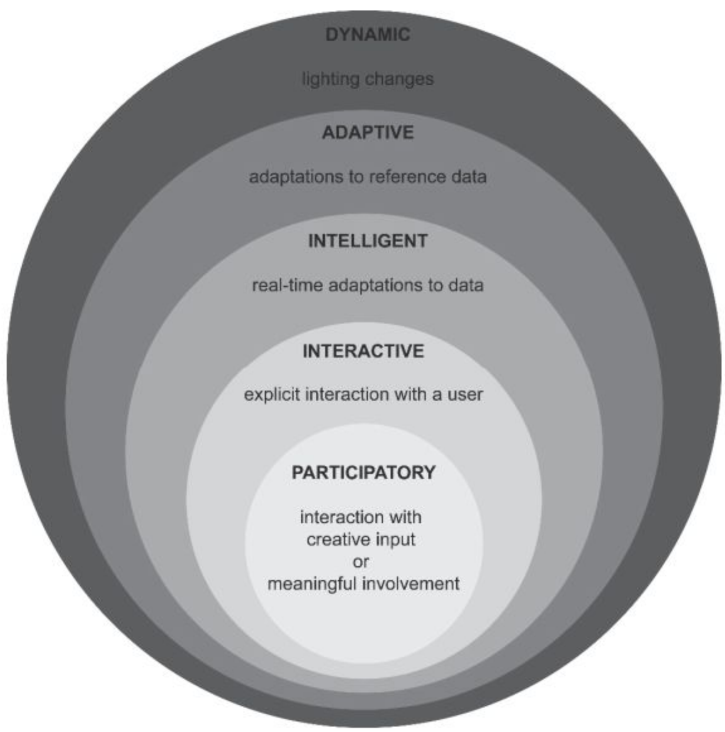

Fig.17 Concepts and Definitions of Adaptive Lighting, Pihlajaniemi 2016

She also states “adaptive lighting is approached as an element of architecture and urban space, which has an influence on the human environmental experience at various levels.” The nature of this project means that my objective is to create an experience that should at a minimum be classed as Intelligent lighting, but ideally should provide a Participatory experience for the audience in order to achieve a meaningful response.

Fritsch et. al. (2008) also explore the idea of engaging interaction, with the aim of “exploring and experimenting with the ways in which situated interactive systems and installations may engage users in participation and inquiry” which is the idea that the value of our theories of the world must be evaluated by their consequences in practice and the extent to which these ideas help us experience and understand public spaces. They considered that using technology to change spatial experiences can be both the experience and a means of “altering future experience.”

5 The Transdisciplinary Approach

I have followed the AAU Transdisciplinary approach to project work, "where technical, creative and humanistic methods and knowledge is transferred, translated and transformed in the creative (lighting) design process" . This thesis will show how architectural facades can be transformed with media technology to create different emotional experiences for the viewer. I will use the programming language Processing to create generative art, showing how biological forms, modelled in computer simulations, can demonstrate how basic artificial life forms can provoke different reactions in the audience via changes in atmosphere, thus incorporating a phenomenological approach to the problem.

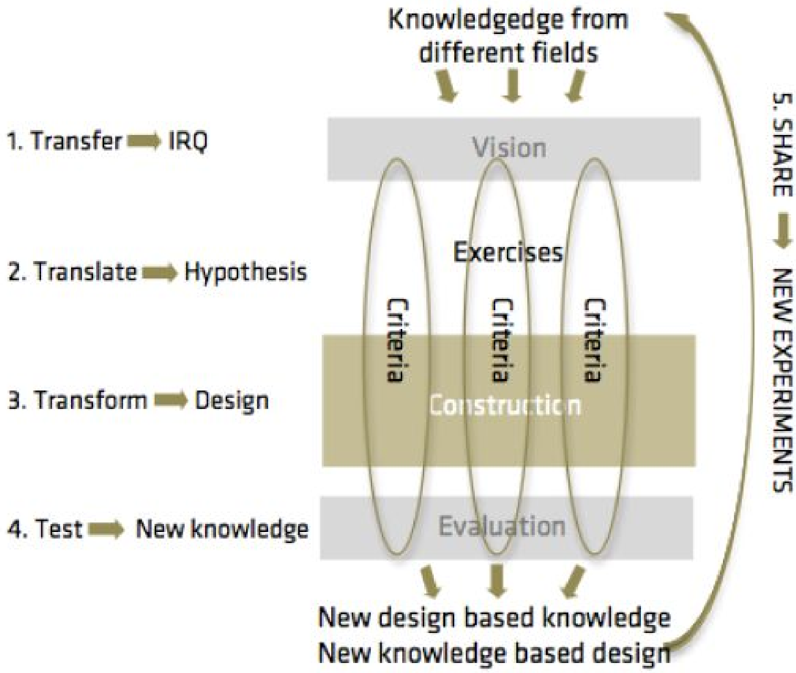

Fig.18 Process Model of the Architectural Experiment, combining knowledge in the creative design process (Hansen, E.K., 2014)

5.1 Problem Statement

Since the development of the incandescent light bulb, Architects, lighting designers and town planners have known that illuminating a place's most imposing, beautiful or unique public buildings, spaces and monuments draws attention to them. People may travel great distances to experience them, and illumination may provide an even more interesting experience at night. Images shared over the internet inspire others to visit, contributing to the prosperity of the place and confirming its reputation as a 'destination'. Even for the residents, seeing a beloved building tastefully illuminated gives a sense of pride in the accomplishments of previous members of the community and provides enjoyable shared visual experiences on a regular basis.

But what of those places without any inspiring buildings to use as a visual focal point for lighting? An ordinary building's facade may be visually pleasing in daylight, but the lack of accurate colour rendition of sodium street lighting may not bring out the best aesthetics of the facade at night. A lack of direct illumination would cause the facade to be completely disregarded at night. Certainly no-one would want to visit it, or the millions of other equally ordinary buildings around the world. We see therefore that the majority of facades are purely functional and contribute very little to the aesthetics of a place.

In addition to this, the use of media facades in towns and cities has grown hugely since the advent of LED screens, in most cases the transfer of information still travels in one direction, from the content provider to the observer via the facade. This can be seen as yet another example of public space being co-opted by vested interests with few stakeholders being involved with the consultation process. This does little to create a sense of involvement in the community when the message being communicated is either being lectured to by the public sector or being treated solely as a consumer by the private sector.

5.2 Vision

My vision therefore is to explore how a building's facade can be transformed into a interactive digital canvas. By temporarily reclaiming a space for public interaction I would like to use generative art to foster a sense of inclusiveness and community by showing how this change of use can have a beneficial effect on public wellbeing.

5.3 Hypothesis

In his essay "Media Architecture and the Role of Urban Media Art in Digital Placemaking" , Colangelo (Hespanhol et al. 2017) argues that public artworks which are large scale, networked, interactive, participatory and digital mark the convergence of media architecture and digital placemaking. The interactions and experiences generated can help to revitalise urban spaces and bring people together. Urban media art “can transform basic and often heavily commercialised urban screens into more nuanced and open urban media environments”.

The value added as a result of this is to foster a sense of community by enabling human connection. What we mean by this is that we want to provoke positive emotions and feelings. The Circumplex of Affectation model is used in Psychology to help record responses to stimuli: the horizontal axis measures valence, i.e. whether or not the feelings produced are pleasant or unpleasant, while the vertical axis measures arousal, or the extent of the stimulation.

Posner et. al. (2005) state that “clinicians and researchers have long noted the difficulty that people have in assessing, discerning, and describing their own emotions. This difficulty suggests that individuals do not experience, or recognize, emotions as isolated, discrete entities, but that they rather recognize emotions as ambiguous and overlapping experiences. Similar to the spectrum of color, emotions seem to lack the discrete borders that would clearly differentiate one emotion from another... Subjects rarely describe feeling a specific positive emotion without also claiming to feel other positive emotions.

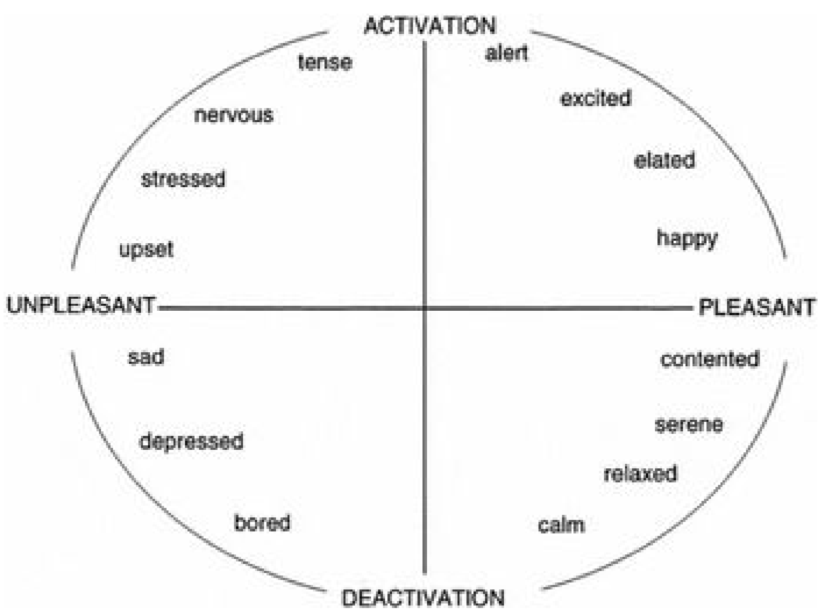

Fig. 19 Circumplex of affectation model, Posner et.al.

This would suggest, therefore, that it would be better to aim for creating general emotions with the installation as opposed to trying to create a specific feeling. Due to the subjective nature of human responses to art, I would almost certainly fail, plus the work may also trigger responses (both positive and negative) that I may not even have considered. Instead, I will focus on how the installation affects people emotionally; the cognitive aspect of the associations brought forth in their imaginations; how the installation is perceived by the viewer and any changes in behaviour that the installation may provoke.

Fritsch et al. (2008) discuss the “Affective experience” to describe the perceptive, emotional and cognitive elements of the possible urban interactions spurred by media façades. He quotes Massumi to describe Affect as “the level of experience where our non-conscious bodily experience meets our conscious cognitive experience of ourselves and the world.” The concept is useful as it can help us analyse how we perceive urban interactive design on a micro-level.

On a macro-level, “the concept of affect offers a way to understand how human practice in the world is determined by a person's ability to affect the world and a person's ability to be affected by the world (Clough ed. 2007, in Fritsch et. al. 2008). It connects how people think and feel to how their bodies act and move in the world. Therefore, we should always consider how “an installation might spur long-lasting effects on a cultural and social level by giving the individual user the means to change his, her or others’ conception of a given situation.” (Fritsch et.al. 2008)

5.4 Research Question

In the previous chapters it was discussed how facades have adapted to change by modifying their uses according to the demands of architects, business, advertisers and the public as their needs change over time. I have also shown how technology can be used with and incorporated into facades to provide new experiences which capture the eyes and attention of observers. This has lead me to formulate the following research question:

How may we provoke different emotional responses in observers by projecting interactive generative art onto a facade?

5.5 Concept

I have chosen to focus on creating a design using Flocking boids in Processing. Flocking voids simulate the flocking behaviour of birds, or schools of fish, and can be made more complex by the introduction of ‘predators’, or other variables to add more drama to the simulation. In addition, the boids can be programmed to interact with architectural features such as windows, by being attracted to or repelled by them as well as each other.

Furthermore, an online article written by Craig Reynolds (1995), the creator of Boids, discusses how “the interaction between simple behaviors of individuals produce complex yet organized group behavior. The component behaviors are inherently nonlinear, so mixing them gives the emergent group dynamics a chaotic aspect. At the same time, the negative feedback provided by the behavioral controllers tends to keep the group dynamics ordered. The result is *life-like* group behavior. A significant property of life-like behavior is *unpredictability* over moderate time scales. For example at one moment, the boids in the applet above might be flying primarily from left to right. It would be all but impossible to predict which direction they will be moving (say) five minutes later. At very short time scales the motion is quite predictable: one second from now a boid will be traveling in approximately the same direction. This property is unique to complex systems and contrasts with both *chaotic* behavior (which has neither short nor long term predictability) and ordered (*static* or *periodic*) behavior. This fits with Langton's 1990 observation that life-like phenomena exist poised at the *edge of chaos."*

It is this idea of 'life-like' behaviour which interests me and I believe that the juxtaposition of these temporary digital life forms with the permanence of old architectural structures provides a compelling contrast on which to base the installation. Werner (2007) references John Dewey (1934, pg10) who defines an aesthetic experience specifically as an emotional reaction to an event that happens in relation to nature. Therefore it is interesting to investigate how people would react to simulations of natural behaviour.

The possibilities for interaction with Boids are endless: during the research for this project I investigated the potential for interactions including having the Boids react to ambient city noise, specific sounds, data flows, interactions with people using buttons, sliders, keyboards and cameras to translate the movement of people into the movement of the Boids. Finally, due to time constraints and the need to focus on the aesthetics of the installation, it was determined to focus on the interactions of the Boids with each other and the boundaries as they would on a representation of a facade. This simplicity would also make it easier to evaluate the emotional responses of the observers.

6 Methodology

Having made the decision to project flocking boids onto a facade, my first task was to find a suitable simulator to adapt to my project. Using Google, I was able to search for examples of flocking boids and to use their source code as a basis to work with. From there I experimented with different parameters to create a range of potential visual effects which could be incorporated into the design. This was based on the work of Reynolds (1995), who identified three ´steering behaviours´ which describe how a boid maneuvers, depending on the position and movements of other boids in the shoal, flock or swarm: Separation, Alignment and Cohesion

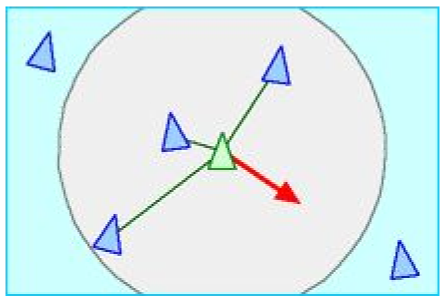

Fig. 20 Separation: steer to avoid crowding local flockmates

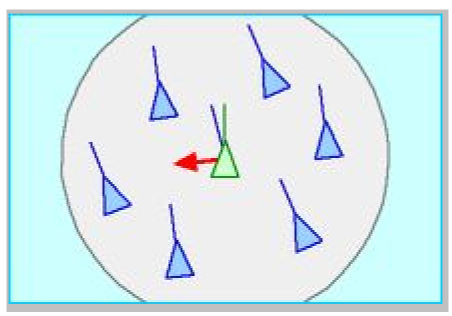

Fig. 21 Alignment: steer towards the average heading of local flockmates

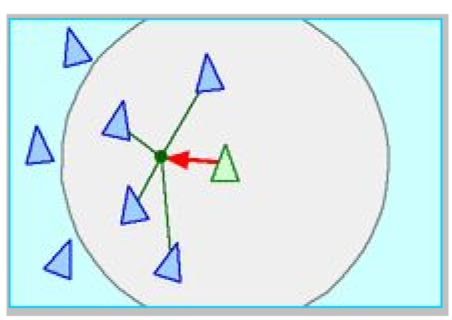

Fig. 22 Cohesion: steer to move toward the average position of local flockmates

Each boid only reacts to boids in its immediate vicinity, or neighbourhood, which is defined as the distance from the Boid and the angle relative to the Boid’s direction of movement. Any boids outside the neighbourhood are ignored.

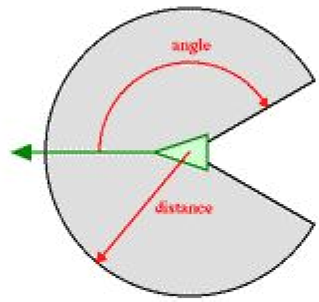

Fig. 23 A Boid´s Neighbourhood

To avoid reinventing the wheel, the design was based upon a piece of digital art named “boids with boundaries”, written by aadebdeb, and found on the website www.openprocessing.org. (7.1) This had the advantage of providing the essential elements of my design: circular boundaries for the boids to flock around, and variables of both the boids and the boundaries which could be adjusted to create different effects.

Furthermore, the author of the original code had included other variables which I could then alter to create different effects. In addition to manipulating the strength and range of the Separation, Alignment and Cohesion variables, I was also able to experiment with the strength and range of the boundaries, and the size, speed and total number of the boids. I also changed the appearance of the Boids by removing the stroke around each Boid and adding a white fill, so that they would appear brighter against a dark background.

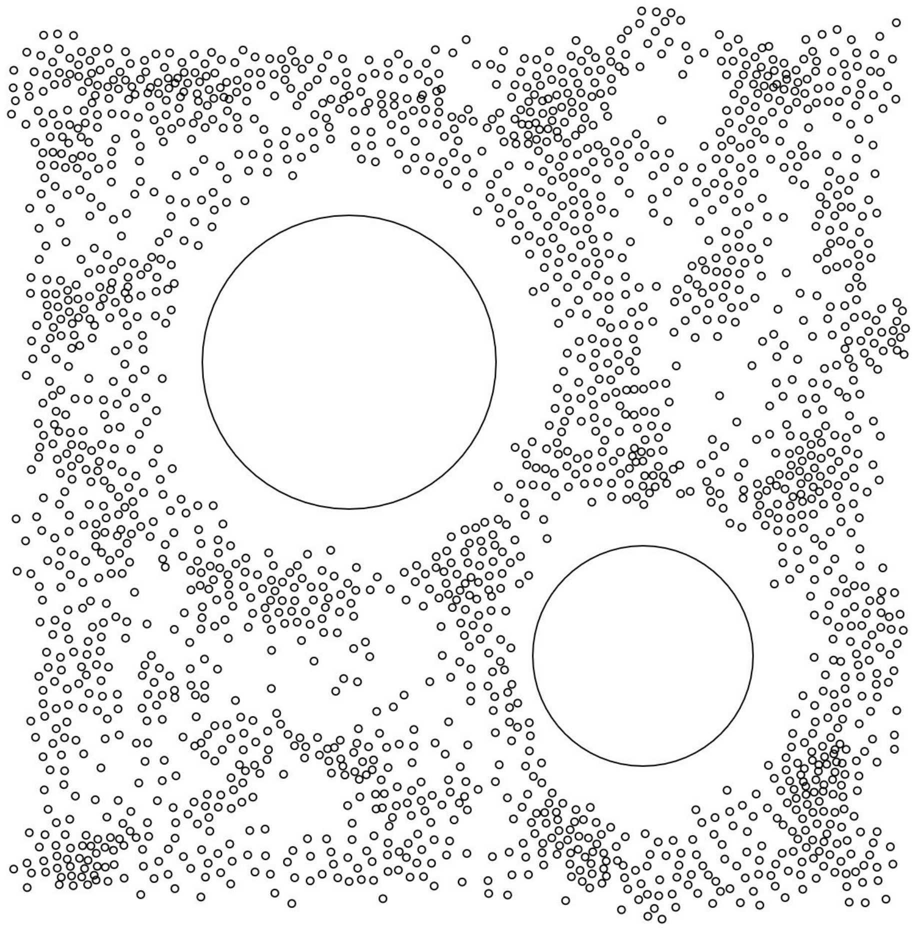

Fig. 24 “boids with boundaries”, by aadebdeb

Once I had done this, I was able to identify which variables created the most interesting effects, causing different emotional responses and reactions in the viewer. The final four versions can be loosely classified according to the size and speed of the Boids, namely Small/Slow, Small/Fast, Large/Slow and Large/Fast. A control was also included which featured small Boids which appeared to be stationary.

The installations were tested in the Light Lab at AAU using an Epson EMP-74L to project the images onto a white wall, an ASUS Zenbook Pro to run the code and a Sony Vaio to record the student volunteers’ responses via a questionnaire created in Google Forms.

The questionnaire contains eleven questions on a 5 point Likert scale for ease of analysis, divided into two areas of interest: The first six questions concern the observers’ emotional responses, both positive and negative, to each of the installations, whereas the following five questions seek to quantify the volunteer’s responses to the effectiveness of the placemaking aspect of the installations. Each set of questions was repeated for all five of the installations shown, and a space for comments was included at the end of the questionnaire.

Boids Evaluation

Observe the Boids: How do you feel after watching them?

All items are on a 5-point Likert scale ranging from zero to five
(1 = Not at all, 2 = Very Little, 3 = Sometimes, 4 = Often, 5 = A lot)

1. **I feel stimulated**
   Mark only one oval.

	1	2	3	4	5	
Not at all	◯	◯	◯	◯	◯	A lot

2. **I feel relaxed**
   Mark only one oval.

	1	2	3	4	5	
Not at all	◯	◯	◯	◯	◯	A lot

3. **I feel contented**
   Mark only one oval.

	1	2	3	4	5	
Not at all	◯	◯	◯	◯	◯	A lot

4. **I feel stressed**
   Mark only one oval.

	1	2	3	4	5	
Not at all	◯	◯	◯	◯	◯	A lot

5. **I feel bored**
   Mark only one oval.

	1	2	3	4	5	
Not at all	◯	◯	◯	◯	◯	A lot

6. **I feel uncomfortable**
   Mark only one oval.

	1	2	3	4	5	
Not at all	◯	◯	◯	◯	◯	A lot

7. **Does the installation enhance the facade?**
Mark only one oval.

	1	2	3	4	5	
Not at all	◯	◯	◯	◯	◯	A lot

8. **Does the movement of the Boids look organic?**
Mark only one oval.

	1	2	3	4	5	
Not at all	◯	◯	◯	◯	◯	A lot

9. **Does the movement of the Boids improve the experience?**
Mark only one oval.

	1	2	3	4	5	
Not at all	◯	◯	◯	◯	◯	A lot

10. **Does the speed of the Boids improve the experience?**
Mark only one oval.

	1	2	3	4	5	
Not at all	◯	◯	◯	◯	◯	A lot

11. **Does the size of the Boids improve the experience?**
Mark only one oval.

	1	2	3	4	5	
Not at all	◯	◯	◯	◯	◯	A lot

12. **Any other comments?**

Fig. 25 Questionnaire for testing phase

7 Designs

Using SketchUp, I first created a simple facade with 3 circular windows over 3 stories. I used both large and small sized windows to provide variety in the movements and interactions of the Boids. I added a grey brick effect to the facade, and glass in the windows which showed a reflection of a night sky.

I then changed the dimensions of the canvas to match that of the facade, and added seven extra circular boundaries to the code. After that I positioned the circles over the windows so that the installations would show the Boids interacting with and around the windows

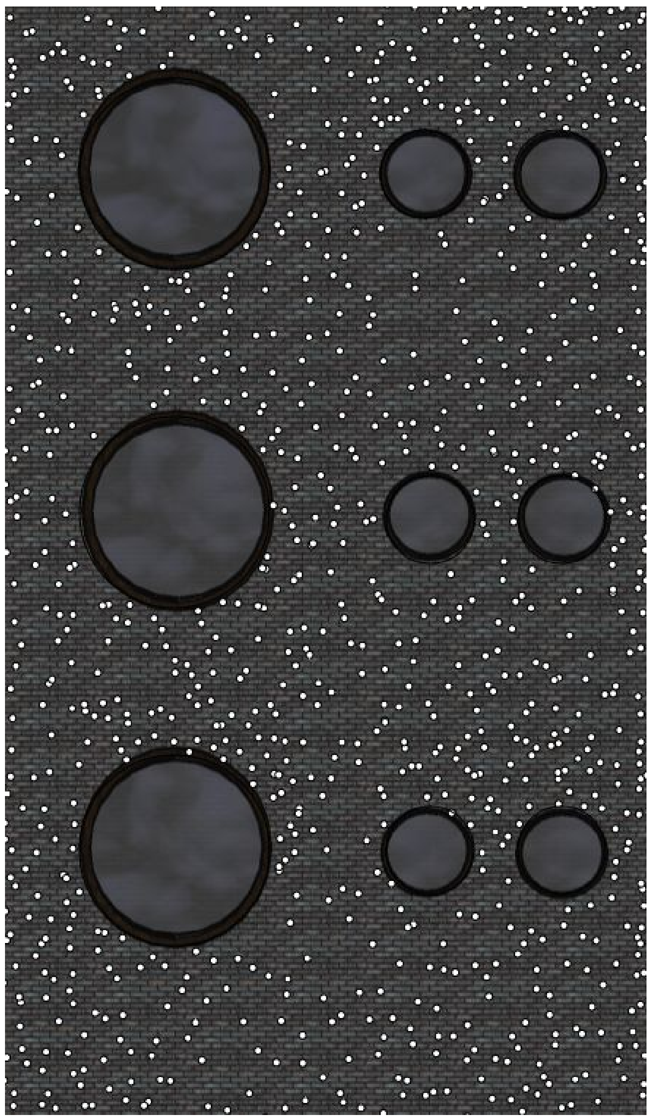

Fig. 26 Facade and Boids

At this point I was able to start experimenting with the variables to create different effects. The following table shows the modifications to the variables used to create each installation:

	Original	Control	Small/Slow	Small/Fast	Large/Slow	Large/Fast
Separation Range	10.0	60.0	60.0	20.0	60.0	20.0
Alignment Range	30.0	20.0	20.0	20.0	60.0	20.0
Cohesion Range	15.0	60.0	120.0	60.0	60.0	60.0
Separation Strength	15.0	50.0	60.0	20.0	60.0	60.0
Alignment Strength	0.01	0.1	1.5	0.5	1.3	0.15
Cohesion Strength	0.01	0.2	1.5	0.9	1.0	0.10
Boundary Separation Range	20.0	30.0	30.0	10.0	30.0	100.0
Boundary Separation Strength	5.0	100.0	150.0	150.0	140.0	220.0
Maximum Velocity	5.0	1.0	60.0	200.0	50.0	200.0
Number of Boids	2000	1000	1000	1000	500	500
Size of Boid Ellipse	7,7	7,7	7,7	7,7	15,15	15,15

Fig. 27 Table of Variables

The main difficulty encountered when created each version for the test was that a change of one variable inevitably caused a cascade of other unwanted changes, so a great deal of time was spent achieving a balance between the variables. For example, increasing the speed of the Boids meant that the range or strength of the boundaries also had to be increased so that the

Boids did not cross over the boundaries. It was also important to adjust the separation, alignment and cohesion values to ensure that the flocking effect did not dissipate when interacting with the boundary forces.

The following gifs show the different results of the experiments, and are shown without the facade background to highlight the movement of the Boids around the canvas.

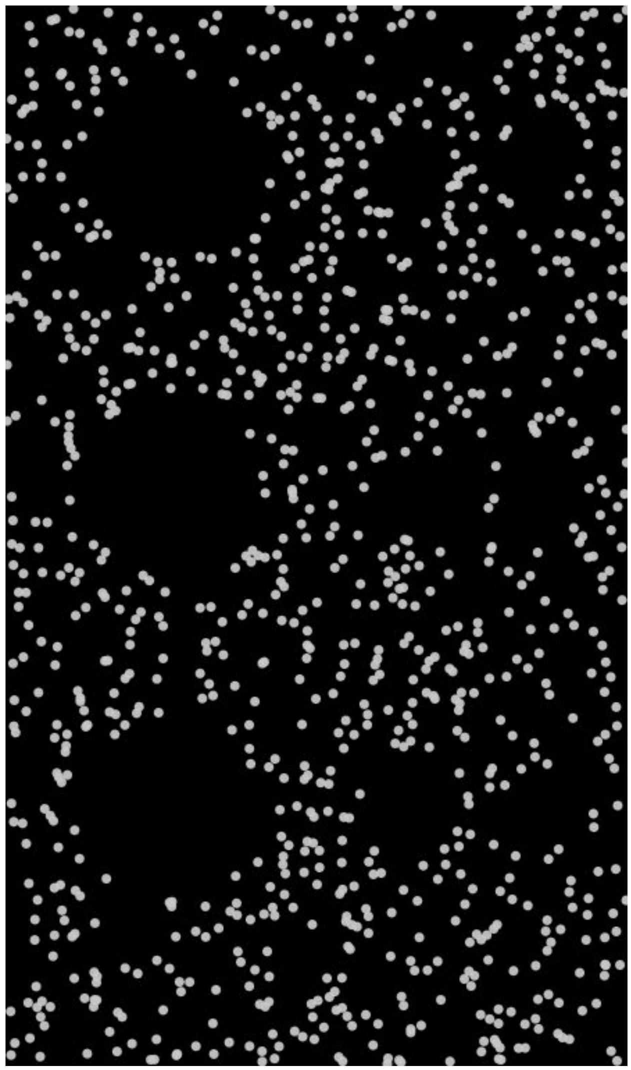

Fig. 28 Control Boids

The velocity of 1.0 causes the Boids to appear stationary, and are distributed randomly but evenly-spaced round the facade. The lack of movement acts as a control, and it is expected that the results will be negative compared to the responses to the other variations.

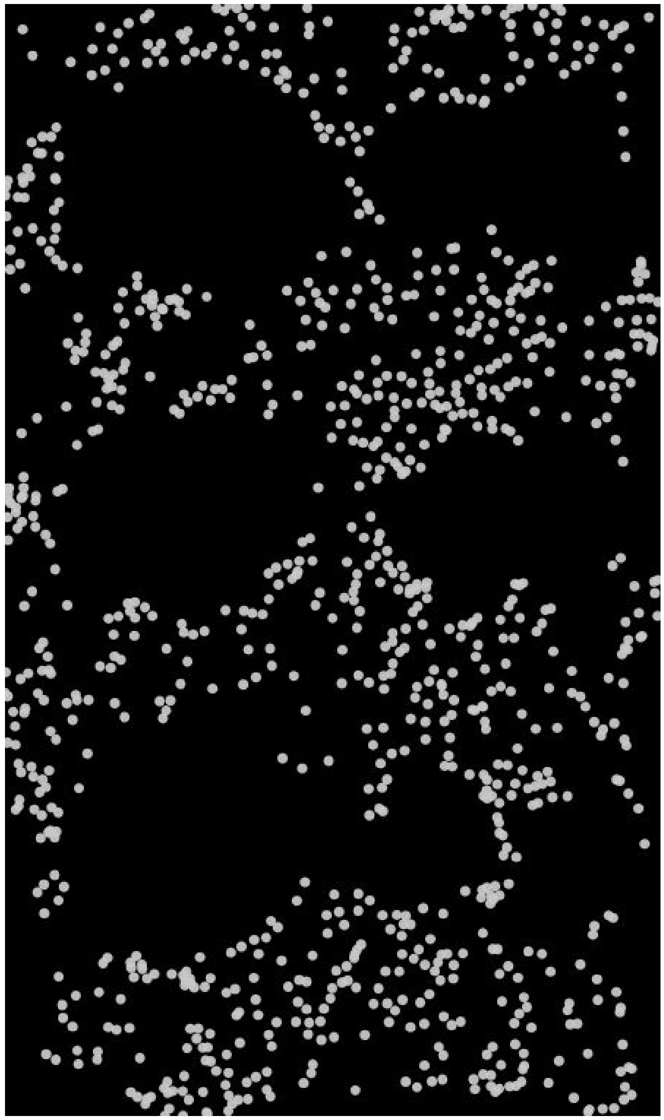

Fig.29 Small Slow Boids

This version shows small boids moving slowly. The relatively high cohesion values cause a pleasing effect as the boids orientate themselves around each other, and also around the boundaries. I found this effect to be quite relaxing while still providing an unusual visual experience.

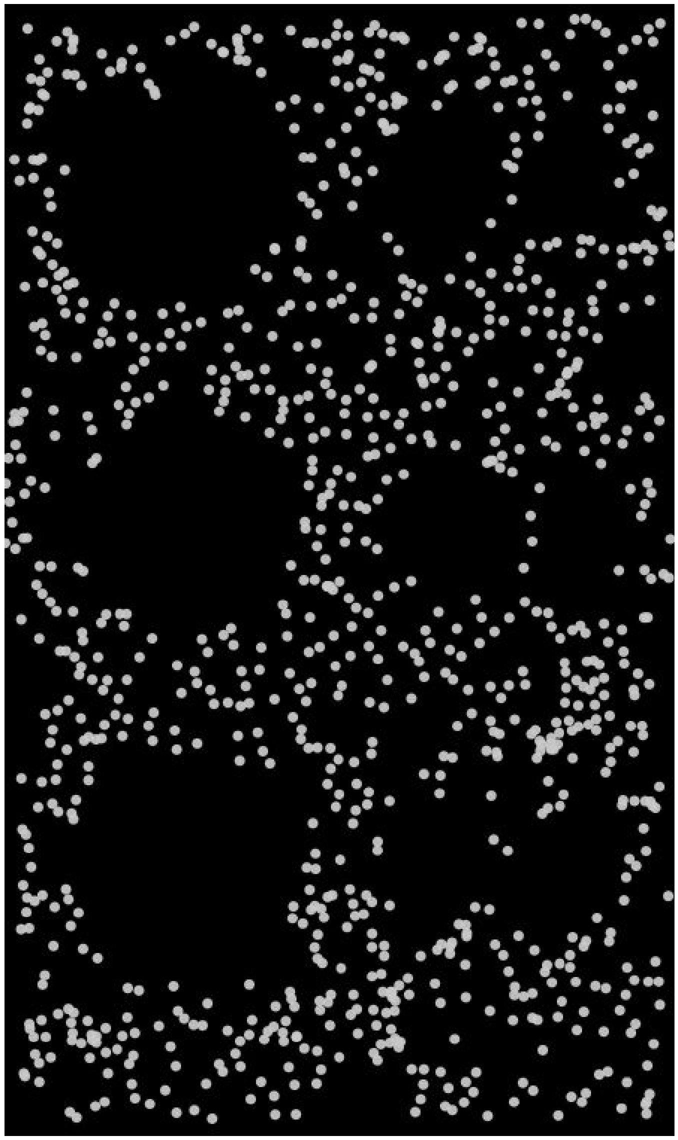

Fig. 30 Small Fast Boids

This version increases the velocity to 200, and required a subsequent increase in the boundary strength to 150.0, with a decrease in boundary range to 10.0 to avoid the Boids passing through the boundaries. By reducing the separation strength to 20.0 and the alignment strength to 0.9 I created an interesting effect where the Boids tended to clump together while still flocking around the boundaries.

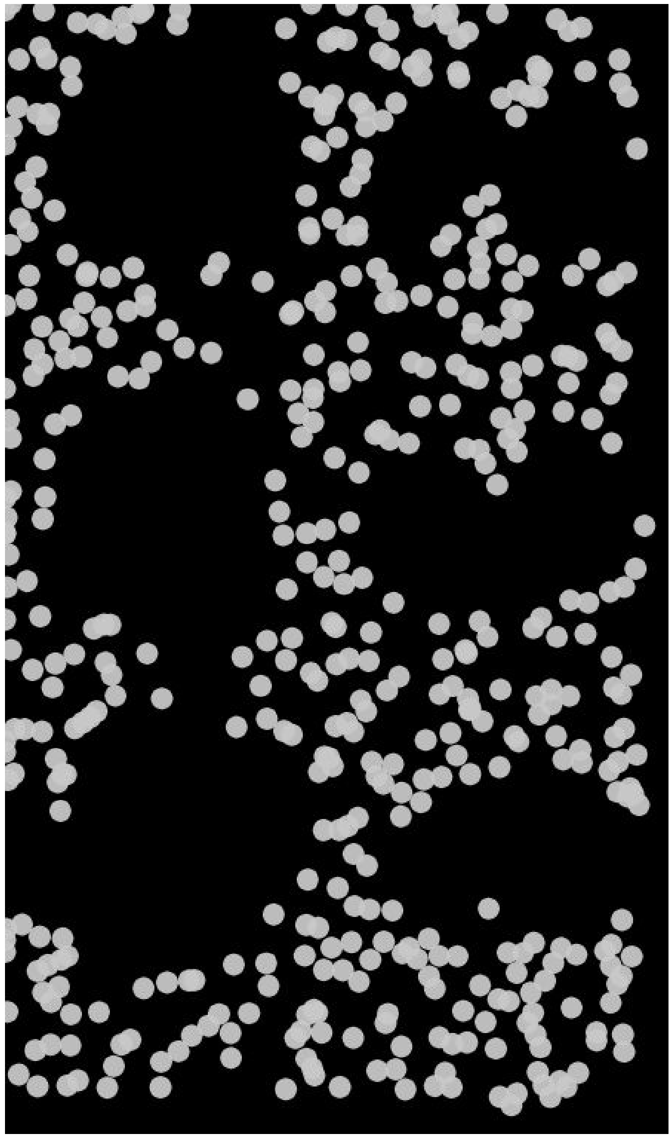

Fig. 31 Large Slow Boids

This version, showing Large Boids moving slowly, reminded me of leaves swirling in a slow moving river, or ther movement of lava in a lava lamp. I achieved this by using a boundary separation strength of 140.0, a velocity of 50.0 and values of alignment strength and cohesion strength of 1.3 and 1.0 respectively. For both of the versions featuring large Boids I halved the total number of Boids to 500 and doubled their size.

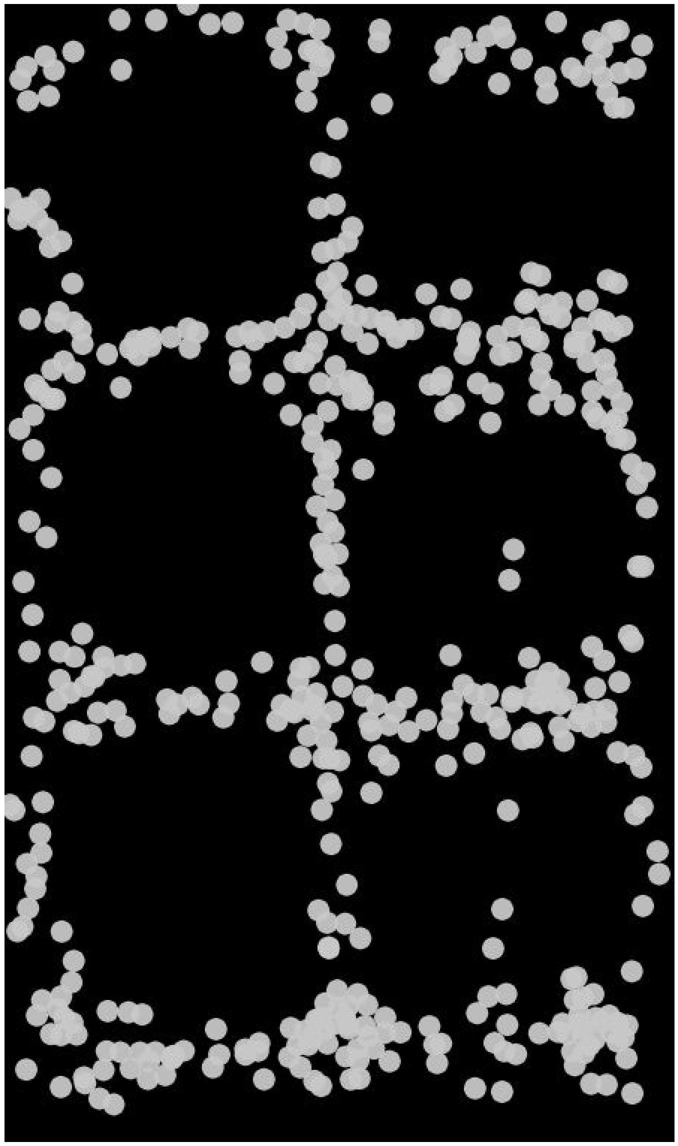

Fig. 32 Large Fast Boids

Finally, to make a fast moving installation with large boids I increased the boundary strength to 220.0 while increasing the boundary range to 100.0. This created the dramatic starting effect seen above, as the Boids are strongly repelled by the boundaries before being quickly attracted to each other and flocking around the windows.

8 Testing and Results

As mentioned previously, the testing process covered both the observers' emotional response to each of the installations and their effectiveness at placemaking. I shall first examine the results of the questions relating emotional response first. Each question on the Likert scale has a maximum value of 5 points so with a total of 15 respondents the maximum score for each was 75 points. The table below shows the total number of points allocated for each question.

The results for the emotional response were as follows:

	Control	Small/Slow	Small/Fast	Large/Slow	Large/Fast
Stimulated	30	50	56	47	63
Relaxed	50	51	41	63	45
Contented	39	49	44	53	48
Stressed	22	31	42	18	28
Bored	52	33	30	32	28
Uncomfortable	17	25	33	24	24

Fig. 33 Table showing results for Emotional Response

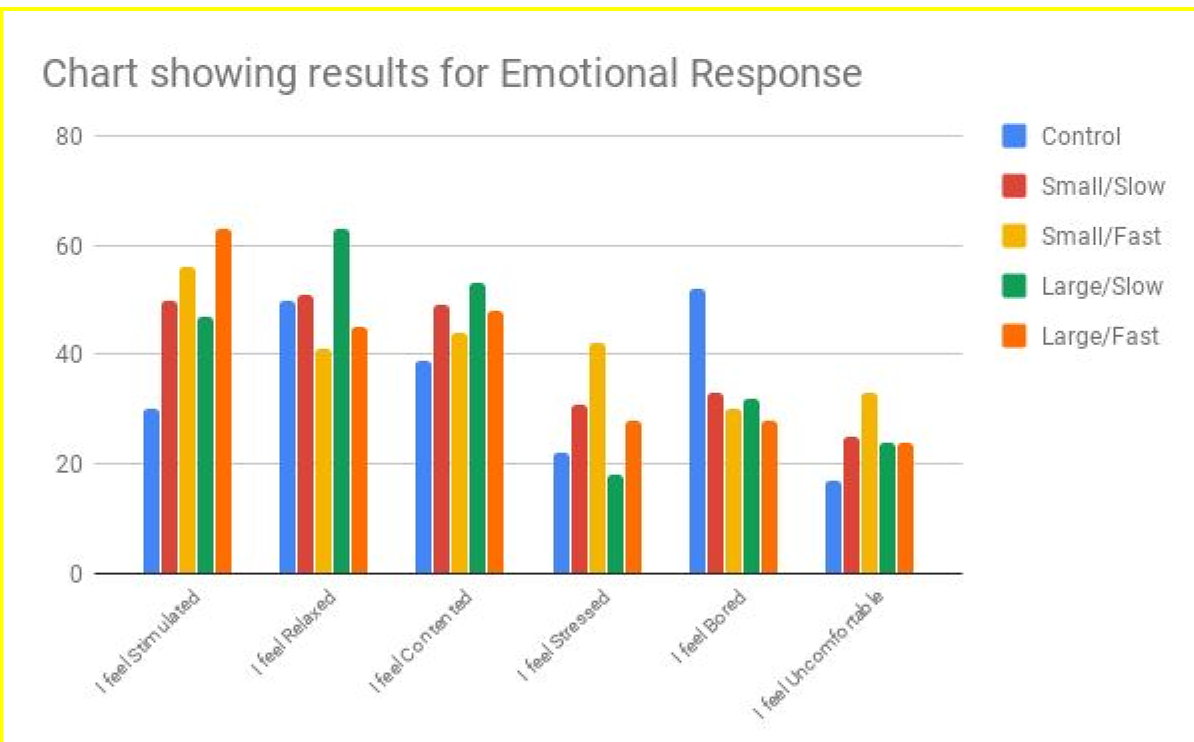

Fig. 34 Chart showing results for Emotional Response

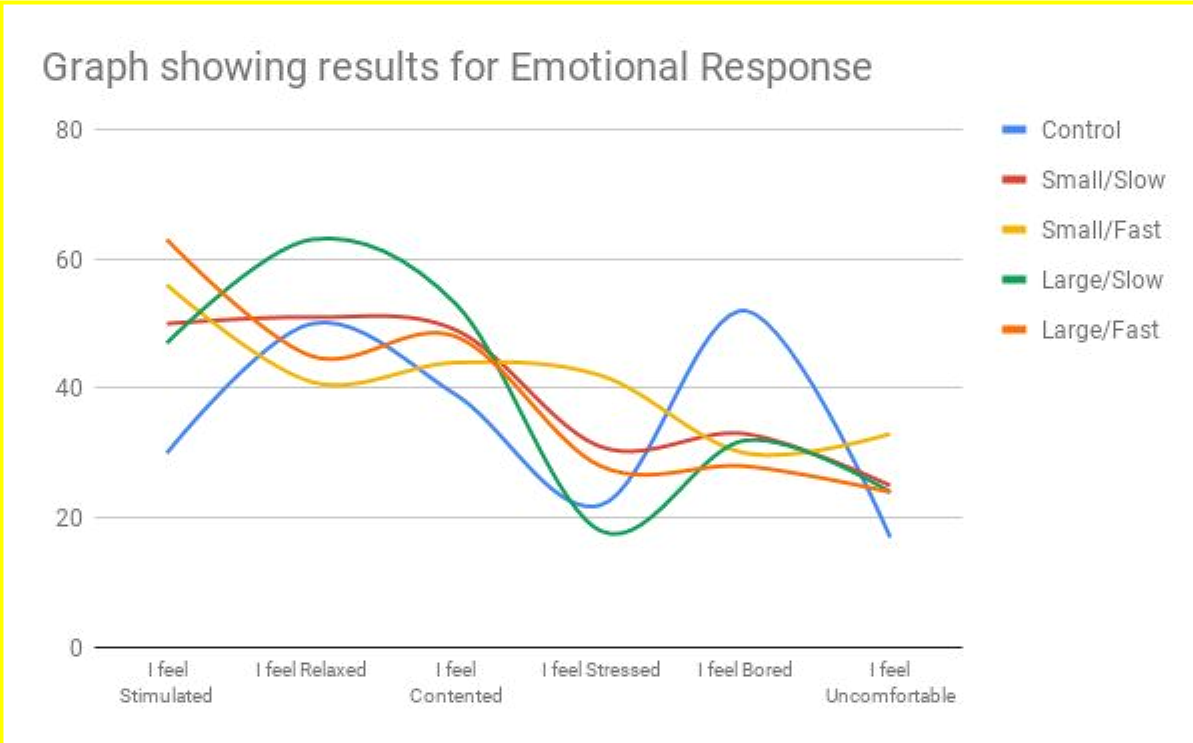

Fig. 35 Graph showing results for Emotional Response

	Control	Small/Slow	Small/Fast	Large/Slow	Large/Fast
Does the installation enhance the facade?	38	46	44	47	58
Does the movement of the Boids look organic?	28	56	41	51	57
Does the movement of the Boids improve the experience?	22	58	42	59	57
Does the speed of the Boids improve the experience?	22	51	42	55	57
Does the size of the Boids improve the experience?	34	48	40	53	55

Fig. 36 Table showing results for Placemaking Response

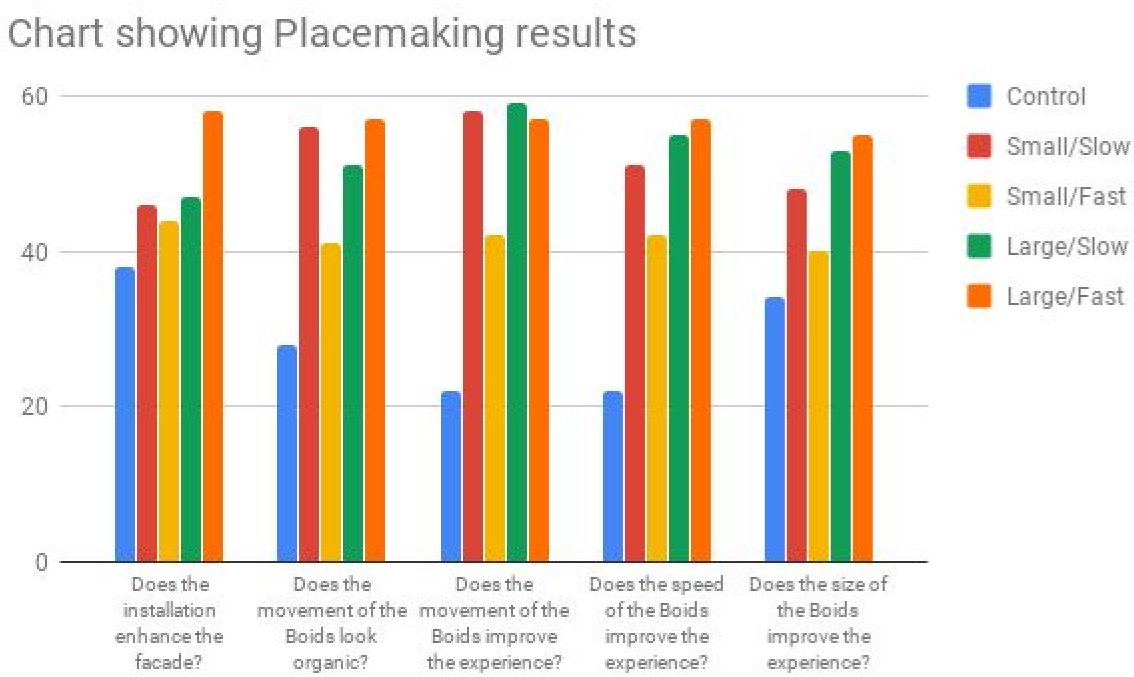

Fig. 37 Chart showing results for Placemaking Response

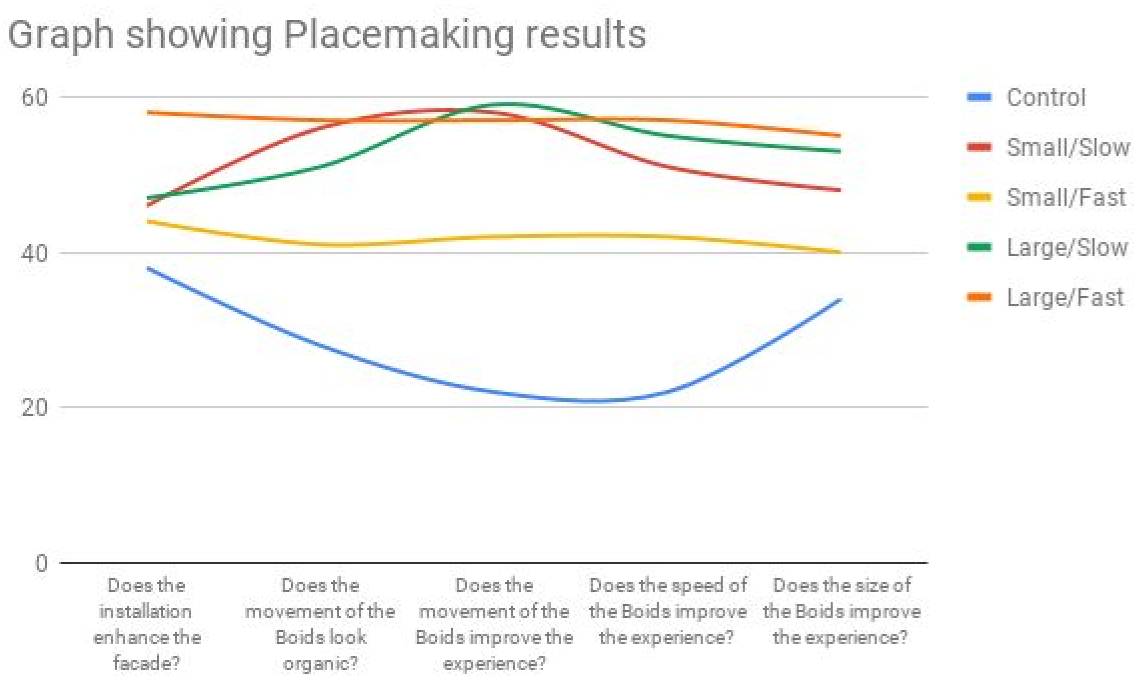

Fig. 38 Graph showing results for Placemaking Response

8.1 Discussion

As expected, the results for control installation were mostly negative, with a majority of respondents feeling bored while watching them. In fact, it could be argued that it would be better to not have an installation that to use the control. The results showed that the small slow boids were judged to be both stimulating and relaxing, which would appear to be a paradox. It may mean that the variables should be adjusted to ensure that the movement of the boids be less ambiguous.

The large slow boids were the preferred option for making the audience feel contented and relaxed, despite comments received from two volunteers who stated that the smaller boids were better, and that they had a preference for the small fast boids as they were “the most natural and organic”. This is interesting, as of the negative emotions accounted for, the small fats boids were outliers in making the observers feel the most stressed and uncomfortable.

The graph of the emotional responses shows a clear downward trend as it moves from the positive to negative emotions, with the large slow boids causing the largest number of respondents to feel most relaxed and conversely, least stressed. We can conclude therefore that the large slow boid installation is the one which contributes most to the emotional well-being of the viewers.

The Placemaking responses are designed to evaluate how the audience perceive the installation in a wider context, to determine if the installation improves the facade and provides and enjoyable experience in a public space. As seen previously, the small fast boids are the least preferred option by a considerable number, so it is safe to say that this version should be disregarded as it does not provide a good experience for the audience.

The observers showed a marked preference for the large fast boids as the version which enhanced the facade the best. It was also scored equally as high for the other four questions, indicating that this version was universally popular among the respondents in terms of its effect on the surroundings. The installation with large slow Boids also scored very highly in terms of how the size, speed and movement of the Boids provided a better experience. This would imply that an even better installation might feature both slow and fast moving large Boids, or that varying the speed of the large Boids according to another input or interaction would improve the viewing experience even more.

The small slow boids were considered to be the most effective in showing the most organic movement, which also improved the experience the most, but given what we have discovered earlier regarding the emotional response it appears that the large Boids would appeal to a greater number of viewers.

The graph showing the placemaking responses bears this out: There is a clear preference for small slow, large slow and large fast Boids. The large Boids show a constant high preference across all 5 questions. In contrast, the small fast Boids were universally disliked in their effectiveness at placemaking.

We can conclude then that if this project were to be enacted in the real world, we would want to use a design featuring slow moving large Boids to provoke feelings of relaxation and contentedness in the audience, but also to feature a design using faster moving Boids to provide a sense of fun and excitement to draw people to a place and establish an previously ignored space as a worthy (temporary) destination. In fact, these results confirm that an effective light installation should be varied in order to be most effective at reaching the greatest number of people with their own tastes and interests.

9 Conclusion

While I am aware that the installations created for this project are not ready for public consumption, given what we know about the enormous quantity of information being disseminated every day over a plethora of public and private media, and the rapidly decreasing attention spans needed to process this in 21st century modern life, I believe that we have gained some valuable insights from this project.

The simplicity inherent in the designs have allowed me to conclude that the size and speed of the Boids do have an impact on the viewer: larger Boids have a more favourable impact than small ones, perhaps because they are easier to focus on, as do their speed of movement. Slower moving boids provoke feelings of relaxation and contentedness while faster moving Boids provide a better overall experience.

In this way, I have successfully answered my research question “How may we provoke different emotional responses in observers by projecting interactive generative art onto a facade?” by demonstrating how changes to a visual light installation can affect feelings and perceptions of its observers.

As technology advances and becomes cheaper and more accessible, we will surely see new and ever evolved media in which we transmit and receive information. In the words of Stojsic (2017), “the contemporary city requires rethinking of the urban space that is now supposed to integrate empathetic and responsive urban digital media...the goal of relational aesthetics art is to create a social circumstance; it operates in the realm of constructed social environment in which the viewer’s experience becomes the art.”

10 Future Works

With a project such as this there is scope for future work in two areas, namely the complexity of the code used in the design and the evaluation of the emotional response. Due to time constraints the code used in this thesis is necessarily basic, so with more time to increase the complexity of the design (and my own coding skills) I would make the following improvements:

I would include ‘predator’ Boids into the design. These are Boids that use the same rules of attraction, cohesion and separation to chase the ‘prey’ Boids, which in turn flee from the predators much like a a school of fish in the open sea. An example of this was written by Abel Jnsm and can be found at the following webPg.:
https://www.openprocessing.org/sketch/126516.
This would make the design more exciting and realistic, and would certainly capture the attention of the observers, especially if they were able to interact with the predators.

Other improvements could be to make the design 3D to provide the illusion of depth on the facade which might also draw more visitors to the installation. Further to this the design could be combined with other effects such as changing colours or Perlin noise to make a more aesthetically pleasing experience.

The level and type of interactions could also be experimented with. The use of cameras such as the Kinect could capture the movement of people or traffic passing by the hotel which could then be translated into changes in the motion of the Boids, as seen in section 3.2.8 with Moura’s ‘B/SIDE’ installation.

Regarding the testing phase of the project, an expanded range of scenarios could allow for the testing of a wider range of emotion response in a controlled environment. For instance, there could be a greater focus on negative responses: could a design make the observer feel anxious or nauseous? In what situations would this be appropriate? It would be worth using the Circumplex of affectation model to identify and test a wider emotional range. I would also ensure that I had at least double the sample size to make the results statistically significant.

For greater accuracy and realism a scale model of the facade could be built so that the full visual effect of projection mapping could be experienced, as opposed to mimicking the effect on a 2D surface.

The ideal solution, of course, would be to take a design using large Boids moving fast and slowly, and use this as a basis to develop a narrative tailored to a specific building or space.
By adding more interactive features, colours etc I believe there would be the potential to create a viable installation.

11 References

aadebdeb, "boid with boundaries" http://www.openprocessing.org/sketch/540690
Licensed under Creative Commons Attribution ShareAlike
https://creativecommons.org/licenses/by-sa/3.0
https://creativecommons.org/licenses/GPL/2.0/

Arcara, Olivia Rose; Mapping the Mundane,The University at Buffalo, State University of New York, 2016

Böhme, Gernot; Atmosphere as mindful physical presence in space, Journal for Architecture Issue 91, 2013

Böhme, Gernot; The art of the stage set as a paradigm for an aesthetics of atmospheres, http://ambiances.revues.org/315 ISSN: 2266-839X, 2013

Brynskov, Martin; Dalsgaard, Peter; Halskov, Kim; Understanding Media Architecture (Better): One Space, Three Cases, Center for Advanced Visualization and Interaction, Aarhus University, 2013

Caplescu, Oana; Augmented Spaces - A Look beyond Advertising, Urbanism. Arhitectură. Construcții • Vol. 5 • Nr. 1 • 2014

Dewey, J. (1934); The Aesthetic in Experience, in Art as Experience. New York: G. Putnam's Sons

Fritsch, Jonas; Dalgaard, Peter; Media Façades beyond interaction, OZCHI 2008, December 8-12, 2008

Gehring, Sven Hendrik; Interaction with Media Facades: The design of interactive systems for large-scale urban screens, Universitat des Saarlandes, 2014

Hale, Jonathan; Merleau-Ponty for Architects, Routledge, 2017

Hansen, E.K.; Process Model of the Architectural Experiment, combining knowledge in the creative design process, 2014

Hespanhol, Luke; Haeusler, M. Hank; Tomitsch, Martin; Tscherteu, Gernot; Media Architecture Compendium, AVEdition Gmbh, 2017

Hespanhol, Luke; Cities of Electronic Clay: Media Architecture for Malleable Public Spaces pg 55 (Hespanhol et.al.)

Haeusler, M. Hank; Media Facades: History, Technology, Content, AVEdition Gmbh, 2009

Huhtamo, Erkki; Pre-envisioning Mediatecture: a media-archaeological perspective (Mediatecture, Kronhagel, Ed. 2010)

Jackson, Davina; Superlux: Smart Light Art, Design and Architecture for Cities, Thames and Hudson, 2015

Jones, Brett; The Illustrated History of Projection Mapping, http://projection-mapping.org/the-history-of-projection-mapping/, 2012

Krautsack, Daniela; 3D Projection Mapping and its Impact on Media & Architecture in Contemporary and Future Urban Spaces, Journal of the New Media Caucus | ISSN: 1942-017X, Summer 2011: v.07 n.01

Kronhagel, Christoph (Ed.); Mediatecture, SpringerWienNewYork, 2010

Lieser, Wolf; Digital Art for Media Facades, found in Mediatecture, Kronhagel et al, Springer Wien New York 2010

Lynch, Patrick; Projection Mapping Light Show Tells the Story of the Guggenheim Bilbao on Its 20th Anniversary, 2017, https://www.archdaily.com/881591/projection-mapping-light-show-tells-the-story-of-the-guggenheim-bilbao-on-its-20th-anniversary

Merleau-Ponty, Phenomenology of Perception, trans Colin Smith, London and New York, Routledge 1962

Paul, Christiane; Digital Art , Thames and Hudson, (3rd ed.) 2015

Pearson, Matt; Generative Art: A Practical Guide using Processing, Manning, 2011

Pihlajaniemi, Henrika; Designing and experiencing adaptive lighting: case studies with adaptation, interaction and participation, 2007 http://jultika.oulu.fi/Record/isbn978-952-62-1090-2

Posner, Jonathan; Russell, James; Peterson, Bradley; The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology, PMC2367156, 2005

Project for Public Spaces; What is Placemaking? (unaccredited) https://www.pps.org/article/what-is-placemaking

Raheja, Kishore M; Architecture + Media Digital Augmentations through Meta Media, Mappings, Real-time Communication and Interactivity, 2007

Reynolds, Craig; Boids: Background and Update, 1995, http://www.red3d.com/cwr/boids/

Schielke, Thomas; Light Matters: 3D Video Mapping, Making Architecture The Screen for Our Urban Stories, 2013
https://www.archdaily.com/432355/light-matters-how-3d-video-mapping-makes-architecture-the-screen-for-our-urban-stories

Singer, Harald; Origins of Mediatecture in ag4 pg 36 (Mediatecture, Kronhagel, Ed. 2010)

Stojšić, Milica; (New) Media Facades: Architecture and/as a Medium in Urban Context
AM Journal of Art and Media Studies, April 2017

Vogels, Ingrid; Atmosphere Metrics, Technische Universiteit Eindhoven, 2008

Werner, James Patrick; Immersive Installation Art: Digital technology, its philosophies, and the rise of a new genre in fine art, Columbia University, 2007

.

12 Appendix

12.1 Code

To view the designs, please copy and paste the code into the following website: https://www.openprocessing.org/sketch/540690

12.1.1 Control code

```
var SEPARATION_RANGE = 60.0;
var ALIGNMENT_RANGE = 20.0;
var COHESION_RANGE = 60.0;
var SEPARATION_STRENGH =
50.0;
var ALIGNMENT_STRENGH = 0.1;
var COHESION_STRENGH = 0.2;

var
BOUNDARY_SEPARATION_RANG
E = 30.0;
var
BOUNDARY_SEPARATION_STRE
NGTH = 100.0;

var MAX_VELOCITY = 1.0;

var N = 1000;
var positions;
var velocities;
var forces;

var gridBucket;
var boundaryBucket;
var boundaries;
var maxRange;
var prevMillis = 0.0;


document.addEventListener('keydow
n', function(event){
  switch(event.key) {
    case '2':
      MAX_VELOCITY += 1;
      console.log('max velocity' +
MAX_VELOCITY);
        break;
  case '1':
      MAX_VELOCITY -= 1;
      console.log('min velocity' +
MAX_VELOCITY);
        break;

            case 'w':
      SEPARATION_STRENGH += 1;
      console.log('increase sep
strength' + MAX_VELOCITY);
        break;
  case 'q':
      SEPARATION_STRENGH -= 1;
      console.log('decrease sep
strength' + MAX_VELOCITY);
        break;

  case 'w':
      ALIGNMENT_STRENGH += 1;
      console.log('increase alignment
strength' + MAX_VELOCITY);
        break;
  case 'q':
      ALIGNMENT_STRENGH -= 1;
      console.log('decrease alignment
strength' + MAX_VELOCITY);
        break;

            case 's':
      COHESION_STRENGH += 1;
      console.log('increase cohesion
strength' + MAX_VELOCITY);
        break;
  case 'a':
      COHESION_STRENGH -= 1;
      console.log('decrease cohesion
strength' + MAX_VELOCITY);
        break;

            case 'x':

BOUNDARY_SEPARATION_STRE
NGTH += 1;
      console.log('increase bss' +
MAX_VELOCITY);
        break;
  case 'z':

BOUNDARY_SEPARATION_STRE
NGTH -= 1;
      console.log('decrease bss' +
MAX_VELOCITY);
        break;


  case 'r':
      console.log('reset!');
      SEPARATION_RANGE = 50.0;
       ALIGNMENT_RANGE = 30.0;
       COHESION_RANGE = 50.0;
       SEPARATION_STRENGH =
30.0;
       ALIGNMENT_STRENGH =
0.01;
       COHESION_STRENGH = 0.1;


BOUNDARY_SEPARATION_RANG
E = 50.0;

BOUNDARY_SEPARATION_STRE
NGTH = 50.0;

       MAX_VELOCITY = 5.0;
     default:
        console.log('default - no key
pressed');
}
});

function setup() {
  createCanvas(448, 761);

  var myImage = new Image(448,
761);
  myImage.src =
'https://s6.postimg.cc/d0s0fev3l/Thes
is_pic3cropped.png';
  document.body.style.background =
"url('https://s6.postimg.cc/d0s0fev3l/
Thesis_pic3cropped.png')
no-repeat";
  initialize();
}

function initialize() {
  createBoundaryBucket();
  positions = new Array(N);
  velocities = new Array(N);
  forces = new Array(N);
  for (var i = 0; i < N; i++) {
    positions[i] = getInitialPosition();
    velocities[i] = randomInCircle();
    forces[i] = createVector(0, 0);
  }

  maxRange =
max([SEPARATION_RANGE,
```

```
ALIGNMENT_RANGE,
COHESION_RANGE]);
  createGridBucket();
}

function getInitialPosition() {
  while(true) {
    var p =
createVector(random(width),
random(height));
    var isAvailable =
boundaries.every(function(boundary)
{
      if (boundary.isCircleBoundary) {
        return p.dist(boundary.center)
>= boundary.radius;
      } else {
        return true;
      }
    });
    if (isAvailable) return p;
  }
}

function createGridBucket() {
  var x = ceil(width / maxRange) + 2;
  var y = ceil(height / maxRange) + 2;
  gridBucket = new Array(x);
  for (var xi = 0; xi < x; xi++) {
    gridBucket[xi] = new Array(y);
    for (var yi = 0; yi < y; yi++) {
      gridBucket[xi][yi] = [];
    }
  }
}

function createBoundaryBucket() {
  boundaries = [
    { // left wall
      calcDistance: function(p) {return
p.x;},
      calcNormal: function(p) {return
createVector(1.0, 0.0);}
    }, { // right wall
      calcDistance: function(p) {return
max(width - p.x, 0.0);},
      calcNormal: function(p) {return
createVector(-1.0, 0.0);}
    }, { // top wall
      calcDistance: function(p) {return
p.y;},
      calcNormal: function(p) {return
createVector(0.0, 1.0);}
    }, { // bottom wall
      calcDistance: function(p) {return
max(height - p.y, 0.0);},
      calcNormal: function(p) {return
createVector(0.0, -1.0);}
    } ,
                              new
CircleBoundary(createVector(120,
117), 65), //big three
                              new
CircleBoundary(createVector(120,
347), 65),
                              new
CircleBoundary(createVector(120,
577), 65),

                              new
CircleBoundary(createVector(291,
117), 30),//top
                              new
CircleBoundary(createVector(382,
117), 30),

                              new
CircleBoundary(createVector(385,
352), 30), //middle
                              new
CircleBoundary(createVector(295,
352), 30),

                              new
CircleBoundary(createVector(385,
577), 30), //bottom
                              new
CircleBoundary(createVector(295,
577), 30)

  ];

  var x = ceil(width /
BOUNDARY_SEPARATION_RANG
E) + 2;
  var y = ceil(height /
BOUNDARY_SEPARATION_RANG
E) + 2;
  boundaryBucket = new Array(x);
  for (var xi = 0; xi < x; xi++) {
    boundaryBucket[xi] = new
Array(y);
    for (var yi = 0; yi < y; yi++) {
      boundaryBucket[xi][yi] = [];
      if (xi === 0 || xi === x - 1 || yi ===
0 || yi === y - 1) continue;

boundaries.forEach(function(boundar
y) {
        var d = min([

boundary.calcDistance(getBoundary
BucketPosition({x: xi, y: yi})),

boundary.calcDistance(getBoundary
BucketPosition({x: xi + 1, y: yi})),

boundary.calcDistance(getBoundary
BucketPosition({x: xi, y: yi + 1})),

boundary.calcDistance(getBoundary
BucketPosition({x: xi + 1, y: yi + 1}))
        ]);
        if (d <
BOUNDARY_SEPARATION_RANG
E) {

boundaryBucket[xi][yi].push(boundar
y);
        }
      });
    }
  }
}

function
getBoundaryBucketPosition(idx) {
  return createVector(
    (idx.x - 1) *
BOUNDARY_SEPARATION_RANG
E,
    (idx.y - 1) *
BOUNDARY_SEPARATION_RANG
E
  );
}

function randomInCircle() {
  var r = random();
  var a = random(TWO_PI);
  return createVector(r * cos(a), r *
sin(a));
}

function draw() {
  clear();
  render();
  update();
}

function render() {

  fill(255);
  for (var i = 0; i < N; i++) {
    var pos = positions[i];
    fill(200,200,200,240);
                        noStroke();
    ellipse(pos.x, pos.y, 7, 7);
  }
  noFill();
```

```
boundaries.forEach(function(boundar
y) {
    if (boundary.isCircleBoundary) {
      ellipse(boundary.center.x,
boundary.center.y, boundary.radius *
2.0, boundary.radius * 2.0);
    }
  });
}

function update() {
  updateGridBucket();
  calcSeparation();
  calcAlignment();
  calcCohesion();
  calcBoundarySeparation();
  move();
  resolveBoundary();
}

function updateGridBucket() {
  clearGridBucket();
  for (var i = 0; i < N; i++) {
    var bucket = getBucketIdx(i);

gridBucket[bucket.x][bucket.y].push(i
);
  }
}

function clearGridBucket() {
  for (var xi = 0; xi <
gridBucket.length; xi++) {
    for (var yi = 0; yi <
gridBucket[xi].length; yi++) {
      gridBucket[xi][yi] = [];
    }
  }
}

function getBucketIdx(idx) {
  return {
    x: floor(positions[idx].x /
maxRange) + 1,
    y: floor(positions[idx].y /
maxRange) + 1
  }
}

function getBoundaryBucketIdx(idx) {
  return {
    x: floor(positions[idx].x /
BOUNDARY_SEPARATION_RANG
E) + 1,
    y: floor(positions[idx].y /
BOUNDARY_SEPARATION_RANG
E) + 1
  }
}

function weight(dist, maxDist) {
  return dist < maxDist ? 1.0 - (dist /
maxDist) : 0.0;
}

function calcSeparation() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var force = createVector(0.0, 0.0);
    var bucket = getBucketIdx(i);
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {

gridBucket[xi][yi].forEach(function(j) {
          if (i === j) return;
          var posj = positions[j];
          var d = posi.dist(posj);
          var w = weight(d,
SEPARATION_RANGE);
          if (w <= 0.0) return;
          force.add(p5.Vector.sub(posi,
posj).normalize().mult(w));
        });
      }
    }

forces[i].add(force.mult(SEPARATIO
N_STRENGH));
  }
}

function calcAlignment() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var sumW = 0;
    var sumVel = createVector(0.0,
0.0);
    var bucket = getBucketIdx(i);
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {

gridBucket[xi][yi].forEach(function(j) {
          if (i === j) return;
          var posj = positions[j];
          var d = posi.dist(posj);
          var w = weight(d,
ALIGNMENT_RANGE);
          if (w <= 0.0) return;
          sumW = w;

sumVel.add(velocities[j].copy().mult(
w));
        });
      }
    }
  if (sumW < 0.001) continue;
  sumVel.div(sumW);
  var force = p5.Vector.sub(sumVel,
velocities[i]);

forces[i].add(force.mult(ALIGNMENT
_STRENGH));
  }
}

function calcCohesion() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var force = createVector(0.0, 0.0);
    var bucket = getBucketIdx(i);
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {

gridBucket[xi][yi].forEach(function(j) {
          if (i === j) return;
          var posj = positions[j];
          var d = posi.dist(posj);
          var w = weight(d,
COHESION_RANGE);
          if (w <= 0.0) return;
          force.add(p5.Vector.sub(posj,
posi).normalize().mult(w));
        });
      }
    }

forces[i].add(force.mult(COHESION_
STRENGH));
  }
}

function calcBoundarySeparation() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var force = createVector(0.0, 0.0);
    var bucket =
getBoundaryBucketIdx(i);
    var checkedBoundaries = [];
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {
```

```
boundaryBucket[xi][yi].forEach(functi
on(boundary) {
        if
(checkedBoundaries.some(function(c
hecked) {checked === boundary;}))
return;

checkedBoundaries.push(boundary);
        var d =
boundary.calcDistance(posi);
        var w = weight(d,
BOUNDARY_SEPARATION_RANG
E);
        if (w <= 0.0) return;

force.add(boundary.calcNormal(posi)
.mult(w));
      });
    }
  }

forces[i].add(force.mult(BOUNDARY
_SEPARATION_STRENGTH));
 }
}

function move() {
 var currentMillis = millis();
 var dt = min((currentMillis -
prevMillis) * 0.0001, 0.02);
 for (var i = 0; i < N; i++) {

velocities[i].add(p5.Vector.mult(force
s[i], dt));

velocities[i].limit(MAX_VELOCITY);

positions[i].add(p5.Vector.mult(veloci
ties[i], dt));
  forces[i].set(0.0, 0.0);
 }
 prevMillis = currentMillis;
}

function resolveBoundary() {
 for (var i = 0; i < N; i++) {
  var pos = positions[i];
  var vel = velocities[i];
  if (pos.x < 0.0) {
   vel.x *= -1;
   pos.x = 0.0;
  }
  if (pos.x >= width) {
   vel.x *= -1;
   pos.x = width - 0.1;
  }
  if (pos.y < 0.0) {
   vel.y *= -1;
   pos.y = 0.0;
  }
  if (pos.y >= height) {
   vel.y *= -1;
   pos.y = height - 0.1;
  }
 }
}

function CircleBoundary(center,
radius) {
 this.center = center;
 this.radius = radius;
 this.isCircleBoundary = true;
}

CircleBoundary.prototype.calcDistan
ce = function(p) {


 return max(p.dist(this.center) -
this.radius, 0.0);
}

CircleBoundary.prototype.calcNorma
l = function(p) {
 return p5.Vector.sub(p,
this.center).normalize();
}
```

12.1.2 Code Small/Slow Boids

```
var SEPARATION_RANGE = 60.0;
var ALIGNMENT_RANGE = 20.0;
var COHESION_RANGE = 120.0;
var SEPARATION_STRENGH =
60.0;
var ALIGNMENT_STRENGH = 1.50;
var COHESION_STRENGH = 1.50;

var
BOUNDARY_SEPARATION_RANG
E = 30.0;
var
BOUNDARY_SEPARATION_STRE
NGTH = 150.0;

var MAX_VELOCITY = 60.0;

var N = 1000;
var positions;
var velocities;
var forces;

var gridBucket;
var boundaryBucket;
var boundaries;
var maxRange;
var prevMillis = 0.0;


document.addEventListener('keydow
n', function(event){
  switch(event.key) {
    case '2':
      MAX_VELOCITY += 1;
      console.log('max velocity' +
MAX_VELOCITY);
        break;
  case '1':
      MAX_VELOCITY -= 1;
      console.log('min velocity' +
MAX_VELOCITY);
        break;

                case 'w':
      SEPARATION_STRENGH += 1;
      console.log('increase sep
strength' + MAX_VELOCITY);
        break;
  case 'q':
      SEPARATION_STRENGH -= 1;
      console.log('decrease sep
strength' + MAX_VELOCITY);
        break;

  case 'w':
      ALIGNMENT_STRENGH += 1;
      console.log('increase alignment
strength' + MAX_VELOCITY);
        break;
  case 'q':
      ALIGNMENT_STRENGH -= 1;
      console.log('decrease alignment
strength' + MAX_VELOCITY);
        break;

                case 's':
      COHESION_STRENGH += 1;
      console.log('increase cohesion
strength' + MAX_VELOCITY);
        break;
  case 'a':
      COHESION_STRENGH -= 1;
      console.log('decrease cohesion
strength' + MAX_VELOCITY);
        break;

                case 'x':

BOUNDARY_SEPARATION_STRE
NGTH += 1;
      console.log('increase bss' +
MAX_VELOCITY);
        break;
  case 'z':

BOUNDARY_SEPARATION_STRE
NGTH -= 1;
      console.log('decrease bss' +
MAX_VELOCITY);
        break;


  case 'r':
      console.log('reset!');
      SEPARATION_RANGE = 50.0;
       ALIGNMENT_RANGE = 30.0;
       COHESION_RANGE = 50.0;
       SEPARATION_STRENGH =
30.0;
       ALIGNMENT_STRENGH =
0.01;
       COHESION_STRENGH = 0.1;


BOUNDARY_SEPARATION_RANG
E = 50.0;

BOUNDARY_SEPARATION_STRE
NGTH = 50.0;
         MAX_VELOCITY = 5.0;
    default:
        console.log('default - no key
pressed');
}
});

function setup() {
  createCanvas(448, 761);

  var myImage = new Image(448,
761);
  myImage.src =
'https://s6.postimg.cc/d0s0fev3l/Thes
is_pic3cropped.png';
  document.body.style.background =
"url('https://s6.postimg.cc/d0s0fev3l/
Thesis_pic3cropped.png')
no-repeat";
  initialize();
}

function initialize() {
  createBoundaryBucket();
  positions = new Array(N);
  velocities = new Array(N);
  forces = new Array(N);
  for (var i = 0; i < N; i++) {
    positions[i] = getInitialPosition();
    velocities[i] = randomInCircle();
    forces[i] = createVector(0, 0);
  }

  maxRange =
max([SEPARATION_RANGE,
ALIGNMENT_RANGE,
COHESION_RANGE]);
  createGridBucket();
}

function getInitialPosition() {
  while(true) {
    var p =
createVector(random(width),
random(height));
    var isAvailable =
boundaries.every(function(boundary)
{
      if (boundary.isCircleBoundary) {
        return p.dist(boundary.center)
>= boundary.radius;
      } else {
        return true;
      }
    });
    if (isAvailable) return p;
```

```
  }
}

function createGridBucket() {
  var x = ceil(width / maxRange) + 2;
  var y = ceil(height / maxRange) + 2;
  gridBucket = new Array(x);
  for (var xi = 0; xi < x; xi++) {
    gridBucket[xi] = new Array(y);
    for (var yi = 0; yi < y; yi++) {
      gridBucket[xi][yi] = [];
    }
  }
}

function createBoundaryBucket() {
  boundaries = [
    { // left wall
      calcDistance: function(p) {return
p.x;},
      calcNormal: function(p) {return
createVector(1.0, 0.0);}
    }, { // right wall
      calcDistance: function(p) {return
max(width - p.x, 0.0);},
      calcNormal: function(p) {return
createVector(-1.0, 0.0);}
    }, { // top wall
      calcDistance: function(p) {return
p.y;},
      calcNormal: function(p) {return
createVector(0.0, 1.0);}
    }, { // bottom wall
      calcDistance: function(p) {return
max(height - p.y, 0.0);},
      calcNormal: function(p) {return
createVector(0.0, -1.0);}
    } ,
                        new
CircleBoundary(createVector(120,
117), 65), //big three
                        new
CircleBoundary(createVector(120,
347), 65),
                        new
CircleBoundary(createVector(120,
577), 65),

                        new
CircleBoundary(createVector(291,
117), 30),//top
                        new
CircleBoundary(createVector(382,
117), 30),
                        new
CircleBoundary(createVector(385,
352), 30), //middle
                        new
CircleBoundary(createVector(295,
352), 30),

                        new
CircleBoundary(createVector(385,
577), 30), //bottom
                        new
CircleBoundary(createVector(295,
577), 30)

  ];

  var x = ceil(width /
BOUNDARY_SEPARATION_RANG
E) + 2;
  var y = ceil(height /
BOUNDARY_SEPARATION_RANG
E) + 2;
  boundaryBucket = new Array(x);
  for (var xi = 0; xi < x; xi++) {
    boundaryBucket[xi] = new
Array(y);
    for (var yi = 0; yi < y; yi++) {
      boundaryBucket[xi][yi] = [];
      if (xi === 0 || xi === x - 1 || yi ===
0 || yi === y - 1) continue;

boundaries.forEach(function(boundar
y) {
        var d = min([

boundary.calcDistance(getBoundary
BucketPosition({x: xi, y: yi})),

boundary.calcDistance(getBoundary
BucketPosition({x: xi + 1, y: yi})),

boundary.calcDistance(getBoundary
BucketPosition({x: xi, y: yi + 1})),

boundary.calcDistance(getBoundary
BucketPosition({x: xi + 1, y: yi + 1}))
        ]);
        if (d <
BOUNDARY_SEPARATION_RANG
E) {

boundaryBucket[xi][yi].push(boundar
y);
        }
      });
    }
  }
}

function
getBoundaryBucketPosition(idx) {
  return createVector(
    (idx.x - 1) *
BOUNDARY_SEPARATION_RANG
E,
    (idx.y - 1) *
BOUNDARY_SEPARATION_RANG
E
  );
}

function randomInCircle() {
  var r = random();
  var a = random(TWO_PI);
  return createVector(r * cos(a), r *
sin(a));
}

function draw() {
  clear();
  render();
  update();
}

function render() {

  fill(255);
  for (var i = 0; i < N; i++) {
    var pos = positions[i];
    fill(200,200,200,240);
                        noStroke();
    ellipse(pos.x, pos.y, 8, 8);
  }
  noFill();

boundaries.forEach(function(boundar
y) {
    if (boundary.isCircleBoundary) {
      ellipse(boundary.center.x,
boundary.center.y, boundary.radius *
2.0, boundary.radius * 2.0);
    }
  });
}

function update() {
  updateGridBucket();
  calcSeparation();
  calcAlignment();
  calcCohesion();
  calcBoundarySeparation();
  move();
  resolveBoundary();
}
```

```
function updateGridBucket() {
  clearGridBucket();
  for (var i = 0; i < N; i++) {
    var bucket = getBucketIdx(i);

gridBucket[bucket.x][bucket.y].push(i
);
  }
}

function clearGridBucket() {
  for (var xi = 0; xi <
gridBucket.length; xi++) {
    for (var yi = 0; yi <
gridBucket[xi].length; yi++) {
      gridBucket[xi][yi] = [];
    }
  }
}

function getBucketIdx(idx) {
  return {
    x: floor(positions[idx].x /
maxRange) + 1,
    y: floor(positions[idx].y /
maxRange) + 1
  }
}

function getBoundaryBucketIdx(idx) {
  return {
    x: floor(positions[idx].x /
BOUNDARY_SEPARATION_RANG
E) + 1,
    y: floor(positions[idx].y /
BOUNDARY_SEPARATION_RANG
E) + 1
  }
}

function weight(dist, maxDist) {
  return dist < maxDist ? 1.0 - (dist /
maxDist) : 0.0;
}

function calcSeparation() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var force = createVector(0.0, 0.0);
    var bucket = getBucketIdx(i);
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {

gridBucket[xi][yi].forEach(function(j) {
          if (i === j) return;
          var posj = positions[j];
          var d = posi.dist(posj);
          var w = weight(d,
SEPARATION_RANGE);
          if (w <= 0.0) return;
          force.add(p5.Vector.sub(posi,
posj).normalize().mult(w));
        });
      }
    }

forces[i].add(force.mult(SEPARATIO
N_STRENGH));
  }
}

function calcAlignment() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var sumW = 0;
    var sumVel = createVector(0.0,
0.0);
    var bucket = getBucketIdx(i);
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {

gridBucket[xi][yi].forEach(function(j) {
          if (i === j) return;
          var posj = positions[j];
          var d = posi.dist(posj);
          var w = weight(d,
ALIGNMENT_RANGE);
          if (w <= 0.0) return;
          sumW = w;

sumVel.add(velocities[j].copy().mult(
w));
        });
      }
    }
  if (sumW < 0.001) continue;
  sumVel.div(sumW);
  var force = p5.Vector.sub(sumVel,
velocities[i]);

forces[i].add(force.mult(ALIGNMENT
_STRENGH));
  }
}

function calcCohesion() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var force = createVector(0.0, 0.0);
    var bucket = getBucketIdx(i);
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {

gridBucket[xi][yi].forEach(function(j) {
          if (i === j) return;
          var posj = positions[j];
          var d = posi.dist(posj);
          var w = weight(d,
COHESION_RANGE);
          if (w <= 0.0) return;
          force.add(p5.Vector.sub(posj,
posi).normalize().mult(w));
        });
      }
    }

forces[i].add(force.mult(COHESION_
STRENGH));
  }
}

function calcBoundarySeparation() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var force = createVector(0.0, 0.0);
    var bucket =
getBoundaryBucketIdx(i);
    var checkedBoundaries = [];
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {

boundaryBucket[xi][yi].forEach(functi
on(boundary) {
          if
(checkedBoundaries.some(function(c
hecked) {checked === boundary;}))
return;

checkedBoundaries.push(boundary);
          var d =
boundary.calcDistance(posi);
          var w = weight(d,
BOUNDARY_SEPARATION_RANG
E);
          if (w <= 0.0) return;

force.add(boundary.calcNormal(posi)
.mult(w));
        });
      }
    }
```

```
forces[i].add(force.mult(BOUNDARY
_SEPARATION_STRENGTH));
  }
}

function move() {
  var currentMillis = millis();
  var dt = min((currentMillis -
prevMillis) * 0.0001, 0.02);
  for (var i = 0; i < N; i++) {

velocities[i].add(p5.Vector.mult(force
s[i], dt));

velocities[i].limit(MAX_VELOCITY);

positions[i].add(p5.Vector.mult(veloci
ties[i], dt));
    forces[i].set(0.0, 0.0);
  }
  prevMillis = currentMillis;
}

function resolveBoundary() {
  for (var i = 0; i < N; i++) {
    var pos = positions[i];
    var vel = velocities[i];
    if (pos.x < 0.0) {
      vel.x *= -1;
      pos.x = 0.0;
    }
    if (pos.x >= width) {
      vel.x *= -1;
      pos.x = width - 0.1;
    }
    if (pos.y < 0.0) {
      vel.y *= -1;
      pos.y = 0.0;
    }
    if (pos.y >= height) {
      vel.y *= -1;
      pos.y = height - 0.1;
    }
  }
}

function CircleBoundary(center,
radius) {
  this.center = center;
  this.radius = radius;
  this.isCircleBoundary = true;
}

CircleBoundary.prototype.calcDistan
ce = function(p) {
  return max(p.dist(this.center) -
this.radius, 0.0);
}

CircleBoundary.prototype.calcNorma
l = function(p) {
  return p5.Vector.sub(p,
this.center).normalize();
}
```

12.1.3 Code Small/Fast Boids

```
var SEPARATION_RANGE = 20.0;
var ALIGNMENT_RANGE = 20.0;
var COHESION_RANGE = 60.0;
var SEPARATION_STRENGH = 20;
var ALIGNMENT_STRENGH = 0.5;
var COHESION_STRENGH = 0.9;

var
BOUNDARY_SEPARATION_RANG
E = 10;
var
BOUNDARY_SEPARATION_STRE
NGTH = 150;

var MAX_VELOCITY = 200.0;

var N = 1000;
var positions;
var velocities;
var forces;

var gridBucket;
var boundaryBucket;
var boundaries;
var maxRange;
var prevMillis = 0.0;


document.addEventListener('keydow
n', function(event){
  switch(event.key) {
    case '2':
      MAX_VELOCITY += 1;
      console.log('max velocity' +
MAX_VELOCITY);
        break;
  case '1':
      MAX_VELOCITY -= 1;
      console.log('min velocity' +
MAX_VELOCITY);
        break;

                case 'w':
      SEPARATION_STRENGH += 1;
      console.log('increase sep
strength' + MAX_VELOCITY);
        break;
  case 'q':
      SEPARATION_STRENGH -= 1;
      console.log('decrease sep
strength' + MAX_VELOCITY);
        break;
  case 'w':
      ALIGNMENT_STRENGH += 1;
      console.log('increase alignment
strength' + MAX_VELOCITY);
        break;
  case 'q':
      ALIGNMENT_STRENGH -= 1;
      console.log('decrease alignment
strength' + MAX_VELOCITY);
        break;

                case 's':
      COHESION_STRENGH += 1;
      console.log('increase cohesion
strength' + MAX_VELOCITY);
        break;
  case 'a':
      COHESION_STRENGH -= 1;
      console.log('decrease cohesion
strength' + MAX_VELOCITY);
        break;

                case 'x':

BOUNDARY_SEPARATION_STRE
NGTH += 1;
      console.log('increase bss' +
MAX_VELOCITY);
        break;
  case 'z':

BOUNDARY_SEPARATION_STRE
NGTH -= 1;
      console.log('decrease bss' +
MAX_VELOCITY);
        break;


  case 'r':
      console.log('reset!');
      SEPARATION_RANGE = 50.0;
       ALIGNMENT_RANGE = 30.0;
       COHESION_RANGE = 50.0;
       SEPARATION_STRENGH =
30.0;
       ALIGNMENT_STRENGH =
0.01;
       COHESION_STRENGH = 0.1;


BOUNDARY_SEPARATION_RANG
E = 50.0;

BOUNDARY_SEPARATION_STRE
NGTH = 50.0;

       MAX_VELOCITY = 5.0;
    default:
        console.log('default - no key
pressed');
}
});

function setup() {
  createCanvas(448, 761);

  var myImage = new Image(448,
761);
  myImage.src =
'https://s6.postimg.cc/d0s0fev3l/Thes
is_pic3cropped.png';
  document.body.style.background =
"url('https://s6.postimg.cc/d0s0fev3l/
Thesis_pic3cropped.png')
no-repeat";
  initialize();
}

function initialize() {
  createBoundaryBucket();
  positions = new Array(N);
  velocities = new Array(N);
  forces = new Array(N);
  for (var i = 0; i < N; i++) {
    positions[i] = getInitialPosition();
    velocities[i] = randomInCircle();
    forces[i] = createVector(0, 0);
  }

  maxRange =
max([SEPARATION_RANGE,
ALIGNMENT_RANGE,
COHESION_RANGE]);
  createGridBucket();
}

function getInitialPosition() {
  while(true) {
    var p =
createVector(random(width),
random(height));
    var isAvailable =
boundaries.every(function(boundary)
{
      if (boundary.isCircleBoundary) {
        return p.dist(boundary.center)
>= boundary.radius;
      } else {
        return true;
      }
    });
    if (isAvailable) return p;
  }
}
```

```
function createGridBucket() {
  var x = ceil(width / maxRange) + 2;
  var y = ceil(height / maxRange) + 2;
  gridBucket = new Array(x);
  for (var xi = 0; xi < x; xi++) {
    gridBucket[xi] = new Array(y);
    for (var yi = 0; yi < y; yi++) {
      gridBucket[xi][yi] = [];
    }
  }
}

function createBoundaryBucket() {
  boundaries = [
    { // left wall
      calcDistance: function(p) {return
p.x;},
      calcNormal: function(p) {return
createVector(1.0, 0.0);}
    }, { // right wall
      calcDistance: function(p) {return
max(width - p.x, 0.0);},
      calcNormal: function(p) {return
createVector(-1.0, 0.0);}
    }, { // top wall
      calcDistance: function(p) {return
p.y;},
      calcNormal: function(p) {return
createVector(0.0, 1.0);}
    }, { // bottom wall
      calcDistance: function(p) {return
max(height - p.y, 0.0);},
      calcNormal: function(p) {return
createVector(0.0, -1.0);}
    } ,
                        new
CircleBoundary(createVector(120,
117), 65), //big three
                        new
CircleBoundary(createVector(120,
347), 65),
                        new
CircleBoundary(createVector(120,
577), 65),

                        new
CircleBoundary(createVector(291,
117), 30),//top
                        new
CircleBoundary(createVector(382,
117), 30),

                        new
CircleBoundary(createVector(385,
352), 30), //middle
                        new
CircleBoundary(createVector(295,
352), 30),

                        new
CircleBoundary(createVector(385,
577), 30), //bottom
                        new
CircleBoundary(createVector(295,
577), 30)

  ];

  var x = ceil(width /
BOUNDARY_SEPARATION_RANG
E) + 2;
  var y = ceil(height /
BOUNDARY_SEPARATION_RANG
E) + 2;
  boundaryBucket = new Array(x);
  for (var xi = 0; xi < x; xi++) {
    boundaryBucket[xi] = new
Array(y);
    for (var yi = 0; yi < y; yi++) {
      boundaryBucket[xi][yi] = [];
      if (xi === 0 || xi === x - 1 || yi ===
0 || yi === y - 1) continue;

boundaries.forEach(function(boundar
y) {
        var d = min([

boundary.calcDistance(getBoundary
BucketPosition({x: xi, y: yi})),

boundary.calcDistance(getBoundary
BucketPosition({x: xi + 1, y: yi})),

boundary.calcDistance(getBoundary
BucketPosition({x: xi, y: yi + 1})),

boundary.calcDistance(getBoundary
BucketPosition({x: xi + 1, y: yi + 1}))
        ]);
        if (d <
BOUNDARY_SEPARATION_RANG
E) {

boundaryBucket[xi][yi].push(boundar
y);
        }
      });
    }
  }
}

function
getBoundaryBucketPosition(idx) {
  return createVector(
    (idx.x - 1) *
BOUNDARY_SEPARATION_RANG
E,
    (idx.y - 1) *
BOUNDARY_SEPARATION_RANG
E
  );
}

function randomInCircle() {
  var r = random();
  var a = random(TWO_PI);
  return createVector(r * cos(a), r *
sin(a));
}

function draw() {
  clear();
  render();
  update();
}

function render() {

  fill(255);
  for (var i = 0; i < N; i++) {
    var pos = positions[i];
    fill(200,200,200,240);
                        noStroke();
    ellipse(pos.x, pos.y, 7, 7);
  }
  noFill();

boundaries.forEach(function(boundar
y) {
    if (boundary.isCircleBoundary) {
      ellipse(boundary.center.x,
boundary.center.y, boundary.radius *
2.0, boundary.radius * 2.0);
    }
  });
}

function update() {
  updateGridBucket();
  calcSeparation();
  calcAlignment();
  calcCohesion();
  calcBoundarySeparation();
  move();
  resolveBoundary();
}

function updateGridBucket() {
```

```
  clearGridBucket();
  for (var i = 0; i < N; i++) {
    var bucket = getBucketIdx(i);
gridBucket[bucket.x][bucket.y].push(i
);
  }
}

function clearGridBucket() {
  for (var xi = 0; xi <
gridBucket.length; xi++) {
    for (var yi = 0; yi <
gridBucket[xi].length; yi++) {
      gridBucket[xi][yi] = [];
    }
  }
}

function getBucketIdx(idx) {
  return {
    x: floor(positions[idx].x /
maxRange) + 1,
    y: floor(positions[idx].y /
maxRange) + 1
  }
}

function getBoundaryBucketIdx(idx) {
  return {
    x: floor(positions[idx].x /
BOUNDARY_SEPARATION_RANG
E) + 1,
    y: floor(positions[idx].y /
BOUNDARY_SEPARATION_RANG
E) + 1
  }
}

function weight(dist, maxDist) {
  return dist < maxDist ? 1.0 - (dist /
maxDist) : 0.0;
}

function calcSeparation() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var force = createVector(0.0, 0.0);
    var bucket = getBucketIdx(i);
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {
gridBucket[xi][yi].forEach(function(j) {
          if (i === j) return;
          var posj = positions[j];
          var d = posi.dist(posj);
          var w = weight(d,
SEPARATION_RANGE);
          if (w <= 0.0) return;
          force.add(p5.Vector.sub(posi,
posj).normalize().mult(w));
        });
      }
    }
forces[i].add(force.mult(SEPARATIO
N_STRENGH));
  }
}

function calcAlignment() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var sumW = 0;
    var sumVel = createVector(0.0,
0.0);
    var bucket = getBucketIdx(i);
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {
gridBucket[xi][yi].forEach(function(j) {
          if (i === j) return;
          var posj = positions[j];
          var d = posi.dist(posj);
          var w = weight(d,
ALIGNMENT_RANGE);
          if (w <= 0.0) return;
          sumW = w;
sumVel.add(velocities[j].copy().mult(
w));
        });
      }
    }
  if (sumW < 0.001) continue;
  sumVel.div(sumW);
  var force = p5.Vector.sub(sumVel,
velocities[i]);
forces[i].add(force.mult(ALIGNMENT
_STRENGH));
  }
}

function calcCohesion() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var force = createVector(0.0, 0.0);
    var bucket = getBucketIdx(i);
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {
gridBucket[xi][yi].forEach(function(j) {
          if (i === j) return;
          var posj = positions[j];
          var d = posi.dist(posj);
          var w = weight(d,
COHESION_RANGE);
          if (w <= 0.0) return;
          force.add(p5.Vector.sub(posj,
posi).normalize().mult(w));
        });
      }
    }
forces[i].add(force.mult(COHESION_
STRENGH));
  }
}

function calcBoundarySeparation() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var force = createVector(0.0, 0.0);
    var bucket =
getBoundaryBucketIdx(i);
    var checkedBoundaries = [];
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {
boundaryBucket[xi][yi].forEach(functi
on(boundary) {
          if
(checkedBoundaries.some(function(c
hecked) {checked === boundary;}))
return;
checkedBoundaries.push(boundary);
          var d =
boundary.calcDistance(posi);
          var w = weight(d,
BOUNDARY_SEPARATION_RANG
E);
          if (w <= 0.0) return;
force.add(boundary.calcNormal(posi)
.mult(w));
        });
      }
    }
```

```
forces[i].add(force.mult(BOUNDARY
_SEPARATION_STRENGTH));
  }
}

function move() {
  var currentMillis = millis();
  var dt = min((currentMillis -
prevMillis) * 0.0001, 0.02);
  for (var i = 0; i < N; i++) {

velocities[i].add(p5.Vector.mult(force
s[i], dt));

velocities[i].limit(MAX_VELOCITY);

positions[i].add(p5.Vector.mult(veloci
ties[i], dt));
    forces[i].set(0.0, 0.0);
  }
  prevMillis = currentMillis;
}

function resolveBoundary() {
  for (var i = 0; i < N; i++) {
    var pos = positions[i];
    var vel = velocities[i];
    if (pos.x < 0.0) {
      vel.x *= -1;
      pos.x = 0.0;
    }
    if (pos.x >= width) {
      vel.x *= -1;
      pos.x = width - 0.1;
    }
    if (pos.y < 0.0) {
      vel.y *= -1;
      pos.y = 0.0;
    }
    if (pos.y >= height) {
      vel.y *= -1;
      pos.y = height - 0.1;
    }
  }
}

function CircleBoundary(center,
radius) {
  this.center = center;
  this.radius = radius;
  this.isCircleBoundary = true;
}

CircleBoundary.prototype.calcDistan
ce = function(p) {
  return max(p.dist(this.center) -
this.radius, 0.0);
}

CircleBoundary.prototype.calcNorma
l = function(p) {
  return p5.Vector.sub(p,
this.center).normalize();
}
```

12.1.4 Code Large/Slow Boids

```
var SEPARATION_RANGE = 60.0;
var ALIGNMENT_RANGE = 60.0;
var COHESION_RANGE = 60.0;
var SEPARATION_STRENGH =
60.0;
var ALIGNMENT_STRENGH = 1.3;
var COHESION_STRENGH = 1.0;

var
BOUNDARY_SEPARATION_RANG
E = 30.0;
var
BOUNDARY_SEPARATION_STRE
NGTH = 140.0;

var MAX_VELOCITY = 50.0;

var N = 500;
var positions;
var velocities;
var forces;

var gridBucket;
var boundaryBucket;
var boundaries;
var maxRange;
var prevMillis = 0.0;


document.addEventListener('keydow
n', function(event){
  switch(event.key) {
    case '2':
      MAX_VELOCITY += 1;
      console.log('max velocity' +
MAX_VELOCITY);
        break;
  case '1':
      MAX_VELOCITY -= 1;
      console.log('min velocity' +
MAX_VELOCITY);
        break;

            case 'w':
      SEPARATION_STRENGH += 1;
      console.log('increase sep
strength' + MAX_VELOCITY);
        break;
  case 'q':
      SEPARATION_STRENGH -= 1;
      console.log('decrease sep
strength' + MAX_VELOCITY);
        break;
  case 'w':
      ALIGNMENT_STRENGH += 1;
      console.log('increase alignment
strength' + MAX_VELOCITY);
        break;
  case 'q':
      ALIGNMENT_STRENGH -= 1;
      console.log('decrease alignment
strength' + MAX_VELOCITY);
        break;

            case 's':
      COHESION_STRENGH += 1;
      console.log('increase cohesion
strength' + MAX_VELOCITY);
        break;
  case 'a':
      COHESION_STRENGH -= 1;
      console.log('decrease cohesion
strength' + MAX_VELOCITY);
        break;

            case 'x':

BOUNDARY_SEPARATION_STRE
NGTH += 1;
      console.log('increase bss' +
MAX_VELOCITY);
        break;
  case 'z':

BOUNDARY_SEPARATION_STRE
NGTH -= 1;
      console.log('decrease bss' +
MAX_VELOCITY);
        break;


  case 'r':
      console.log('reset!');
      SEPARATION_RANGE = 50.0;
       ALIGNMENT_RANGE = 30.0;
       COHESION_RANGE = 50.0;
       SEPARATION_STRENGH =
30.0;
       ALIGNMENT_STRENGH =
0.01;
       COHESION_STRENGH = 0.1;


BOUNDARY_SEPARATION_RANG
E = 50.0;

BOUNDARY_SEPARATION_STRE
NGTH = 50.0;
       MAX_VELOCITY = 5.0;
    default:
        console.log('default - no key
pressed');
}
});

function setup() {
  createCanvas(448, 761);

  var myImage = new Image(448,
761);
  myImage.src =
'https://s6.postimg.cc/d0s0fev3l/Thes
is_pic3cropped.png';
  document.body.style.background =
"url('https://s6.postimg.cc/d0s0fev3l/
Thesis_pic3cropped.png')
no-repeat";
  initialize();
}

function initialize() {
  createBoundaryBucket();
  positions = new Array(N);
  velocities = new Array(N);
  forces = new Array(N);
  for (var i = 0; i < N; i++) {
    positions[i] = getInitialPosition();
    velocities[i] = randomInCircle();
    forces[i] = createVector(0, 0);
  }

  maxRange =
max([SEPARATION_RANGE,
ALIGNMENT_RANGE,
COHESION_RANGE]);
  createGridBucket();
}

function getInitialPosition() {
  while(true) {
    var p =
createVector(random(width),
random(height));
    var isAvailable =
boundaries.every(function(boundary)
{
      if (boundary.isCircleBoundary) {
        return p.dist(boundary.center)
>= boundary.radius;
      } else {
        return true;
      }
    });
    if (isAvailable) return p;
  }
```

```
}

function createGridBucket() {
  var x = ceil(width / maxRange) + 2;
  var y = ceil(height / maxRange) + 2;
  gridBucket = new Array(x);
  for (var xi = 0; xi < x; xi++) {
    gridBucket[xi] = new Array(y);
    for (var yi = 0; yi < y; yi++) {
      gridBucket[xi][yi] = [];
    }
  }
}

function createBoundaryBucket() {
  boundaries = [
    { // left wall
      calcDistance: function(p) {return
p.x;},
      calcNormal: function(p) {return
createVector(1.0, 0.0);}
    }, { // right wall
      calcDistance: function(p) {return
max(width - p.x, 0.0);},
      calcNormal: function(p) {return
createVector(-1.0, 0.0);}
    }, { // top wall
      calcDistance: function(p) {return
p.y;},
      calcNormal: function(p) {return
createVector(0.0, 1.0);}
    }, { // bottom wall
      calcDistance: function(p) {return
max(height - p.y, 0.0);},
      calcNormal: function(p) {return
createVector(0.0, -1.0);}
    } ,
                        new
CircleBoundary(createVector(120,
117), 65), //big three
                        new
CircleBoundary(createVector(120,
347), 65),
                        new
CircleBoundary(createVector(120,
577), 65),

                        new
CircleBoundary(createVector(291,
117), 30),//top
                        new
CircleBoundary(createVector(382,
117), 30),

                        new
CircleBoundary(createVector(385,
352), 30), //middle
                        new
CircleBoundary(createVector(295,
352), 30),

                        new
CircleBoundary(createVector(385,
577), 30), //bottom
                        new
CircleBoundary(createVector(295,
577), 30)

  ];

  var x = ceil(width /
BOUNDARY_SEPARATION_RANG
E) + 2;
  var y = ceil(height /
BOUNDARY_SEPARATION_RANG
E) + 2;
  boundaryBucket = new Array(x);
  for (var xi = 0; xi < x; xi++) {
    boundaryBucket[xi] = new
Array(y);
    for (var yi = 0; yi < y; yi++) {
      boundaryBucket[xi][yi] = [];
      if (xi === 0 || xi === x - 1 || yi ===
0 || yi === y - 1) continue;

boundaries.forEach(function(boundar
y) {
        var d = min([

boundary.calcDistance(getBoundary
BucketPosition({x: xi, y: yi})),

boundary.calcDistance(getBoundary
BucketPosition({x: xi + 1, y: yi})),

boundary.calcDistance(getBoundary
BucketPosition({x: xi, y: yi + 1})),

boundary.calcDistance(getBoundary
BucketPosition({x: xi + 1, y: yi + 1}))
        ]);
        if (d <
BOUNDARY_SEPARATION_RANG
E) {

boundaryBucket[xi][yi].push(boundar
y);
        }
      });
    }
  }
}

function
getBoundaryBucketPosition(idx) {
  return createVector(
    (idx.x - 1) *
BOUNDARY_SEPARATION_RANG
E,
    (idx.y - 1) *
BOUNDARY_SEPARATION_RANG
E
  );
}

function randomInCircle() {
  var r = random();
  var a = random(TWO_PI);
  return createVector(r * cos(a), r *
sin(a));
}

function draw() {
  clear();
  render();
  update();
}

function render() {

  fill(255);
  for (var i = 0; i < N; i++) {
    var pos = positions[i];
    fill(200,200,200,240);
                        noStroke();
    ellipse(pos.x, pos.y, 15, 15);
  }
  noFill();

boundaries.forEach(function(boundar
y) {
    if (boundary.isCircleBoundary) {
      ellipse(boundary.center.x,
boundary.center.y, boundary.radius *
2.0, boundary.radius * 2.0);
    }
  });
}

function update() {
  updateGridBucket();
  calcSeparation();
  calcAlignment();
  calcCohesion();
  calcBoundarySeparation();
  move();
  resolveBoundary();
}

function updateGridBucket() {
```

```
  clearGridBucket();
  for (var i = 0; i < N; i++) {
    var bucket = getBucketIdx(i);

gridBucket[bucket.x][bucket.y].push(i
);
  }
}

function clearGridBucket() {
  for (var xi = 0; xi <
gridBucket.length; xi++) {
    for (var yi = 0; yi <
gridBucket[xi].length; yi++) {
      gridBucket[xi][yi] = [];
    }
  }
}

function getBucketIdx(idx) {
  return {
    x: floor(positions[idx].x /
maxRange) + 1,
    y: floor(positions[idx].y /
maxRange) + 1
  }
}

function getBoundaryBucketIdx(idx) {
  return {
    x: floor(positions[idx].x /
BOUNDARY_SEPARATION_RANG
E) + 1,
    y: floor(positions[idx].y /
BOUNDARY_SEPARATION_RANG
E) + 1
  }
}

function weight(dist, maxDist) {
  return dist < maxDist ? 1.0 - (dist /
maxDist) : 0.0;
}

function calcSeparation() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var force = createVector(0.0, 0.0);
    var bucket = getBucketIdx(i);
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {

gridBucket[xi][yi].forEach(function(j) {
          if (i === j) return;
          var posj = positions[j];
          var d = posi.dist(posj);
          var w = weight(d,
SEPARATION_RANGE);
          if (w <= 0.0) return;
          force.add(p5.Vector.sub(posi,
posj).normalize().mult(w));
        });
      }
    }

forces[i].add(force.mult(SEPARATIO
N_STRENGH));
  }
}

function calcAlignment() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var sumW = 0;
    var sumVel = createVector(0.0,
0.0);
    var bucket = getBucketIdx(i);
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {

gridBucket[xi][yi].forEach(function(j) {
          if (i === j) return;
          var posj = positions[j];
          var d = posi.dist(posj);
          var w = weight(d,
ALIGNMENT_RANGE);
          if (w <= 0.0) return;
          sumW = w;

sumVel.add(velocities[j].copy().mult(
w));
        });
      }
    }
  if (sumW < 0.001) continue;
  sumVel.div(sumW);
  var force = p5.Vector.sub(sumVel,
velocities[i]);

forces[i].add(force.mult(ALIGNMENT
_STRENGH));
  }
}

function calcCohesion() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var force = createVector(0.0, 0.0);
    var bucket = getBucketIdx(i);
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {

gridBucket[xi][yi].forEach(function(j) {
          if (i === j) return;
          var posj = positions[j];
          var d = posi.dist(posj);
          var w = weight(d,
COHESION_RANGE);
          if (w <= 0.0) return;
          force.add(p5.Vector.sub(posj,
posi).normalize().mult(w));
        });
      }
    }

forces[i].add(force.mult(COHESION_
STRENGH));
  }
}

function calcBoundarySeparation() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var force = createVector(0.0, 0.0);
    var bucket =
getBoundaryBucketIdx(i);
    var checkedBoundaries = [];
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {

boundaryBucket[xi][yi].forEach(functi
on(boundary) {
          if
(checkedBoundaries.some(function(c
hecked) {checked === boundary;}))
return;

checkedBoundaries.push(boundary);
          var d =
boundary.calcDistance(posi);
          var w = weight(d,
BOUNDARY_SEPARATION_RANG
E);
          if (w <= 0.0) return;

force.add(boundary.calcNormal(posi)
.mult(w));
        });
      }
    }
```

```
forces[i].add(force.mult(BOUNDARY
_SEPARATION_STRENGTH));
  }
}

function move() {
  var currentMillis = millis();
  var dt = min((currentMillis -
prevMillis) * 0.0001, 0.02);
  for (var i = 0; i < N; i++) {

velocities[i].add(p5.Vector.mult(force
s[i], dt));

velocities[i].limit(MAX_VELOCITY);

positions[i].add(p5.Vector.mult(veloci
ties[i], dt));
    forces[i].set(0.0, 0.0);
  }
  prevMillis = currentMillis;
}

function resolveBoundary() {
  for (var i = 0; i < N; i++) {
    var pos = positions[i];
    var vel = velocities[i];
    if (pos.x < 0.0) {
      vel.x *= -1;
      pos.x = 0.0;
    }
    if (pos.x >= width) {
      vel.x *= -1;
      pos.x = width - 0.1;
    }
    if (pos.y < 0.0) {
      vel.y *= -1;
      pos.y = 0.0;
    }
    if (pos.y >= height) {
      vel.y *= -1;
      pos.y = height - 0.1;
    }
  }
}

function CircleBoundary(center,
radius) {
  this.center = center;
  this.radius = radius;
  this.isCircleBoundary = true;
}

CircleBoundary.prototype.calcDistan
ce = function(p) {
  return max(p.dist(this.center) -
this.radius, 0.0);
}

CircleBoundary.prototype.calcNorma
l = function(p) {
  return p5.Vector.sub(p,
this.center).normalize();
}
```

12.1.5 Code Large/Fast Boids

```
var SEPARATION_RANGE = 20.0;
var ALIGNMENT_RANGE = 20.0;
var COHESION_RANGE = 60.0;
var SEPARATION_STRENGH =
60.0;
var ALIGNMENT_STRENGH = 0.15;
var COHESION_STRENGH = 0.10;

var
BOUNDARY_SEPARATION_RANG
E = 100.0;
var
BOUNDARY_SEPARATION_STRE
NGTH = 220.0;

var MAX_VELOCITY = 200.0;

var N = 500;
var positions;
var velocities;
var forces;

var gridBucket;
var boundaryBucket;
var boundaries;
var maxRange;
var prevMillis = 0.0;


document.addEventListener('keydow
n', function(event){
  switch(event.key) {
    case '2':
      MAX_VELOCITY += 1;
      console.log('max velocity' +
MAX_VELOCITY);
        break;
  case '1':
      MAX_VELOCITY -= 1;
      console.log('min velocity' +
MAX_VELOCITY);
        break;

             case 'w':
      SEPARATION_STRENGH += 1;
      console.log('increase sep
strength' + MAX_VELOCITY);
        break;
  case 'q':
      SEPARATION_STRENGH -= 1;
      console.log('decrease sep
strength' + MAX_VELOCITY);
        break;
  case 'w':
      ALIGNMENT_STRENGH += 1;
      console.log('increase alignment
strength' + MAX_VELOCITY);
        break;
  case 'q':
      ALIGNMENT_STRENGH -= 1;
      console.log('decrease alignment
strength' + MAX_VELOCITY);
        break;

             case 's':
      COHESION_STRENGH += 1;
      console.log('increase cohesion
strength' + MAX_VELOCITY);
        break;
  case 'a':
      COHESION_STRENGH -= 1;
      console.log('decrease cohesion
strength' + MAX_VELOCITY);
        break;

             case 'x':

BOUNDARY_SEPARATION_STRE
NGTH += 1;
      console.log('increase bss' +
MAX_VELOCITY);
        break;
  case 'z':

BOUNDARY_SEPARATION_STRE
NGTH -= 1;
      console.log('decrease bss' +
MAX_VELOCITY);
        break;


  case 'r':
      console.log('reset!');
      SEPARATION_RANGE = 50.0;
       ALIGNMENT_RANGE = 30.0;
       COHESION_RANGE = 50.0;
       SEPARATION_STRENGH =
30.0;
       ALIGNMENT_STRENGH =
0.01;
       COHESION_STRENGH = 0.1;


BOUNDARY_SEPARATION_RANG
E = 50.0;

BOUNDARY_SEPARATION_STRE
NGTH = 50.0;
       MAX_VELOCITY = 5.0;
    default:
       console.log('default - no key
pressed');
}
});

function setup() {
  createCanvas(448, 761);

  var myImage = new Image(448,
761);
  myImage.src =
'https://s6.postimg.cc/d0s0fev3l/Thes
is_pic3cropped.png';
  document.body.style.background =
"url('https://s6.postimg.cc/d0s0fev3l/
Thesis_pic3cropped.png')
no-repeat";
  initialize();
}

function initialize() {
  createBoundaryBucket();
  positions = new Array(N);
  velocities = new Array(N);
  forces = new Array(N);
  for (var i = 0; i < N; i++) {
    positions[i] = getInitialPosition();
    velocities[i] = randomInCircle();
    forces[i] = createVector(0, 0);
  }

  maxRange =
max([SEPARATION_RANGE,
ALIGNMENT_RANGE,
COHESION_RANGE]);
  createGridBucket();
}

function getInitialPosition() {
  while(true) {
    var p =
createVector(random(width),
random(height));
    var isAvailable =
boundaries.every(function(boundary)
{
      if (boundary.isCircleBoundary) {
        return p.dist(boundary.center)
>= boundary.radius;
      } else {
        return true;
      }
    });
    if (isAvailable) return p;
  }
```

```
}

function createGridBucket() {
  var x = ceil(width / maxRange) + 2;
  var y = ceil(height / maxRange) + 2;
  gridBucket = new Array(x);
  for (var xi = 0; xi < x; xi++) {
    gridBucket[xi] = new Array(y);
    for (var yi = 0; yi < y; yi++) {
      gridBucket[xi][yi] = [];
    }
  }
}

function createBoundaryBucket() {
  boundaries = [
    { // left wall
      calcDistance: function(p) {return
p.x;},
      calcNormal: function(p) {return
createVector(1.0, 0.0);}
    }, { // right wall
      calcDistance: function(p) {return
max(width - p.x, 0.0);},
      calcNormal: function(p) {return
createVector(-1.0, 0.0);}
    }, { // top wall
      calcDistance: function(p) {return
p.y;},
      calcNormal: function(p) {return
createVector(0.0, 1.0);}
    }, { // bottom wall
      calcDistance: function(p) {return
max(height - p.y, 0.0);},
      calcNormal: function(p) {return
createVector(0.0, -1.0);}
    } ,
                         new
CircleBoundary(createVector(120,
117), 65), //big three
                         new
CircleBoundary(createVector(120,
347), 65),
                         new
CircleBoundary(createVector(120,
577), 65),

                         new
CircleBoundary(createVector(291,
117), 30),//top
                         new
CircleBoundary(createVector(382,
117), 30),

                         new
CircleBoundary(createVector(385,
352), 30), //middle
                         new
CircleBoundary(createVector(295,
352), 30),

                         new
CircleBoundary(createVector(385,
577), 30), //bottom
                         new
CircleBoundary(createVector(295,
577), 30)

  ];

  var x = ceil(width /
BOUNDARY_SEPARATION_RANG
E) + 2;
  var y = ceil(height /
BOUNDARY_SEPARATION_RANG
E) + 2;
  boundaryBucket = new Array(x);
  for (var xi = 0; xi < x; xi++) {
    boundaryBucket[xi] = new
Array(y);
    for (var yi = 0; yi < y; yi++) {
      boundaryBucket[xi][yi] = [];
      if (xi === 0 || xi === x - 1 || yi ===
0 || yi === y - 1) continue;

boundaries.forEach(function(boundar
y) {
        var d = min([

boundary.calcDistance(getBoundary
BucketPosition({x: xi, y: yi})),

boundary.calcDistance(getBoundary
BucketPosition({x: xi + 1, y: yi})),

boundary.calcDistance(getBoundary
BucketPosition({x: xi, y: yi + 1})),

boundary.calcDistance(getBoundary
BucketPosition({x: xi + 1, y: yi + 1}))
        ]);
        if (d <
BOUNDARY_SEPARATION_RANG
E) {

boundaryBucket[xi][yi].push(boundar
y);
        }
      });
    }
  }
}

function
getBoundaryBucketPosition(idx) {
  return createVector(
    (idx.x - 1) *
BOUNDARY_SEPARATION_RANG
E,
    (idx.y - 1) *
BOUNDARY_SEPARATION_RANG
E
  );
}

function randomInCircle() {
  var r = random();
  var a = random(TWO_PI);
  return createVector(r * cos(a), r *
sin(a));
}

function draw() {
  clear();
  render();
  update();
}

function render() {

  fill(255);
  for (var i = 0; i < N; i++) {
    var pos = positions[i];
    fill(200,200,200,240);
                      noStroke();
    ellipse(pos.x, pos.y, 15, 15);
  }
  noFill();

boundaries.forEach(function(boundar
y) {
    if (boundary.isCircleBoundary) {
      ellipse(boundary.center.x,
boundary.center.y, boundary.radius *
2.0, boundary.radius * 2.0);
    }
  });
}

function update() {
  updateGridBucket();
  calcSeparation();
  calcAlignment();
  calcCohesion();
  calcBoundarySeparation();
  move();
  resolveBoundary();
}

function updateGridBucket() {
```

```
  clearGridBucket();
  for (var i = 0; i < N; i++) {
    var bucket = getBucketIdx(i);

gridBucket[bucket.x][bucket.y].push(i
);
  }
}

function clearGridBucket() {
  for (var xi = 0; xi <
gridBucket.length; xi++) {
    for (var yi = 0; yi <
gridBucket[xi].length; yi++) {
      gridBucket[xi][yi] = [];
    }
  }
}

function getBucketIdx(idx) {
  return {
    x: floor(positions[idx].x /
maxRange) + 1,
    y: floor(positions[idx].y /
maxRange) + 1
  }
}

function getBoundaryBucketIdx(idx) {
  return {
    x: floor(positions[idx].x /
BOUNDARY_SEPARATION_RANG
E) + 1,
    y: floor(positions[idx].y /
BOUNDARY_SEPARATION_RANG
E) + 1
  }
}

function weight(dist, maxDist) {
  return dist < maxDist ? 1.0 - (dist /
maxDist) : 0.0;
}

function calcSeparation() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var force = createVector(0.0, 0.0);
    var bucket = getBucketIdx(i);
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {

gridBucket[xi][yi].forEach(function(j) {
          if (i === j) return;
          var posj = positions[j];
          var d = posi.dist(posj);
          var w = weight(d,
SEPARATION_RANGE);
          if (w <= 0.0) return;
          force.add(p5.Vector.sub(posi,
posj).normalize().mult(w));
        });
      }
    }

forces[i].add(force.mult(SEPARATIO
N_STRENGH));
  }
}

function calcAlignment() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var sumW = 0;
    var sumVel = createVector(0.0,
0.0);
    var bucket = getBucketIdx(i);
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {

gridBucket[xi][yi].forEach(function(j) {
          if (i === j) return;
          var posj = positions[j];
          var d = posi.dist(posj);
          var w = weight(d,
ALIGNMENT_RANGE);
          if (w <= 0.0) return;
          sumW = w;

sumVel.add(velocities[j].copy().mult(
w));
        });
      }
    }
  if (sumW < 0.001) continue;
  sumVel.div(sumW);
  var force = p5.Vector.sub(sumVel,
velocities[i]);

forces[i].add(force.mult(ALIGNMENT
_STRENGH));
  }
}

function calcCohesion() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var force = createVector(0.0, 0.0);
    var bucket = getBucketIdx(i);
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {

gridBucket[xi][yi].forEach(function(j) {
          if (i === j) return;
          var posj = positions[j];
          var d = posi.dist(posj);
          var w = weight(d,
COHESION_RANGE);
          if (w <= 0.0) return;
          force.add(p5.Vector.sub(posj,
posi).normalize().mult(w));
        });
      }
    }

forces[i].add(force.mult(COHESION_
STRENGH));
  }
}

function calcBoundarySeparation() {
  for (var i = 0; i < N; i++) {
    var posi = positions[i];
    var force = createVector(0.0, 0.0);
    var bucket =
getBoundaryBucketIdx(i);
    var checkedBoundaries = [];
    for (var xi = bucket.x - 1; xi <=
bucket.x + 1; xi++) {
      for (var yi = bucket.y - 1; yi <=
bucket.y + 1; yi++) {

boundaryBucket[xi][yi].forEach(functi
on(boundary) {
          if
(checkedBoundaries.some(function(c
hecked) {checked === boundary;}))
return;

checkedBoundaries.push(boundary);
          var d =
boundary.calcDistance(posi);
          var w = weight(d,
BOUNDARY_SEPARATION_RANG
E);
          if (w <= 0.0) return;

force.add(boundary.calcNormal(posi)
.mult(w));
        });
      }
    }
```

```
forces[i].add(force.mult(BOUNDARY
_SEPARATION_STRENGTH));
  }
}

function move() {
  var currentMillis = millis();
  var dt = min((currentMillis -
prevMillis) * 0.0001, 0.02);
  for (var i = 0; i < N; i++) {

velocities[i].add(p5.Vector.mult(force
s[i], dt));

velocities[i].limit(MAX_VELOCITY);

positions[i].add(p5.Vector.mult(veloci
ties[i], dt));
    forces[i].set(0.0, 0.0);
  }
  prevMillis = currentMillis;
}
```

```
function resolveBoundary() {
  for (var i = 0; i < N; i++) {
    var pos = positions[i];
    var vel = velocities[i];
    if (pos.x < 0.0) {
      vel.x *= -1;
      pos.x = 0.0;
    }
    if (pos.x >= width) {
      vel.x *= -1;
      pos.x = width - 0.1;
    }
    if (pos.y < 0.0) {
      vel.y *= -1;
      pos.y = 0.0;
    }
    if (pos.y >= height) {
      vel.y *= -1;
      pos.y = height - 0.1;
    }
  }
}
```

```
function CircleBoundary(center,
radius) {
  this.center = center;
  this.radius = radius;
  this.isCircleBoundary = true;
}

CircleBoundary.prototype.calcDistan
ce = function(p) {
  return max(p.dist(this.center) -
this.radius, 0.0);
}

CircleBoundary.prototype.calcNorma
l = function(p) {
  return p5.Vector.sub(p,
this.center).normalize();
}
```