Project 2

Human-Computer Interaction

Aalborg University

Ida Taglioni, Janis Ludvigs Berzins-Berzitis, Jonas Litvinas
Malte Juel Breddam, Morten Porsing, Laurynas Lubys

This is where the date is supposed to be
CHANGE THIS BEFORE HANDIN

Contents

1 | Introduction

As the technology becomes more advanced, our expectations of the technology increases as well. Microprocessors have become increasingly faster and smaller and according to Moore's law[1] the improvements will continue. We are now at the point where you can actually fit a computer, which 10 years ago would be considered state of the art, into your pocket and make it affordable for the general population. As the size of the hardware decreases, so does the physical limitation of the devices. There is now room for specialized sensors in the average smartphone, things like accelerometers, magnetometers and gyroscopes along with support for multi touch, all of which allows the user to interact with their device in new and non-traditional ways. Most applications however do not employ these sensors and as such, a lot of users are not even aware of the possibilities they provide. By using these non-traditional interaction methods one can strive to improve the user's experience by challenging them to use their devices in new and exciting ways. When the users have their preconceptions challenged, a state of flow is established and even tedious tasks may seem fun. This report will explore some of these non-traditional interaction methods, specifically when it comes to navigation in a 3D environment. These methods will be explored by creating an environment using Autodesk Maya, Adobe Photoshop and Unity3d where the users will have a chance to try out various means of navigating through the three dimensional world, in an attempt to determine the most efficient and most intuitive way of controls.

1.1 Initial Problem Statement

How can we improve user experience when navigating a 3D space using non-traditional mobile sensors?

[1] Moore's law predicts that the number of transistors will roughly double every two years.

2 | Analysis

2.1 Target Group

Profound understanding of the target group helps in the process of creating a useful concept. It will be an answer to the user's specific needs and wishes. Dividing the target group into different group categorizations, or segments, gives a possibility to give a deeper understanding of the chosen subjects and leads to a better concept design. Created prototypes will be specifically designed for a segmented and generalized group of people of this project. Connecting with the target group is essential since the product will be used by actual users. In most cases it is not enough to generalize a group of people from surface observations or presumed stereotypes, as people sometimes say one thing but do another. Their actual needs can vary a lot.

However since this project is targeting people who might need to use non-traditional interaction in 3D environment on their mobile devices, it can be used in many fields such as video games, simulations and localization, e.g. a simulative approach of viewing architectural objects, a city and landscape areas as in "Google maps". The target group can also be very wide in all perspectives. This means that the target group is not generalizable and therefore it does not require a deeper analysis.

There are many ways to segment a target audience. Probably the most popular is a geographic or demographic segmentation. However it is most relevant to analyse psychographics as geographics and demographics would give too big a range in age, geographical position etc. This kind of data would not be conclusive.

2.1.1 Psychographics

Psychographic generalization segments target groups according to social class, lifestyle and personality characteristics. [Examstutor.com, 2015] It is important and relevant to understand the customer's needs, their habits and personality since it can partially answer how the app's concept can be developed.

Digital knowledge

In Marc Prensky's "Digital Natives, Digital Immigrants" [Prensky, 2001] article he categorizes his understanding of target groups into two segments, when it comes to understanding or learning digital technology. He categorizes them as "digital natives" and "digital immigrants". There are a few more categorizations that people use, such as "Born digital" or "Digital Settlers", so it is common to separate people into "digital knowledge" groups. Digital immigrants are the ones who was born and grew up before the technological revolution, e.g. a 65 year old man who did not have the computers, digital tools or equipment that people do now. This person adopted the technology at a certain age or point in their life when it was needed. Digital natives are the one who grew up in the technological era, where they had access to the internet, computers and probably experienced one or more ways of learning in a digital environment [Prensky, 2001]. However, Prensky notes that time will make everyone a "digital native", as everyone will be born in a world full of advanced technology. Old generalization terminology will not be suiting in the future. He quotes Albert Einstein - "*The problems that exist in the world today cannot be solved by the level of thinking that created them.*" Prensky later introduces "Digital wisdom" that is a more general term but fitting in this era.[Prensky, 2009]. Since the target group is so wide in demographic and geographic aspects that the only grasp would be the physcographic aspects but focused profession and lifestyle. A person's profession and lifestyle can show if a specific person might use non-traditional interaction when it comes to navigating in 3D environment. E.g. a video gamer would most likely use one or more non-traditional approaches while playing mobile video game. Even a simple game on a smartphone might require multitouch or other sensors to control it. Therefore, a person who has a need to use this, needs to have digital wisdom already. Then the focus is not how to create such interaction but rather how to make it efficient, effective, easily learnable etc.

grasp?

How do you know this? REFERENCES! - i added a "might".

2.1.2 Target group conclusion

It can be concluded that the majority of the target group will have digital-wisdom.

2.2 User Experience Strategy

This project is aiming for developing the intuitive user experience. To have a focus on intuitiveness means that we must be able to understand what that implies. The dictionary defines intuitive as:

- perceiving directly by intuition without rational thought, as a person or the mind.

They define the concept of intuitiveness as human perception by intuition, what then is intuition? again the dictionary will provide a relatively easy answer:

- intuition

    1. The act or faculty of perceiving, or apprehending by means of the senses or of the mind; cognition; understanding.
    2. immediate or intuitive recognition or appreciation, as of moral, psychological, or aesthetic qualities; insight; intuition; discernment: an artist of rare perception.
    3. the result or product of perceiving, as distinguished from the act of perceiving; percept.
    4. Psychology. a single unified awareness derived from sensory processes while a stimulus is present.

From this definition it is clear that the concept of intuitiveness is a human concept, more specifically a human subconscious concept. In an article from 1994 Jef Raskin[Raskin, 1994] talks about how intuitiveness comes from familiarity, while the article is quite old, the observations that he makes does support the idea that intuitiveness is directly linked with the targeted users. In the article Raskin talks about an experiment that he performed, where he asks a test participant to perform a certain task with a mouse. Back in 1994 the mouse was still not a tool that was commonplace and as such, the test subject had no familiarity with how to work with a mouse and required help. Raskin showed the participant how to move the mouse in the correct manner and instantly the participant knew how it worked and did not require any more help. As Raskin notes: *"The directional mapping of the mouse was "intuitive" because in this regard it operated just like joysticks (to say nothing of pencils) with which she* [The test participant] *was familiar"*. [Raskin, 1994] This observation strongly supports the idea of intuition as familiarity. With this in mind the goal of this section becomes clear: first this section will give a brief overview of the topic of user experience. Next the section will try to define what the intuitive user experience is, and lastly how does the gained knowledge translate to being used as guidelines for making an intuitive app for a mobile device.

2.2.1 Introduction to user experience

A study of user experience[1] is a study of how a user feels when interacting with a system. The field encompasses a whole range of different and seemingly unrelated topics. The most known part of UX is probably the concept of usability which will be discussed later in the section. Other things make up UX such as: Design, Accessibility, System performance, Ergonomics, human factors and other concepts[Gube, 2010]. The term user experience was originally coined by Dr. Donald Norman, who was the first to describe the importance of user-centred design. User-centred design is a design concept that lets the users dictate(to a certain degree) what the system should contain and what form it should take. Before user-centred design the general design process looked like:

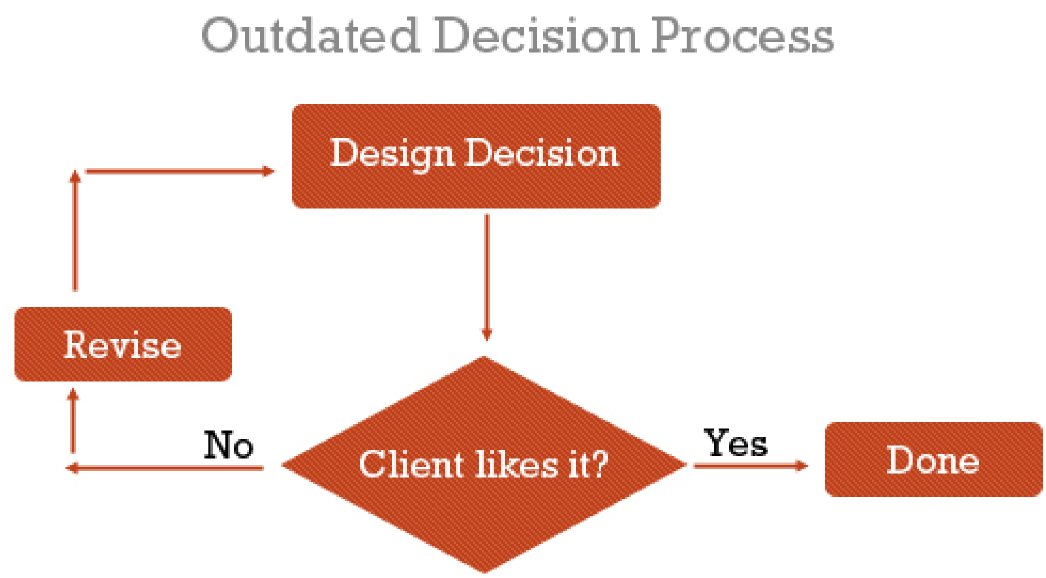

Figure 2.1: old decision process, Jacob Gube 2010

[1]hereafter referred to as UX

Nowhere in the design process was the users a factor, the design was simply made according to how the designers as well as the client felt it should be. Making the same kind of chart for a user-centred approach would look like this:

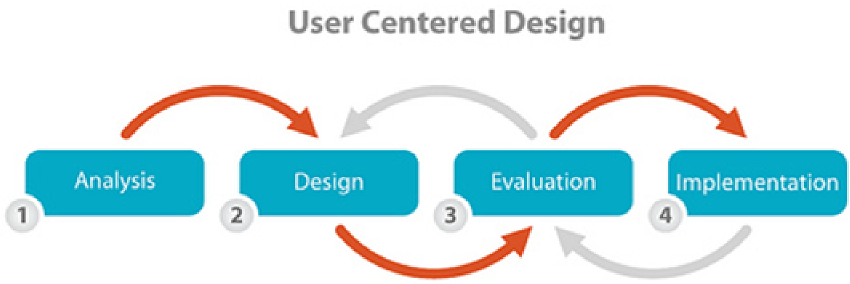

Figure 2.2: a chart of how user-centered design could function, Usabilla 2014

As this chart shows, user-centred design can be an iterative process. The grey arrows represents the user feedback, which shows that the users should be involved in the evaluation of a design.

User Experience and usability is often confused since a large portion of the guidelines for proper usability also applies to giving a good user experience. What sets the user experience apart from usability is that UX deals with the feeling of usage and usability deals with the effectiveness of usage An example of which could be the iBooks app for iPhone and iPad, an application for reading and browsing E-books.

source for the example

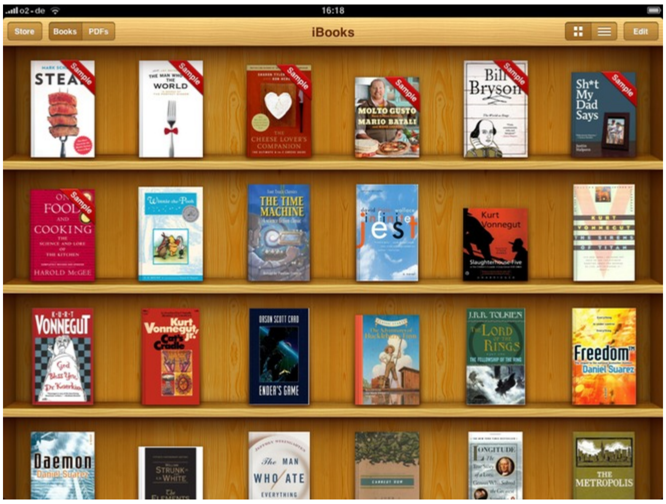

Figure 2.3: Apple iBooks for comparison

might need a source reference

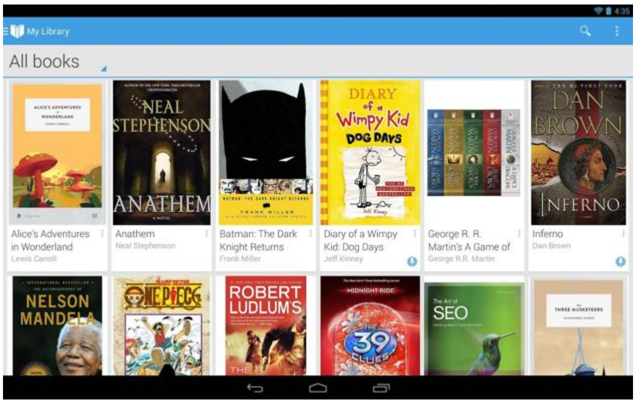

Figure 2.4: Google Play Books for comparison

The layout is simple, it provides an overview of the owned books with a visual representation of the covers which is common for such apps and thus, it does not set itself apart from the state of the art when it comes to usability. However the user experience is greatly improved simply by changing the background to resemble a bookshelf, it makes the experience of logging onto iBooks resemble the experience of going into a book store or library a lot more. This approach relates to the concept of intuition as familiarity, which will be discussed in the next section.

might need a source reference

2.2.2 Intuitiveness is familiarity

As explained in the previous section 2.2.1 user centred design is a main pillar of user experience. This is even more true when talking about intuition as a design concept. As Jared M. Spool mentions in his 2005 article *People Intuit, not Interfaces*[Spool, 2005] the article mentions that it is the users that define whether or not an interface is intuitive, as the interface itself is nothing more than a collection of code. What this shows is that for an interface to be intuitive, a comprehensible knowledge about the targeted users' previous experience with similar interaction, is not only useful but absolutely crucial. The article introduces the concept of a knowledge space, which is the arbitrary space that holds all the knowledge that pertains to a given interface.

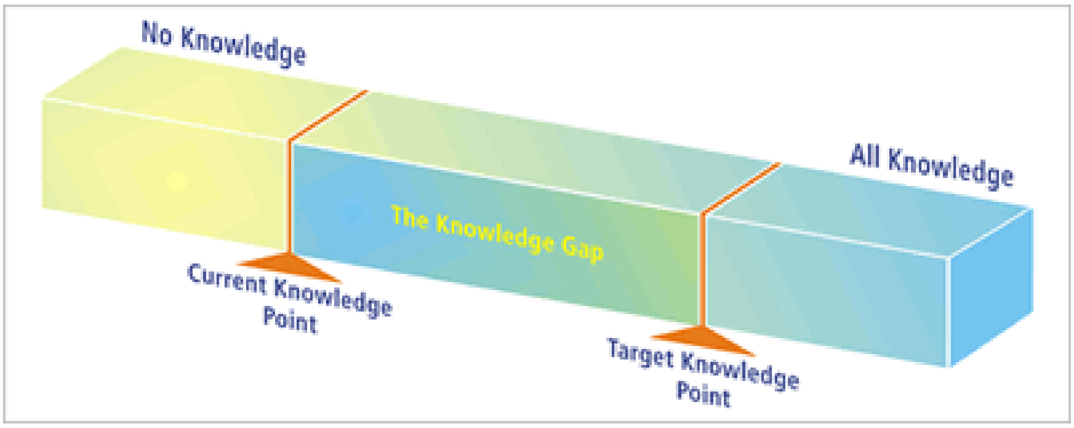

Figure 2.5: The knowledge space viewed as a continuous curve going from *no knowledge* to *all knowledge* [Spool, 2005]

As seen in figure: 2.5 there are two points of interest in the knowledge space that is the *Current Knowledge Point* and the *Target Knowledge Point*. A brief explanation of the two points:

Current Knowledge Point
: This is the expected user knowledge which can be defined by a multitude of ways i.e. user interviews, analysis of similar apps etc. Figuring out what the current knowledge point is will enable the app to fill the knowledge gap without having to guide the user through every tiny detail.

Target Knowledge Point
: This is the amount of knowledge a user needs to be able to use the app/programme as intended.

The Knowledge Gab

The knowledge gab is all the knowledge that the app/programme will have to provide for the user. This is usually done with a series of tutorials.

He puts forth two conditions which he determines are the ones needed before users will classify an interface as being intuitive. These are:

- *Both the current knowledge point and the target knowledge point are identical. When the user walks up to the design, they know everything they need to operate it and complete their objective.*
- *The current knowledge point and the target knowledge point are separate, but the user is completely unaware that the design is helping them bridge the gap. The user is being trained but in a way that seems natural.*

Of these two conditions the latter one will probably be of more use to the project as the navigation with the gyroscope will not be a control scheme that the user necessarily have used before. Since the end product is going to introduce an uncommon way of interacting, it will be important to know which kind of interaction will feel most familiar for the user. This is where the iterative process will enable extensive testing of different interaction models, to determine the correct approach for our users.

Flow through the knowledge gap

As Mihaly Csikszentmihalyi [Csikszentmihalyi, 2005] explains it; for the user to experience a state of flow, the goal and the progress towards it must be clear and well defined as this will be one of the most driving factors when trying to reach this state. The goal should be not too difficult nor too easy but perfect for the user's skill. Just enough to provide the perfect balance between user's skill and the challenges given.[Csikszentmihalyi, 2005]

The real experience begins with when one uses all the concentration on task or challenge at hand making user more and more focused until the user "steps out" of his reality. This is when the state of flow is being reached. [Csikszentmihalyi, 2005] During this state the user may only feel restricted to the rules of the activity instead of the ones in reality. This can also be called a loss of self awareness. During this state the user may loose sense of time or even one's bodily needs if state of flow is very strong.[Csikszentmihalyi, 2005]

To maintain the state of flow receiving immediate feedbacks step by step on how the user is doing is to be established in order to adjust the challenges and skill level necessary to achieve upcoming goals. [Csikszentmihalyi, 2005]

This theory will be used to support and enhance the user's experience when navigating in 3D environment. This will be done by making the challenge given, in this case navigation, challenging for the user yet still balanced in terms of skill that is required to successfully finishing the given task. If successfully implemented, this should establish the state of flow. [Csikszentmihalyi, 2005]

2.2.3 Designing intuitively

The topics discussed in the previous section 2.2.2 helps define what the app has to be able to do but besides these ideas and topics the project will look at the following two structures that can help create a pleasant UX:

- Empowering Users to Complete Tasks Faster
  "When a user has a good experience, one of the first things they say that they liked about it is that it was fast. Since users "equate fast with easy," .[Robinson, 2013] The app that this project will develop does not contain a wide range of features but is a relatively specialized app. While this diminishes the urgency of the app being fast, it should not be neglected. Robinson points to 6 ways of empowering the users effectiveness[Robinson, 2013]:

  1. **Make the app work faster**
     This is a straight forward engineering problem as better/less code results in a faster interface.
  2. **Simplify your users' work flow**
     This means cutting down on the amount of screens that the app employs.
  3. **Make sure your navigation is intuitive**
     As talked about earlier, intuition is related to familiarity and should be able to provide an intuitive navigation within the app.
  4. **Reduce the amount of text**
     In relation to the second point, if an app has a lot of text it will slow down the work flow of the user, at least in the beginning.

5. **Examine your graphics**
   Robinson points to graphics as being an important part of how a user perceives an app, she urges to keep the graphics: *"clean and not distracting"*

6. **Buttons**
   When making any kind of button make sure that the user never questions whether or not it is a button. Furthermore, Robinson also encourages to give the buttons one-word labels such as "send", "buy", "find" etc. Of course the words should represent the action that the button performs.

These points together with the intuitiveness discussion above should enable the app to provide an intuitive user experience.

2.2.4 Usability

According to John Wiley's "Interaction design", there are usability goals that are worth considering when developing interactive and usable systems [Preece et al., 2015] The covered usability goals are:

- Effective to use
  The system does what it suppose to do.

- Efficient to use
  The system supports the user while interacting with it.

- Safe to use

- Good utility
  Provides variety of functionality that the user might need.

- Easy to learn

- Easy to remember how to use

Wiley notes that not all of the goals are needed for most of the systems. Specific systems REFEREN should have specific goals. Probably the most important and relevant goals for this application design is effectiveness and efficiency. In order to help the user navigate in the 3D environment, this system has to have an accurate response which saves time and does tasks quickly. E.g. if the user wants to view 3D objects from a certain angle, it is easy and fast to navigate to the point where the view angle is desired. For the beginners of such interaction with 3D environments, "easy to learn" and "easy to remember" goals would be most relevant. Don Norman states in his "The Design of Everyday Things" that there is a principle of "affordance". [Norman, 1988] The principle is being explained as e.g. A cup is affordable for being picked up, a door is affordable of being opened because of its doorknob, a button affords being pressed etc. So if the user is at a beginner level, the principle of "affordance" is extremely useful. Norman notes that there are two "affordances"; real and perceived. Real is the one in real world as in examples mentioned above and perceived is imitation of reality. E.g. in this design, for beginners especially, navigation buttons needs to look very similar to "real" navigation buttons. For instance, if the user is familiar with any common video gaming console buttons, they have to be represented in a interactive software system with similarity to real one. references!

Don Norman also covers other principles of usability:[Norman, 1988]

- Visibility
  How visible is the button on the screen?
- Feedback
  What kind of feedback does the system gives to the user?
- Constrains
  Visual constraints as e.g. greying out some menu which is not used.
- Mapping
  Layout of menus, buttons etc.
- Consistency
  Consistency through the system in layouts, colours, typography etc.
- Affordance
  Real and perceived affordance, covered above.

Like with usability goals not all of the principles of usability needs to be in one system. However, in this case, the design of the program can include all mentioned principles. E.g. if the 3D navigation system is controlled by the buttons on the screen, they need to be visible, give feedback when pressed, mapped out with consistency rules and visualized with "affordance" principle.

Mobile Usability

Since the design of this system is meant to be on a mobile platform, there are certain aspect to consider. Some users might have limited mobility or problems with manual dexterity - it will cause higher error rates with the interaction [Preece et al., 2015]. Even small aspects like a person's big fingers can cause problems while using a cellphone and especially while interaction needs precision. This problem is mentioned in many articles, reports, researches. In "The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprint"where they introduce possible solutions to this problem. Since the focus of this project is not solving such problems, a deeper analysis will not be done. Some considerations might be done as using Norman's principle of feedback.Buttons pressed on a screen can change the colour or the size. Mobile devices could use some sensors to indicate that the action is achieved, as vibration, blinking flash etc. It is especially useful since most smartphones using on-screen touch buttons and physical sensation of touching the button is non-existing. The user who did not register if he successfully completed the wanted action should get feedback from the mobile device, using its sensors.Feedback is also useful to people who do not put lots of attention to their interaction accuracy; pressing somewhere around button areas and not on it. Feedback would help users to know if a specific action is successful. In general, it is important to consider minorities and helping users effectively understand the efficiency of their actions. Since this non-traditional interaction method can be used by many people which was discovered in analysis of target group,product needs to be optimized to at least fit majority of the users.

REFEREN

REFEREN

REFEREN

Why using its sensors? why not just any form of feedback?

ref to target group

2.2.5 Conclusion

This section has aimed at defining the intuitive user experience which has been defined to be a fulfillment of one of the two conditions mentioned on page 10. The section has also showed that intuitiveness depends on both the users previous knowledge, as well as the design of the interface. Considering Norman's and Wiley's usability goals and principles, it is possible to create a useful application. Usability goals of effectiveness and efficiency will be achieved by using all covered principles. Principles will be used in creation of visual elements and coding.

Having goals as "easy to learn and remember" will help to narrow the knowledge gap when the users first encounter the app/programme. A solution to that is to create interactive navigation using the affordance principle and visual familiarity.

2.3 Interaction Methods in 3d Environment

There are different ways of conveying a 3D environment within the smart device spectrum, in this case smartphones and tablets. It is important that the aim of the project is established before moving on to conclusions on which approach to take. Since the project revolves around navigation in a three dimensional environment, irrelevant approaches will be eliminated right away.

A very important part when developing the application to fit a greatuser experience is to make sure that it works as flawless as possible and that there are no misinterpretations when using the product. For the product that is going to be developed, the "traditional" interaction methods do not cover the functionalities that are needed to cover our initial concept needs. To assure that the alternative interaction is integrated in a convenient manner, knowledge about different sensors and possible combination of two or more to make more intelligent outcomes should be established.

Originally said plea-surable, might need revising

2.3.1 Traditional interaction

As technology evolves, new ways of interacting with computational devices are constantly built. With that, people's needs also change. People adapt to new ways of communication with technology. [Greenfield,] The transition from the classical buttons on a cellphone to a touchscreen has made new ways of interaction possible - the delimitation of physical buttons made it available to have any customized graphical interfaces on the screen possible. This made life easier for casual tasks like zooming a photo using two fingers as multi touch input. This allows registration of multiple points of contact simultaneously, which is much more intuitive than the classical button alternative. Soon enough non-traditional sensors started finding their place in smartphones. The implementation of these sensors in the smartphone allowed new forms of interaction, such as video calling, flashlight and screen orientation, followed by more interesting unusual uses. Application developers started making instrument tuners (GStrings), barcode scanners (Barcode Scanner), radiation detectors (GammaPix), pulse detectors (Instant

Heart Rate), light intensity meters (Light Meter) and countless other applications that uses the sensors to their favour in a non-traditional manner. However, we will only focus on sensors that support our problem area, which points to the ones that can work with 3D environments. The relevant ones for this project are the gyroscope, accelerometer and magnetometer.

2.3.2 Sensors

As it has been established so far there are three main sensors that can support orientation in 3D environments possible. These sensors are accelerometer, gyroscope and magnetometer. In the following section it will be determined whether these sensors are in fact useful to this project's case and if so, to what extent.

Gyroscope, Accelerometer and Magnetometer

Gyroscope, Accelerometer and Magnetometer are the three sensors that most of the newer smartphones have [Developers,] that can measure orientations on X, Y and Z axis, allowing the apps to calculate placement of the device in the 3D environment.

Gyroscope can be used to help determine orientation using gravity.Since it detects rotation in three dimensional space, it can be used in favour of this project to convey the rotation that is in a way similar to the one that is natural i.e. in eyesight. Gyroscope, in comparison to magnetometer and accelerometer, is the physically largest and most expensive sensor, so the possible limitations in the smart devices in-built Gyroscopes have to be considered.

REFEREN

Figure 2.6: iPhone game using a gyroscope sensor

The accelerometer can support other sensors to give the impression of the environment representation in three dimensional space. This helps to stabilize the view angle to represent real world by giving the position perpendicular to the Earth's surface. In other words, it would eliminate wrong position starting point - moving forward horizontally in reality while in the virtual environment it goes upside down could cause cluster. Accelerometer, along with other sensors is commonly used in the augmented reality concepts (Yelp Monocle, Google Ingress, SpecTrek etc). The world position calculation is not necessary for an app that is being developed, but it could be useful when wanting to change between landscape and portrait modes, if the need is established in further design iterations.

The last sensor that is able to collect three dimensional data is the magnetometer. It is typically used to measure the absolute position of the three dimensional orientation in terms of Geographical placement on earth, which is irrelevant for this project. Additionally, magnetic interference can disturb its flow, which may lead to inaccurate results. In future development of the project, an online feature with more precise positioning in relation to other users could be considered. Alternatively, the magnetometer can be used to support the gyroscope and accelerometer with positioning, if they lack precision axis-wise. If there are no problems with accelerometer and gyroscope axis measurement i.e. blind spots that these two sensors can not recognize - implementation of the magnetometer is not necessary.

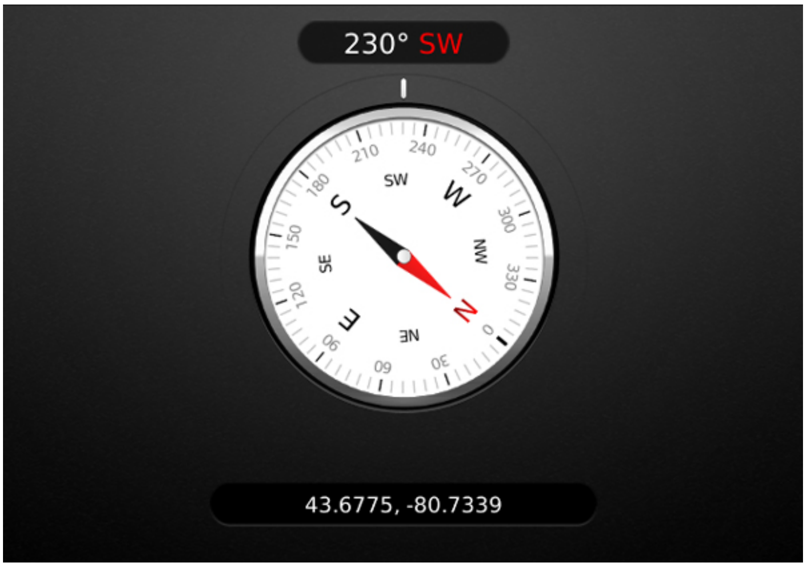

Figure 2.7: a simple compass app that establishes the magnetometer sensor

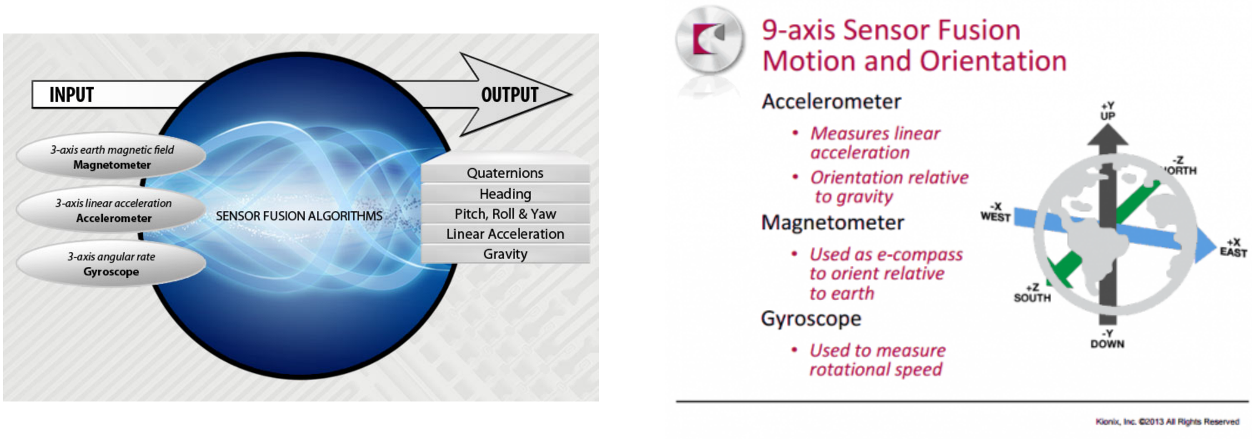

Figure 2.8: uses of sensors

2.3.3 Conclusion

The sensors discussed in earlier sections will help us work towards establishing that the perception of 3D environments is ensured to be conveyed in the best manner. After gaining more knowledge it can be seen that there is a variety of options how the use of sensors can be enabled in different application contexts.

2.4 Graphical Design

In graphical design, like any other design, there are many options to be considered. Colours, fonts, balance and many more factors should be chosen carefully. Most importantly, how to mix these elements together without making a mess. This will reflect greatly on how a design is being perceived. [Ciotti, 2013]

Colours

Colours are not just colours when designing a brand, an app or a website. Colours are perceived in various ways and is a big part of how the design is coming across to the user. [Ciotti, 2013]

It is important to remember that when choosing the colour palette for a design, that how we perceive colour is very different. Also, colours can change according to what it is next to. Yellow might look different next to grey than it will next to purple for instance. [?]

When it comes to colour psychology the truth is, it is too dependent on personal experience. There is no one right answer to which colour that represents a certain feeling. [Ciotti, 2013] There is many studies conducted on this matter. One study shows that 90% of people make snap judgement based on colour alone. [Ciotti, 2013] Another study shows that an intend of purchasing is linked with how a brand is perceived i.e. what kind of "personality" does the brand have?[Ciotti, 2013]

Figure 2.9: Overall image of how colours are generally percieved. [Ciotti, 2013]

But all in all, the concept of the app is key. Almost every study shows that it is greatly more important to choose a colour that shows the personality of your product than picking a stereotype colour. [Ciotti, 2013]

So how does one find the best way to coordinate different colours? Research indicates that the isolation effect is very useful.

Figure 2.10: "The sign-up button stands out because it's like a red "island" in a sea of blue." [Ciotti, 2013]

Using the isolation effect will help the user have a more efficient experience because the most important feature e.g. a "sign up" button, stands out. [Ciotti, 2013] Figure 2.10 Research suggests that a colour scheme that consists of analogues colors and combine it with a accent complimentary color or a tertiary color is preferred among users. [Ciotti, 2013]

When designing your layout it is key to keep everything simple and streamlined. Follow the general rules, left-to-right and top-to-bottom. Make sure the most important feature is in the top left corner where the user will look first.[Sardo, 2009] Be careful, yet not boring, when choosing a colour scheme. In general, keep the graphics clean and simple. No muss, no fuss.

2.5 State of the Art

Def: State of the art *State of the art is the level of knowledge and development achieved in a technique,science, etc, esp at present*
This section will be an analysis of a number of applications that all focus on designing a room or a set of rooms. The previous sections have provided the necessary framework for doing the analysis. Specifically the analysis will cover:

- The familiarity of the different aspects of the apps.
  What parts of the app have been seen in other apps or in real life.
- The knowledge space for the app

Here the analysis will try to determine if the app succesfully bridges the knowledge gap talked about in ux section ref: fig. 2.5.

- The graphical design
  Colours and layout.
- The different interaction methods that is used.

Finally the end of this section will sum up the trends noted and will give a overview of what aspects the different applications have lacked behind with and what aspects this project could aim to improve.

2.5.1 Ikea Kitchen Planner

In this application customers can create accurate measurements of their own kitchen and place the furniture from IKEA's catalogue. It is possible to do different wall measurements, add wallpapers to walls, apply different ceiling and floor covers, add windows and doors. Users can view the layout from top-down view and later see how the furnished layout looks in 3D perspective.

IKEA's web application for designing kitchens is used mainly in actual IKEA stores. This could indicate that users need help using this application It is used as a tool with focus on efficiency and not so much an app you would use at home for interior design. Most of the people that were asked in the initial interview also confirmed that they do not use this application. Some of the participants are familiar with the app and have tried it but do not use it.

When you first enter the app there are no immediate help or tutorial. There is, however, two places in the app where it is possible to get help yourself. Very reasonable layout going from left to right and top to bottom. There is a lot of easy-recognizable buttons which helps the user experience along. It is very slow though which highly affects the usability. The app uses click and drag.

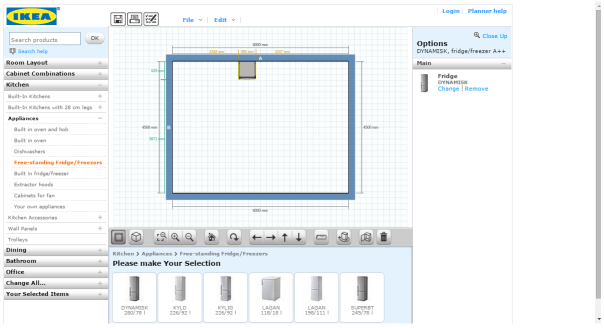

Figure 2.11: IKEAs kitchen planner webapplication in 2D mode.

The app is grey and clinical but again, it is a practical tool for IKEA customers to visualize a kitchen. The application offers plenty of useful features and can be used to give a grasp of how peoples homes would look like prior to buying the actual items, however, it is rarely used by IKEA's customers. The problem could be that the application is hard to use, leading to long time spans used to build the desired kitchen design. A solution to this possible problem could be to create an application that is more intuitive and takes less time to achieve the users needs.

2.5.2 HomeDesign3D

This mobile application is made for interior design. It has most of the basic features; building rooms, placing furniture, windows and doors. The user can then switch to a 3D view. Here there is two settings to choose from - a joystick where you use both thumbs to move around the house, or arrows where you can view the room by moving your finger around and use the arrows to move from room to room. From the 3D view you can paint the walls and change flooring.

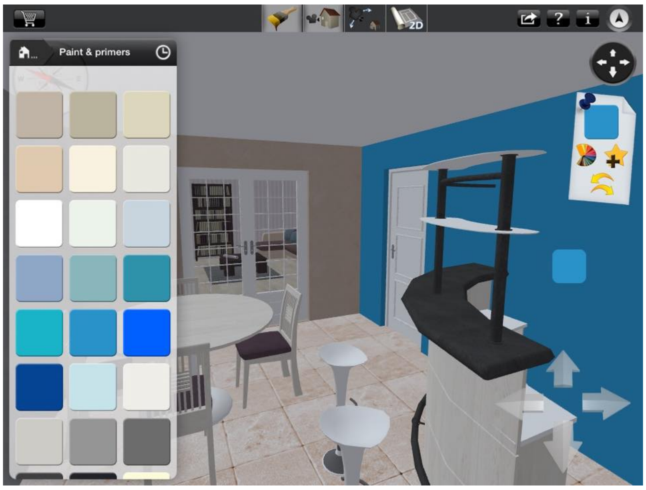

Figure 2.12: HomeDesign3D application. Painting walls.

When you first enter the app you are faced with the option of buying the full version. This is actually the start of a long tutorial. It is very hard to notice though, since the points at the bottom that indicates that you can swipe is covered by a commercial. It is very unlikely that the user will find this help from the beginning and therefore will properly "go back" to the menu. The tutorial is easy to understand but very long. It has both pictures and text. It looks messy because of the background and the hand drawn hand that shows how to do the different thing. It is not very consistent in matters of graphical design. The next step is the design. There is apparently no start menu or the likes. The icons are easy recognizable.

The layout looks very cluttered. There is the big commercial at the bottom and the menu bar in the top. The mix of the black bars and buttons with the beige background does not go very well together. The fact that the background is textured as a wall as well does not help the graphical aspect of this app.

2.5.3 AutoDesks HomeStyler

This app is Autodesk's attempt at making an interior design app. The app does not provide a user guide from when you open up the app, this is opposed to the idea about bridging the knowledge gap with tutorials. The app does however provide a guide for users, once they are actually designing their room but if the user is not able to get to this point then they are stuck. The overall look of the app is reminiscent of the flat design pattern as seen in Windows 8, this makes the app feel very modern. It also makes the app look very exclusive. However during the main activity of the app the design does not exactly match the look when browsing the catalogue, this leads to the app lacking consistency.

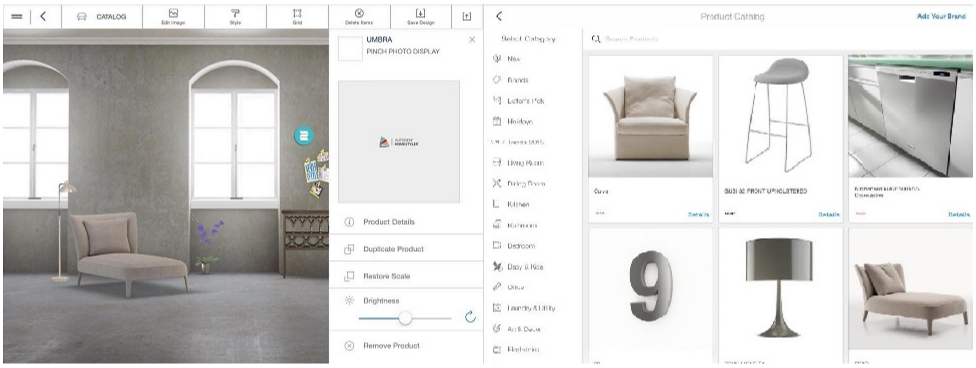

Figure 2.13: Homestyler's lack of consistency. Right: The design of the catalogue. Left: The design of the actual room planning feature.

The app's button design also seem to be focused on bigger screens than a smartphone. The big button at the top, labelled "Redesign" is a good example of both the isolation effect mentioned in **??**and also the button follows the principle of having the buttons be labelled with a single word **??**. The interaction in the app is primarily clicks with some multi touch functions for the more advanced functions such as resizing, moving, rotating furniture. These functions are explained the first time the user is using them and then never again.

fix label

2.5.4 Planner5D

Planner5D is an application made both for web and mobile. The web is however better executed than the mobile version. This app allows you to build rooms and place furniture, windows etc.

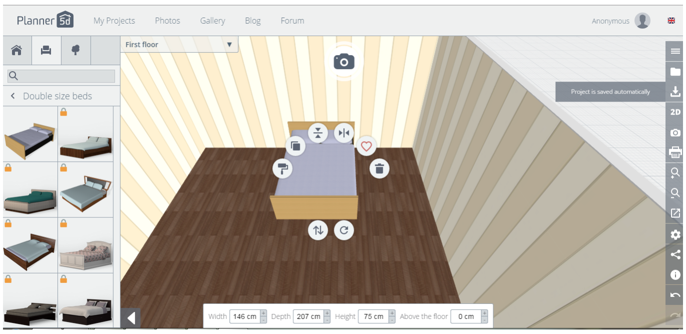

Figure 2.14: Planner5D, placing furniture.

Start screen gives a nice tutorial but is primarily composed of text. The knowledge gap is very tiny, close to nothing. The menu bars is well divided into four sections. The menu on the left is very apple like in its graphical feedback. Besides this there is a toolbox which is nicely divided by category, a bar at the top for social/profile aspects and lastly a menu that appears when you have placed a piece of furniture in the room; this is only in icons and no text hovers over this which makes it unclear what the different buttons do since the icons are not very familiar. A lot of focus on user friendliness.

The app uses mainly point and click with drag. The graphic design is very minimalistic and modern. The different features has been nicely placed which puts the focus on your own design. The furniture interaction does not match the real life movements of the user.

2.5.5 Conclusion

To sum up, the applications uses a variety of controls. Most of them uses click and drag, some a bit of multi touch and one has two options; joystick or buttons. In general the app's seem to have a different focus, therefore a different outcome. Some are very user friendly and meant as a practical tool where as others are more focused on design.

What we can conclude from this, is that only two in four uses non-traditional interaction to their advance. Only one has a tutorial that is not text based and provides the optimal knowledge at the first encounter or makes the user interface so intuitive or familiar that a tutorial is unnecessary.

2.6 Design Requirements

There are two main focus areas: familiarity and knowledge-gap. This chapter will discuss these two aspects and establish specific design requirements. The goal for the prototype is to define which way to control the 3D virtual environment is the most familiar one for the user. As mentioned in the User Experience chapter (**??**), familiarity is very related to intuitiveness. To minimize the knowledge gap for the users with digital knowledge, two conditions from the knowledge gap (section 2.2.2) will be upheld through the concept of familiarity. The users are expected to understand the controls and use them with a small amount of practice. Based on that, several design requirements need to be established. In general, navigation has to give a positive user experience and the user has to be in a state of "flow".

fix this label!!

Technical requirements

- Graphical User Interface
  - Controllers
    - The controllers have to be familiar to the user and the buttons needs to stand out by using perceived affordance, isolation, visibility, mapping and consistency principles [ref 1 -Usability][ref 2 - Graphical design] Fix this label
    - Consider graphical element placement and sizes as the navigation is for mobile platforms
- Software engineering
  - Controls has to be responsive and effective [ref 3 - UX Conclusion]
  - Controls has to give effective feedback [ref 4 - UX Conclusion]

Fix this label

Concept requirements

- Navigation
  - Navigation has to be familiar for the user
    The navigation should either relate to controls the users have used before or relate directly to the real world.
  - Make use of the gyroscope in context of familiarity
  - 3D navigation test level
    The level has to help reveal each control scheme's efficiency and effectiveness [Ref - Usability]The level also has to challenge the user to a certain degree in order to be able to compare the different designs. The level must also make it clear to the user what the objective is and how they should achieve the goal, in order to measure the effectiveness of the controls. The colour scheme of the level should help put the focus on the objective **??**rather than the surroundings. Fix this label Fix this label

Conclusion

These design requirements will help build the necessary prototypes that aim to minimize the knowledge gap using an interface to move in a 3D virtual environment on a mobile platform.

2.7 Conclusion

2.7.1 Final problem statement

How does non-traditional interaction using the concept of familiarity help minimize the knowledge gap for users with digital wisdom, for navigating in a 3D environment?

3 | Design

3.0.2 Introduction

In this chapter, the design of the prototype and the graphical decisions that were made, will be discussed and defined with the requirements set in the analysis, which were further defined in the design requirements section (2.6). When designing, different aspects will be taken into consideration for user experience; usability goals and principles (**??**), mobile usability (**??**), establishing intuition through familiarity. The user experience also depends on the technical side; a fluent implementation of the graphical user interface and non-traditional sensors, while also ensuring that the system performance is smooth. In further design iterations users would be involved in the design process (**??**), as user centred design is one of the elements to designing a good user experience (**??**).

fix all labels in this section

3.0.3 Concept

The goal of the prototype is to help answering which control scheme characteristics is the most familiar and effective for navigation in a 3D environment. Designing this kind of prototype that is not traditional can go in various directions. The concept was narrowed down to using just the sensors, as it could help establish the design requirements earlier (2.6), those being the multi-touch and the gyroscope. To be able to design a prototype with all the necessary aspects, control schemes and a base test level for navigation in 3D environment, have to be designed with the established design requirements (2.6). The prototype has to implement two features to navigate in 3D virtual environment. The moving of the camera and the camera rotation. It was chosen to make extremities of the control schemes to establish familiarity in distinct ways. The familiarity concept will be put into effect as the control schemes for controlling the camera have to be linked with something that the user might be familiar with already.

3.0.4 Interface Design

The currently existing sensors in mobiles enable the creation of unusual ways of controlling a 3D environment. With the goal in mind of achieving familiarity through the non-traditional sensors, two different approaches were established. The knowledge and familiarity to the currently existing products on the market (**??**) and the familiarity of movement representative to the one of movement in real life. For both approaches, the current and the target knowledge (**??**)is expected to be separate. The design needs to be established in a way that will help the users get through the knowledge gap intuitively when they are involved in the designed task. It was chosen to make extremities of the control schemes to establish familiarity in distinct ways. It was important to keep the same movement speed settings for all the control schemes for later evaluations. That meant that in the perfect scenario, the user should be able to get from one position to another, with the same amount of time spent for that task on each of the control schemes.

3.0.5 Control Schemes

Three distinct control schemes were designed for the navigation (see fig. **??**).Buttons-only (1), fix label Joystick-only (2), and one that includes buttons for moving back and forth with a gyroscope for moving the camera (3). The fact that newer smart devices are capable of multi-touch input, has been used as an advantage to enable both the movement and the rotation to be controlled simultaneously. That creates the possibility for the user to walk and turn around at the same time.

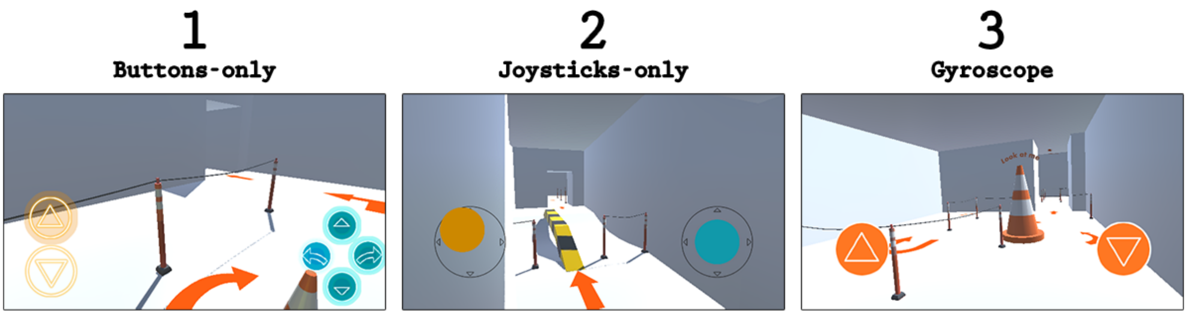

Figure 3.1: Control Schemes

Gyroscopic Controls

Establishing the gyroscopic sensor creates a possibility to interact in an unusual but familiar way. It enables the prototype to be built around what is most familiar to the real life in movement; actually moving around in reality to navigate. Controlling the camera with a built-in gyroscope in the tablet is familiar because it is a natural way for a person to look around. The forward and backward movements were implemented with on-screen buttons, as these were familiar to the target group through the usual human-computer interaction. Most of the controllers for movement are represented as this. E.g. arrows on the computer keyboard, music player, cell phone. Such control scheme may have potential to engage the user in the task the most, if the user reaches the state of flow (**??**). fix labels

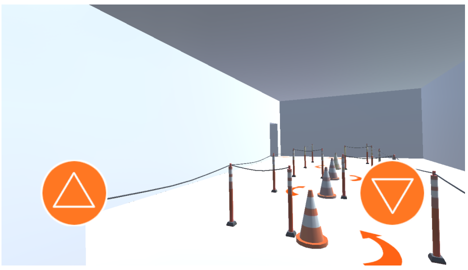

Figure 3.2: Initial sketch of gyroscope 1

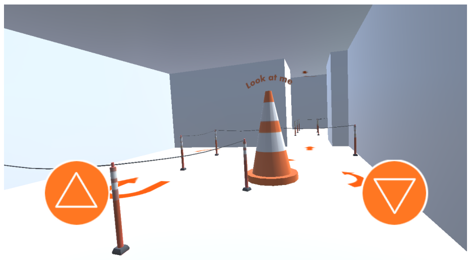

Figure 3.3: Initial sketch of gyroscope 2

Joystick

In this prototype the user has to navigate using two joysticks - one for movement and one for the camera movement/rotation. This should be easy to learn for the users who have experienced using a joystick before. The target group is expected to have some knowledge of how a joystick is supposed to work, because of the popularity in arcade and electronic games where joysticks are placed on game console's remote controls, like Sony Playstation series or Microsoft Xbox. Even for users with no previous joystick controller experience, this should not be a problem. The control scheme is supposed to borrow the same concept as moving a computer mouse on the screen in the direction, both uses 2D directional movement.

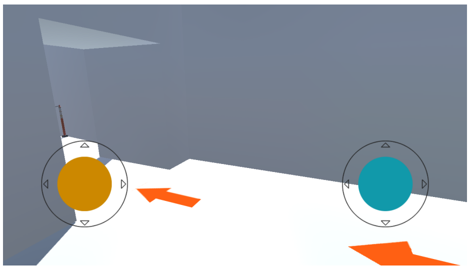

Figure 3.4: Initial sketch of joystick 1

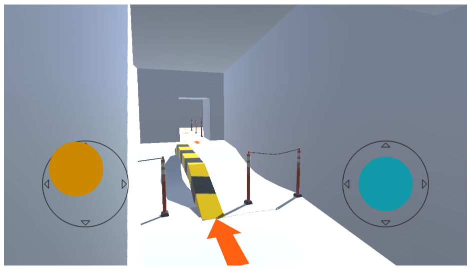

Figure 3.5: Initial sketch of joystick 2

what is the difference between these pics?

On-screen Buttons

The control scheme that should be the most familiar for the users through daily use, only using buttons as the way to move both camera and the character. In this case, the camera would be moved only with arrow keys located on the screen and the same for moving around, arrows indicating movement back and forth. This should be familiar with anyone that has used buttons for navigation of any sorts in virtual environment.

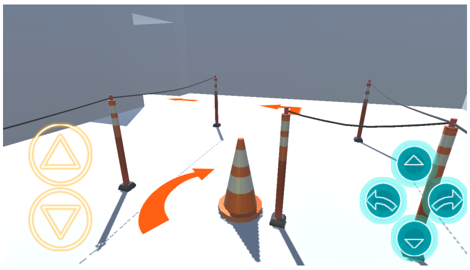

Figure 3.6: Initial sketch of buttons 1

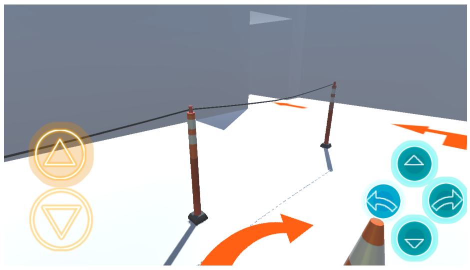

Figure 3.7: Initial sketch of buttons 2

no refs to pics - problem?

3.0.6 Isolation

To further emphasize on the initial design requirements, the concept of isolation (**??** should be used when designing the controllers. This is done by making them stand out from the level content. This is supposed to help the user understand what parts of the application gives feedback upon interaction.

3.0.7 Immediacy and Simplicity

To communicate information faster and simpler, the designs should be represented by concepts that are already familiar to the user. This means, that to communicate information to the user, graphical elements should be represented as symbols that indicate either movement or rotation for buttons. Controls that represent an actual joystick, rather than text. At the same time it emphasizes the concept of affordance **??**, as the buttons represent mechanical buttons, as used in traditional types of controllers as well as with the joystick controls. This will further shape the intuition and familiarity for the application (**??**.

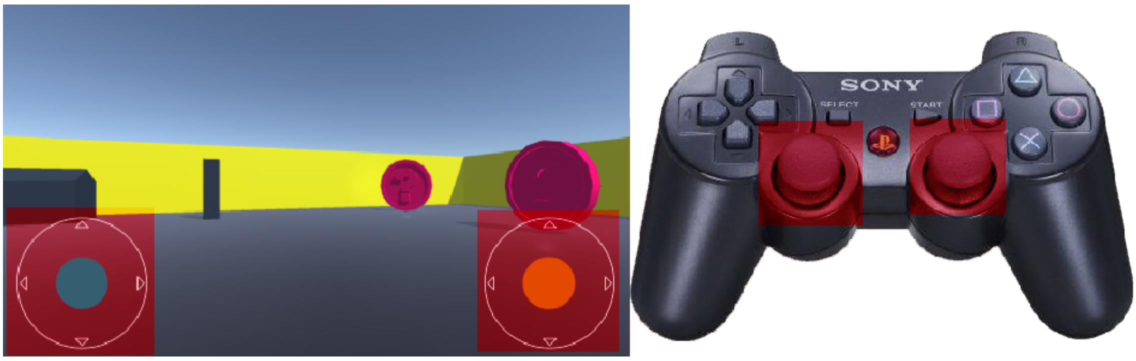

Figure 3.8: Comparison of initial design sketch of on-screen joystick and the Sony Playstation controller.

3.0.8 Graphical element sizes and placement

To further emphasize on the user experience, the button size should be set accordingly. To ensure that the users would not have difficulties by unintentionally tapping the wrong section of the controls (Section **??** mentions a case of Tetraplegia). Individual buttons have to be separated from each other and sized for easy accessibility to reduce the "Fat Finger" problem. To enable a bigger view of the environment horizontally, the application will be built to primarily be viewed when holding the device in a landscape mode. Since the device is supposed to be held sideways and by both hands, all of the interaction should be done on the sides of the screen for easy control access. To show the difference between movement and rotation controls, they should be given different looks. Shapes of directional arrows for buttons as well as color indication for both buttons and joystick controls.

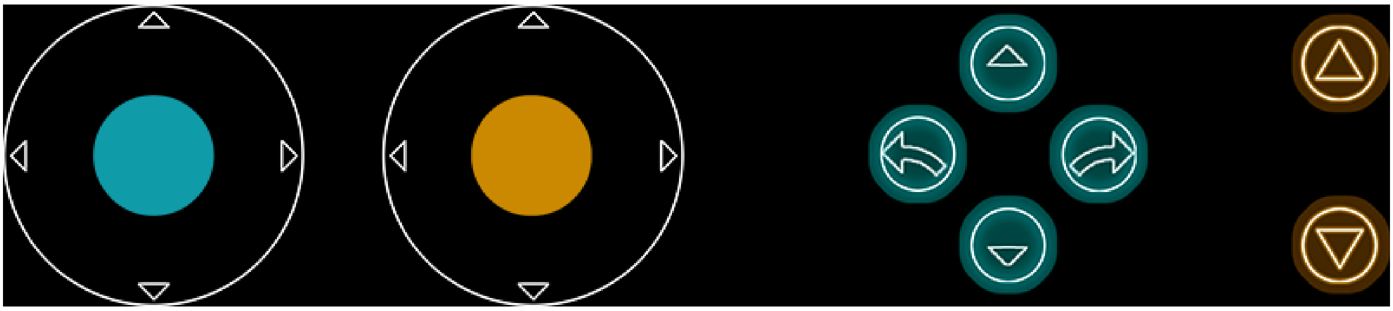

Figure 3.9: Initial sketch. Colour differences to distinguish controllers that hold different functions

3.0.9 Design of 3D testing area

To test different non-traditional control schemes 3D test area was designed. It consists of twisted path that test participants had to walk through as fast as possible 3.10.

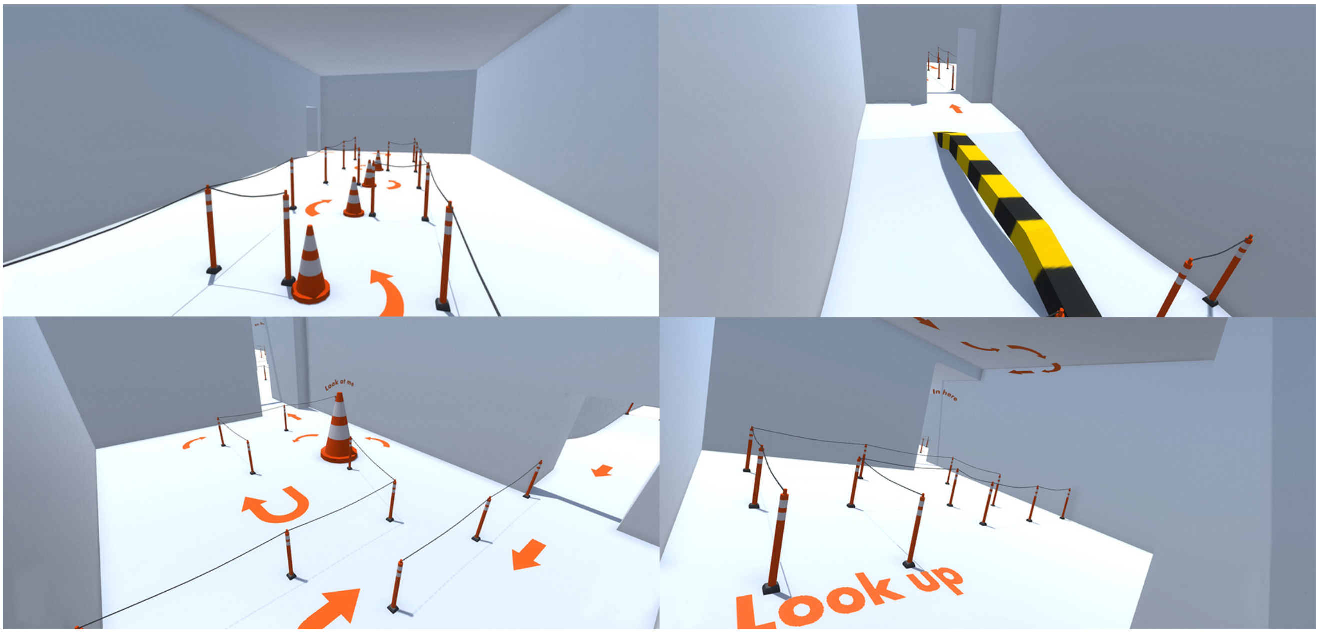

Figure 3.10: Pictures of the testing area

Focus of testing area was only on navigation for different schemes, so effort was placed on path and tasks and not on environment itself. All not important bits of the area were colored white-neutral color and important as navigational arrows, cones and poles, explanatory text notes colored with sharp contrasting colors as red and orange. The "familiarity" consideration was done - navigational buttons and important bits on the level were kept in similar color, to help

user recognize and focus on the purpose. Analysing SOTA's applications gave understanding what important aspects of navigation is.
This helped to create testing goals.

1. Firstly walking fluently around obstacles as furniture, doors, narrow paths. First two levels were developed for this to test. Second consideration was that users need to look around placed furniture which was not considered in most SOTA analyzed applications. Small area with huge cone and text "look at me" was placed and arrows in circular path around it. This represents how user would walk around furniture and inspect it by looking - focusing on one point while walking around 3.11.

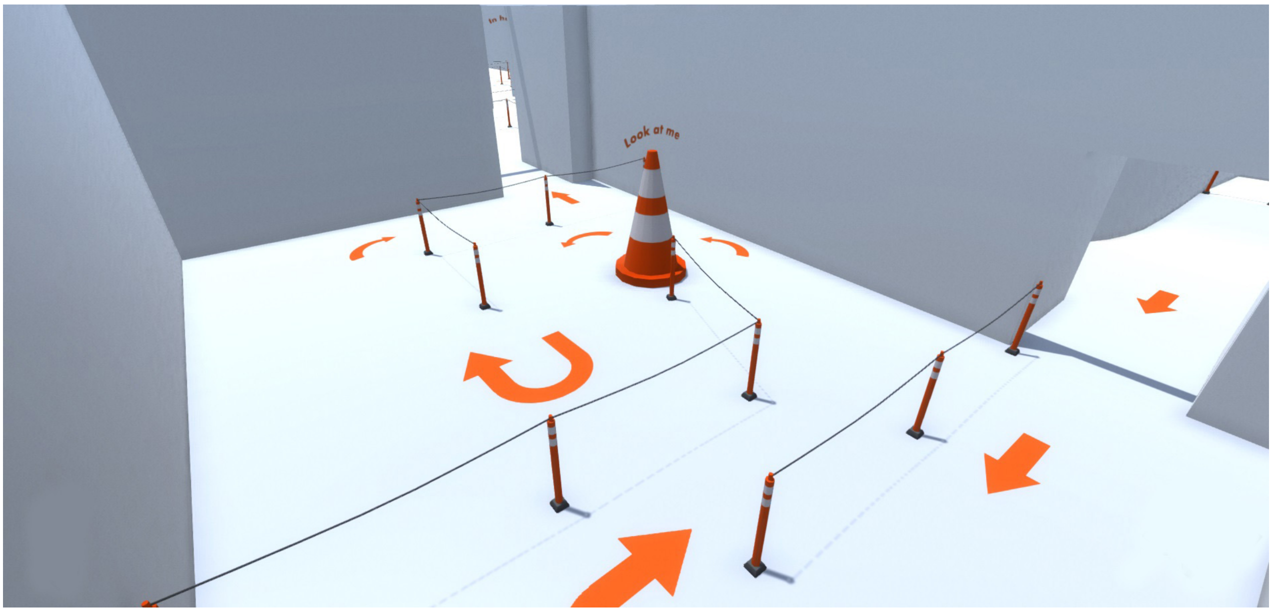

Figure 3.11: Third level of testing area. This level is to test how well user walks around an object.

2. Next two levels were developed to test how efficient it is to look up and down while walking in given direction. It helps to test how well user can walk and look down or up at the same time 3.12.

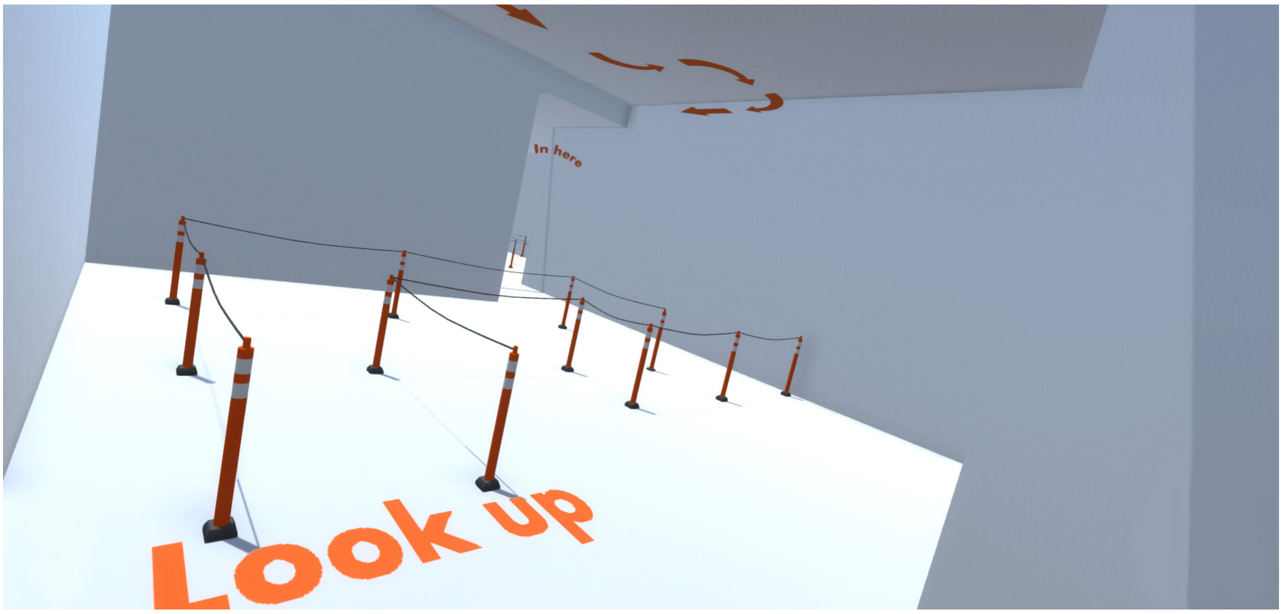

Figure 3.12: Test chamber to test how well user can walk and look up

3. The last test area was created to see how user goes straight but looks to one side as the user would walk-by, but focus at some furniture aside 3.13.

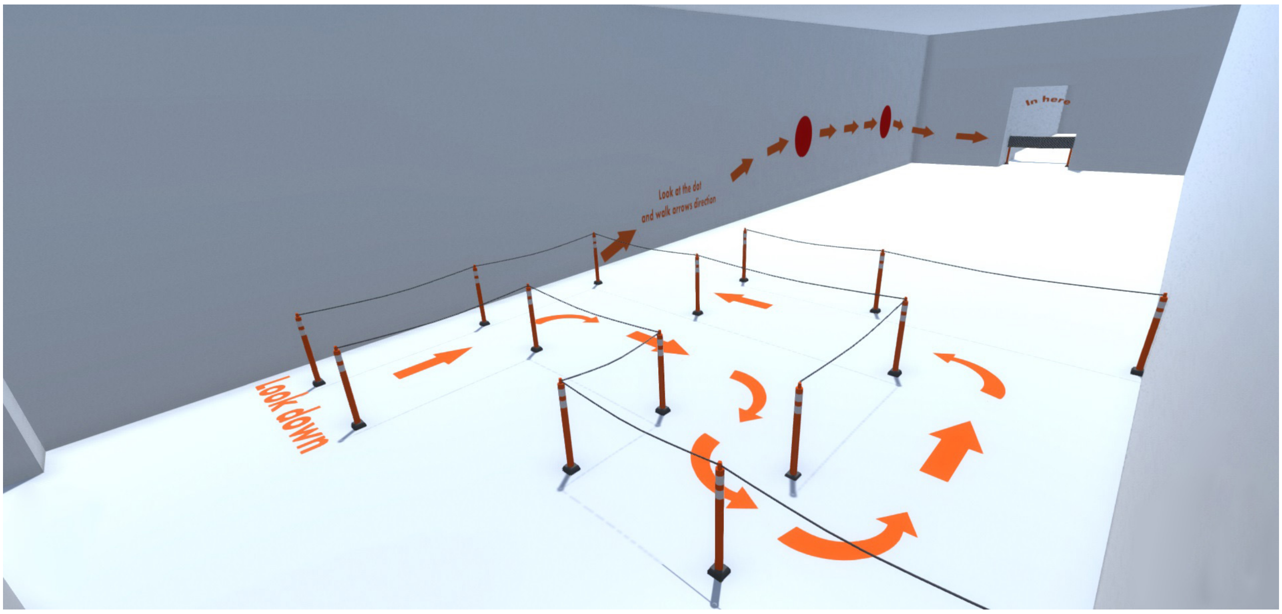

Figure 3.13: Last level of testing area. This level is to test how well the user can walk while looking down and sideways

3.0.10 Design Conclusion

Three distinct prototypes and a test level were designed for navigation in 3d environment with all the initial design requirements in mind, with a goal to create a design that reduces the knowledge gap for the user the most. These control scheme design initial prototypes will be implemented as three seperate mobile applications for tablets and smartphones with gyroscopic sensor for initial prototype testing. Designs will then be evaluated to help in the iterative design process. In addition to that, since these are not the only possible control schemes that the problem area covers, alternative prototype design possibilities will be discussed in the discussion and redesign chapters in the report.

4 | Implementation

This chapter will be a thorough explanation of the implementation of the prototype. As described in the design chapter the following are what will be implemented:

- 3D level
- 3 control schemes:
  - Joystick Controller
  - Buttons Controller
  - Gyroscopic controller
- A script that will enable all the controllers to work with buttons.

4.1 Unity Introduction

For this implementation the choice was made to use the freely available Unity3D engine. This was chosen as Unity provides a lot of features that would otherwise mean that this project would not be possible to complete in the amount of time given. This section will provide a quick glance at what Unity is and introduce some common terms used within Unity. Everything that is seen in a Unity game is composed of `GameObjects`. These objects are what the scripts described in the following sections will be attached to. A newly created `GameObject` will only contain what is known as a `Transform`. The transform is an object that holds the information about where the object is within the 3D space. It also holds the objects rotational information and its scale. The position of the object is expressed as a vector. A vector is a mathematical tool for representing a coordinate in space. As such, vectors are the primary tool that is used in unity for expressing things like velocity, position, forces etc.

4.2 Development of 3D elements

The actual implementation consisted of creating level areas, 3D prefabs; cones, poles with string, sprites, notes and arrows. 3D elements were created using the software "Maya", which is a 3D modeling and animation programme. Textures were made with "Photoshop". Each level started by a simple creation of a square representing a room. Then considerations of what the test should include were done and assets were placed to making different challenges. Picture 4.1 shows a 3D cone and its texture.

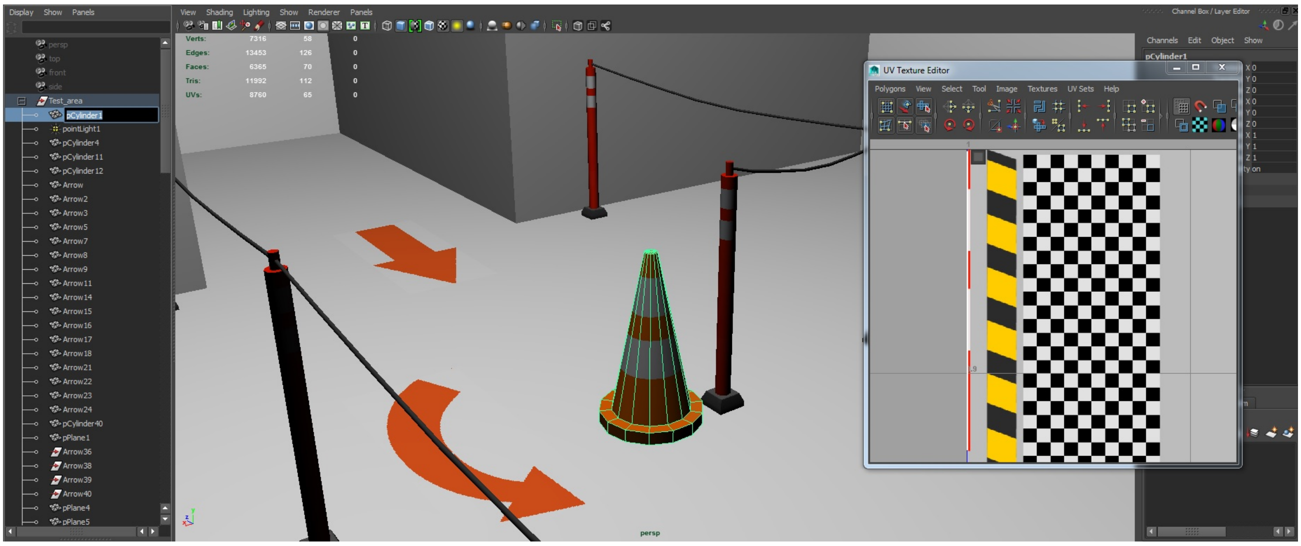

Figure 4.1: Picture of a cone from the testing level

Every object that required texturing, also needed its own UV map. UV maps are a three dimensional model representation in two dimensions. This helps to see where to paint textures using tools as Photoshop. Following picture 4.2 shows UV map as red lines, that helps to see the boundaries for the texturing in Photoshop.

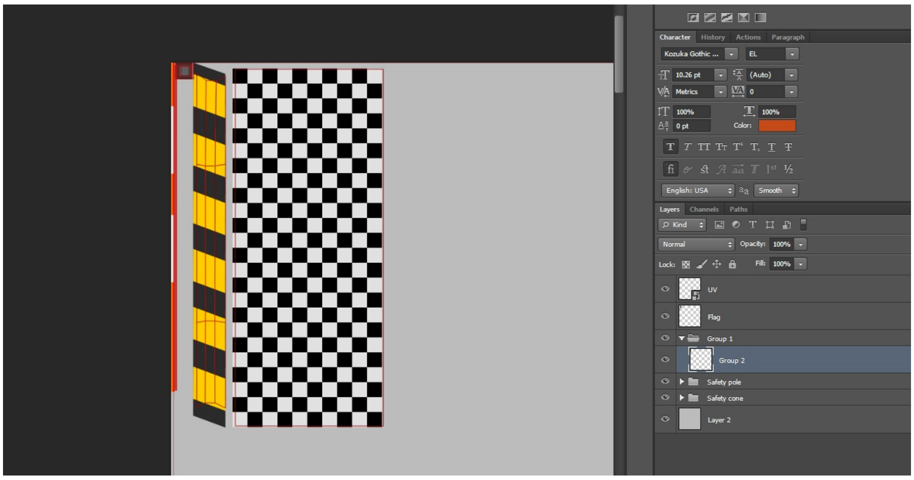

Figure 4.2: Picture of a cone from testing level

Text, directional arrows and other symbols needed to be transparent so it would blend well with most of the background, see fig. 4.3. Such images were exported as .png picture format to reserve the transparency.

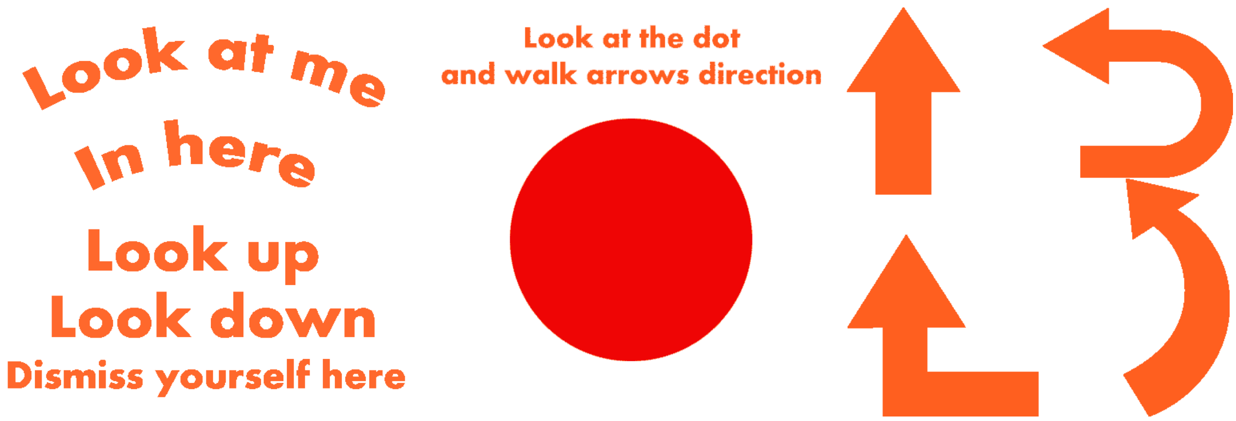

Figure 4.3: Pictures of elements that required transparency for game engine, also called sprites.

Since graphics and aesthetics were not the focus of this test level, textures were created very minimalistic without any shadowing, noise, imitation of being old/used etc. It also helped making the test optimized for hardware performance, since the textures size on a disk were as little as a few kilobytes.

4.3 Buttons

The buttons used in all three schemes are all created using unity's UI system. A button in the UI system is just an empty `GameObject` that holds the following components:

- `Rect Transform`
  This component is equivalent to the regular `Transform`. This component does not contain any information about the objects position in 3D space though, as it is part of the UI it only needs to know its coordinates on a 2D plane. The UI system provides a convenient way of making sure that the UI will look the same, no matter what screen size is used. It does this by having coordinates relative to an anchor point that is set by the developer.
- `Button`
  This is the standard script that unity uses to detect clicks on a button and can call methods on different objects when clicked. This script is the script that has been modified to allow for pressing and holding.
- `Image`
  This component is a component that will draw an image on the canvas. This image will be the full size of the button.

4.3.1 `ButtonScript`

`ButtonScript` inherits from the in-built Unity `Button` object. This is done so that the script can access the method `protected bool IsPressed()`. This script is the one that all buttons in both the buttons and gyroscope scene uses. It is a simple extension. All it has to do is to check if the button is being pressed and if it is, then check which button it is and act accordingly.

4.4 Buttons Scene

The button scene is the only scene that only uses `CamMovement` and `ButtonScript` and is therefore the simplest scene technically. Presented below is the methods contained within `camMovement`:

- `public void move(bool rightButton)`
  This method will, depending on the boolean given to it, move the camera forward or backward. This is done by creating a directional vector formed from the transforms forward vector:

  ```
  Vector3 directionVector = new Vector3(transform.forward.x,
                                        0,
                                        transform.forward.z)
  ```

  The directional vector will always have a y component of 0 as the camera should not be able to move in the y direction. This vector is then multiplied by an integer variable that will take the value of either -1 or 1, depending on the value of `rightButton` by converting the boolean into an integer with the line:
  `int goingForward = rightButton ?  -1 :  1;`
  Finally the directional vector is multiplied with a variable for determining the speed of the movement. This directional vector is then set as the velocity of the rigid body that is attached to the GameObject.

- `public void rotateLeftRight(bool right)`
  This method will rotate the camera left or right in a similar manner to `move`. But where `move` does not do anything to its y component, the `rotateLeftRight` method will only rotate in its y component and keep the rotation of the x component. This method uses Quaternion.euler to convert our euler angles into Quaternions and `Quaternion.Slerp` to smoothly make the camera rotate. This can be seen in figure 4.4.

```
Quaternion target = Quaternion.Euler(transform.rotation.eulerAngles.x ,
                                     transform.rotation.eulerAngles.y + 2 * rotationSpeed * rotateRight,
                                     0);
transform.rotation = Quaternion.Slerp(transform.rotation, target, Time.deltaTime * smooth);
```

Figure 4.4: Conversion of quaternions into euler angles and use of the slerp function.

- `public void rotateUpDown(bool down)`
  This function is more or less the exact same as `rotateLeftRight` but where `rotateLeftRight` makes changes to the y component of the rotation, `rotateUpDown` rotates the x component.

- `public void stopMovement()`
  This function is a simple function to make all movement on the camera stop. It will set both the linear and angular velocity of the object to zero.

4.5 Gyroscope Camera

The gyroscopic camera uses two scripts to work, first and most importantly the `GyroController` which is the script responsible for getting the input from the gyroscope and rotating accordingly. Unity can take inputs from the gyroscope directly but if an object is rotated by the raw data from the gyroscope, there will immediately be a problem as the coordinate system that is used by the gyroscope is a right-handed coordinate system

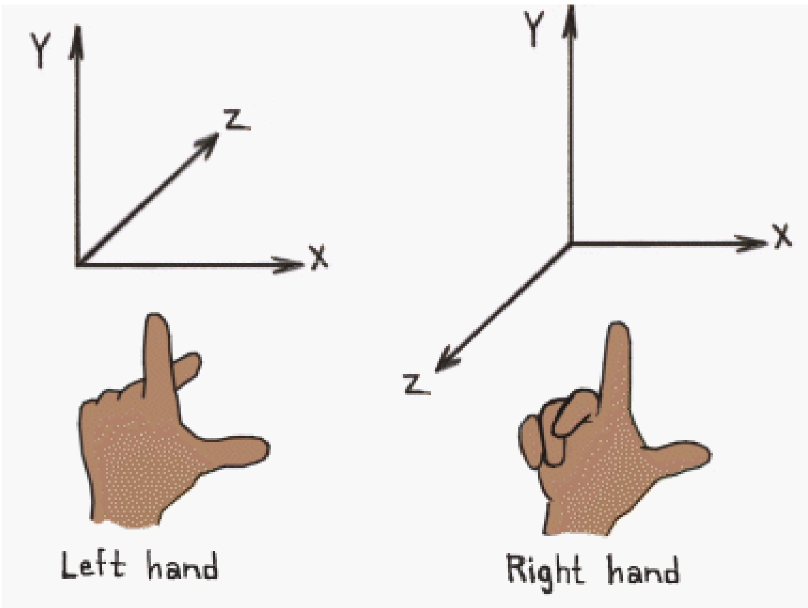

Figure 4.5: Right-handed vs. left-handed coordinate systems

where unity uses a left hand coordinate system. To have the coordinates given by the gyroscope match the coordinates of Unity, the `GyroController` has a method for converting from right-handed to left-handed coordinate systems. This method uses Quaternions which is how Unity handles rotation internally, explaining quaternions is beyond the scope of this project. Besides the `GyroController`, this scene also uses the `CamMovement` and `ButtonScript` explained in the previous section. In this scene, only the forward and backward buttons are used though, as the gyroscope is handling the rotation of the camera.

4.6 Joystick Scene

The joystick scene is technically the most difficult it uses some of Unity's standard assets, these include:

Mobile Joystick

The mobile joystick script creates two axis for unity's input manager. These two axis are then accessed by the first person controller and uses these values to move the camera.

The Character controller

This is actually a fully fleshed component along with components like the transform, renderer etc. For this scene it is primarily used to generate a collider for the camera.

First Person Controller

Unity provides a standard first person controller that comes with a whole range of features. For this scene the first person controller needs to take inputs from the joystick. To do this the first person controller's `RotateView()` method has been modified to take input from the virtual axis created by the mobile joystick:

```
private void RotateView() {
    Vector2 input = new Vector2(CrossPlatformInputManager.GetAxisRaw("HorizontalLook"),
                                CrossPlatformInputManager.GetAxisRaw("VerticalLook"));
    float camX = m_Camera.transform.localEulerAngles.x;
    if ((camX > 280 && camX <= 360) ||
       (camX >= 0 && camX < 80) ||
        (camX >= 80 && camX < 180 && input.y > 0) ||
        (camX > 180 && camX <= 280 && input.y < 0))
        m_Camera.transform.localEulerAngles += new Vector3(-input.y * m_LookSpeed,
                                                           m_Camera.transform.localEulerAngles.y,
                                                           m_Camera.transform.localEulerAngles.z);
    m_YRotation = input.y;
    transform.localEulerAngles += new Vector3(0, input.x * m_LookSpeed, 0);
}
```

Figure 4.6: The modified `RotateView()` from the first person controller.

This method reads the input from the axis named `HorizontalLook` and `VerticalLook` and stores these two values in a vector. This vector is essentially the directional vector we want to add to the camera's rotation. While a solution such as;

```
Vector3 dVector = new Vector3(input.x,input.y,0);
transform.localEulerAngles += dVector;
```

will make the transform rotate in the direction of the vector, it will also cause the camera angles to get stuck. This problem is known as gimbal locking, which is the situation where two rotational axis is pointing in the same direction. To avoid this problem the script checks to make sure that the camera is not rotated into a gimbal lock and adds the rotation to the local rotation of the camera. The script does this by checking if the user is looking down or up at an angle of less than 80 degrees.

5 | Evaluation

The evaluation is conducted to conclude whether the problem stated in this report is being solved. The answer will be acquired when the data gained from evaluation is analysed and concluded upon. This section will attempt to evaluate the control schemes and will cover methods and different approaches planned to be used when conducting the evaluation. It will be discussed why usability testing and performance testing methods were chosen for evaluation of the prototype and why it seemed the most efficient and suitable choice.

5.1 Usability testing

This testing method can reveal various errors that could occur when a user interacts with a product. Usability testing is a method that seeks to test five areas. [Nielsen, 2012]

- Efficiency How quickly can the user complete certain tasks when they know the design. [Nielsen, 2012]
- Satisfaction How satisfied are the users with the design. [Nielsen, 2012]
- Errors How many errors do the users make. Is it severe errors and how quickly will they recover. [Nielsen, 2012]
- Learnability How easy is it for the user to complete certain tasks when they first encounter the design.[Nielsen, 2012]
- Memorability How well do the user remember the design after some time away from the design. [Nielsen, 2012]

The benefits from usability testing is that it identifies major usability issues from a few number of participants. [Nielsen, 2012] According to Jakob Nielsen it is enough to have only five participants as the problems will show clearly based on this alone. [Nielsen, 2012]

There are several ways to conduct a usability test e.g. focus groups, user testing etc. The following will describe the methods we chose to carry out.

5.1.1 Performance testing

Performance testing is a method that focusses on, as the name reveals, performance. This can be measured in different ways. For instance, time and numbers of errors made. It allows the facilitators to obtain measures of effectiveness and efficiency. [per,]

The benefits of performance testing is that it reveals major problems including problems related to the users skills and expectations. [per,] Tell the user how to achieve the goal but not how to do it [] Observe and measure without commenting[]

5.1.2 Card Sorting

Card sorting is a method for discovering latent structured. [car, b] It allows the test participant to give critical feedback without having to do it directly to the testers. You will need fifteen participants for a card sorting test [car, a]. Normally a usability test only need five users but this testing method needs more participants to get a full view of the users preferred structure. This can not be accomplished by five participants. [car, a] As Jakob Nielsen says [car, a], this test differs from other usability tests by being a generative method. This means that we do not yet have a design and need to establish user needs first as we tested for the navigation before we designed the actual app.

According to Jakob Nielsen, the classic way to ruin a card sorting test is to give the user familiar command names. This will make the user look for that specific command name instead of acting as they normally would.[car, a] E.g. Do not say "now you will use a joystick" and thus giving them the information that this should be controlled as a joystick and the usability problems there might be will not be revealed. It will not be certain if the test participant actually came to the conclusion that it should work as a joystick themselves.

Card sorting can be used in various ways. To group words, to name groups of words, to describe a product and more. We used it to make the test participant feel more comfortable choosing a critical word rather than interviewing them. Here is how we used it together with the performance test.

5.2 First test - Navigation

The first thing that is tested is the navigation. This needed to be established before making the actual design and to make sure that the app is usable. It is a usability test that will consist of two parts; performance and sort-carding. The test is set in a controlled environment where the participants need to go from A to B, testing the control schemes, rotating the order of what control scheme they start out with.

Before the test starts, the test participants will watch us go through the level so they know where to go. This should eliminate some bias when it comes to getting to know where to go and should help keep the focus on how to get there. Otherwise the first control scheme would be slower every time as they would have to find their way through the level.

We will time them to see which control scheme is the easiest to control. We will also observe how much they struggle and how fast they get from A to B. Also we will estimate how long time it takes them to have a grip on the controls.

After each control scheme we will ask them to do a card sorting. They pick 5 words and afterwards they would be asked why they chose those particular words. This will help us getting them to say something critical about the controls to us.

We will make a scoreboard so the test participants can see how fast they completed the course compared to the other people who tested it. This should add a competitive element and some fun to the test.

5.2.1 Analysis of data

The first thing that was done after testing, was color coding the cards. This way we could sort them out into categories. Then we made bar charts to get a visual and clear image of the results from the card sorting. The cards were sort after:

- Intuitive/familiar UX
- Positive feelings
- Negative feelings

- Complaints
- Unusable. These cards were not related to the control schemes.

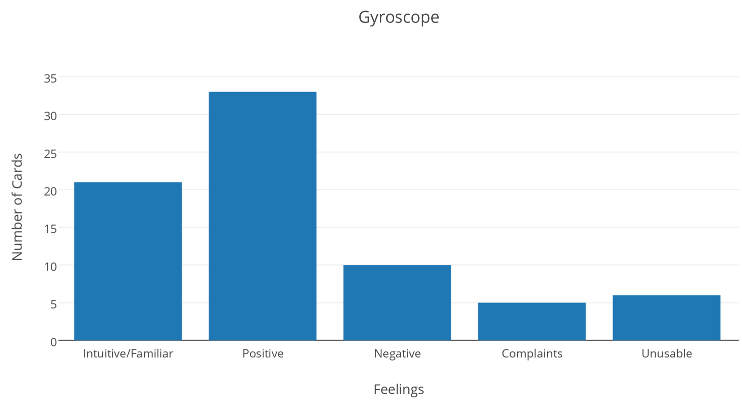

Figure 5.1: Categorisation of the cards for the gyroscope control scheme.

The gyroscope turned out to be very intuitive and quite fun for users. It has a small amount of negative feelings and a few complaints. These were mostly based on the fact that you have to turn all the way around and not being able to sit down while using this and that it moved to slow. The intuitive cards that were selected showed that the users thought it was nice that their movements was met in a virtual world. Also they thought that it was easy to understand once you have practised a bit. The positive feelings towards the gyroscope was quite clear; it was fun, exciting and friendly. Almost all positive cards had something to with the fact that this was new and that the users had not yet experienced this which made it more attractive.

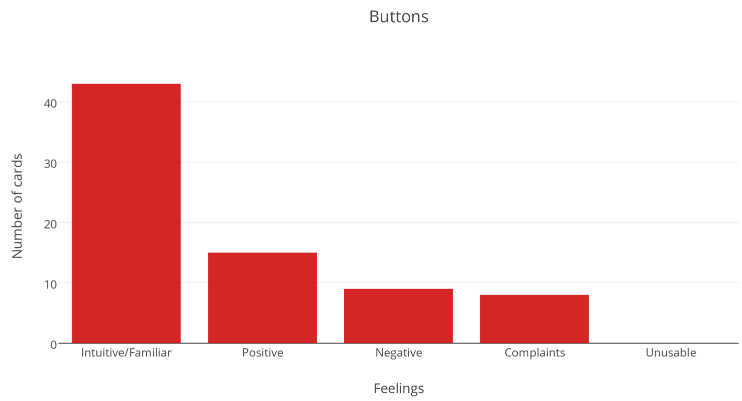

Figure 5.2: Categorisation of the cards for the button control scheme.

The button control scheme has the absolute highest number of intuitive/familiar feelings. This makes sense since it is common for users to use buttons. It has about the same amount of negative feelings and complaints as the gyroscope. These feelings were however quite interesting. 5 out of 9 negative cards said dull or boring. The users found that this was to common and had been seen before. It was clearly the most efficient as the intuitive and positive feelings were all directed towards the speed, how easy it was to use and that it was trustworthy. There is no way we can actually improve on the negative remarks because if the buttons are too last year our hands are tied.

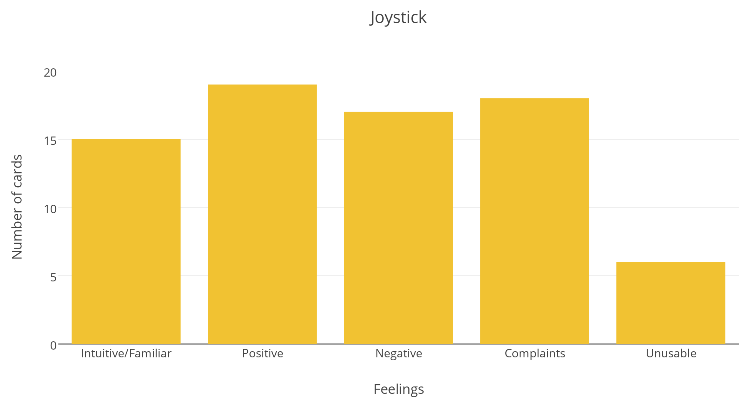

Figure 5.3: Categorisation of the cards for the joystick control scheme.

Lastly, the joystick has a lot of negative feelings and complaints. Even more than it has intuitive/familiar feelings. This clearly shows that the users did not enjoy using this controlling scheme as much as the others and was quite frustrated with it. This could also be provoked by the way we chose to implement this and the fact that it did not resemble a joystick well enough. The main issues that was pointed out to us in the negative feelings was that it was frustrating, hard to control. The users commented on the camera the most. Also 8/18 complaints was directed towards the cameras movement and that the joystick lacked sensitivity. These negative feelings could maybe have been avoided if the implementation had been better. The positive feelings however was flexible and creative. The users thought it was fun, once you get to know it. The few that did give this control scheme a intuitive/familiar card thought that it was very clear that this was a joystick but still struggled to control it due to the cameras movements.

The joystick might not be as bad as it appears in the bar chart. The main reason for the high number of complaints and negative feelings was that the cameras movement was out of control. This could have been prevented by better implementation. So as far as the joystick goes, we will not know if this is the reason for the feedback until we can fix it and test it out once more.

The time from the performance tests was calculated to find the mean and made graphs to demonstrate this as well.

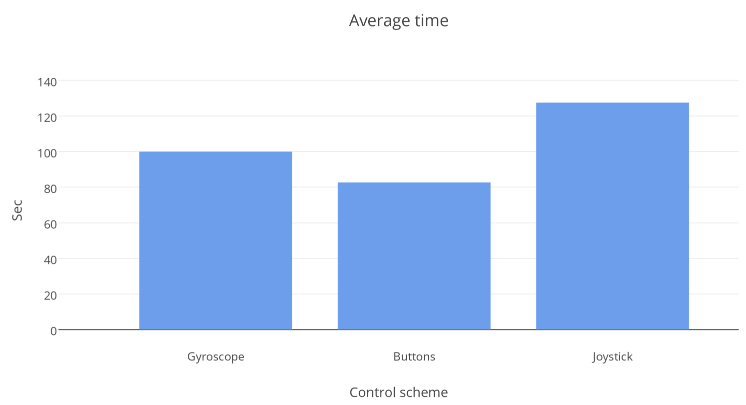

Figure 5.4: The average time it took for the participants to complete the level within the different control schemes.

It is fairly easy to conclude that the buttons were the most efficient and the joystick the most problematic for our users based on the time alone. But is the buttons the most successful after all?

It shows very clearly that the users had most intuitive/familiar comments about the buttons scheme. And the one they had the hardest time coping with was the joystick. Both the performance test and the card sorting test showed that the fastest, most intuitive and most efficient was the buttons scheme. However, their emotional response to this was not as we hoped. The one controlling scheme the users found the most exciting was the gyroscope. They did have some complaints, but these were mostly based on the fact that there was a learning curve and some technical issues that can be fixed.

If we were to compare the gyroscope and the button scheme for final conclusion the users had 58 positive feelings/intuitive about the button scheme and 17 negative feelings/complaints. Where as the gyroscope had 54 positive feelings/intuitive and 15 negative feelings/complaints.

5.3 Conclusion of results

In conclusion, we now know that we will not be using the joystick. Whether we choose the gyroscope or the buttons ultimately depends on our focus. Do we want a control scheme that is exiting, new and fun or do we want one that we know for sure will cause the users no usability problems and be the most efficient although they think it is dull? The final problem statement is interested in making an application that minimizes the knowledge gap using familiarity. Also, the user experience section suggest that when people think something is fast, it is good UX. 11 And furthermore, it is concluded that we want an app that is easy to remember and learn how to use. 15

Based on this, we can conclude that the buttons is the optimal choice for our navigation app. Even though the users were more emotionally engaged in the gyroscope, the focus for this app lies within familiarity and not excitement. And even though the negative cards that were chosen in the card sorting suggested that the users found this boring, it still sets the bar high with 43 cards chosen on familiarity and intuition.

6 | Discussion

This first section of the discussion will focus on the evaluation chapter. Next the discussion will look into the main points that can be used to make the redesign better. The discussion will then look at some of the choices made along the way for which control schemes that were implemented and look at what alternatives there might exist. Finally the chapter will look into some of the technical difficulties that the Joystick control scheme experienced as well as look into what could have made the test level better suited for testing navigation.

Test related discussion

The testing of this implementation, have a few points that should change, specifically three points have been identified that has had a negative impact on the test. These are:

- Test environment
- Tested users can be classified as gamers
- Competition

Test environment

During the testing of this prototype the environment was not taken into account as it was done in the team's group room. This introduced some problems that should not have been present, for instance: Two test participants were brought to test for the testing team, these were not separated during their individual tests and the testers were not told not to talk during the test. This might have affected their responses as well as their completion time. This could have been fixed by having the participants test the app in one room and do the card sorting in another.

If the tests had been carried out in two rooms, one room could have held a one-way mirror where the testers could have observed the test participants from and also filmed. This could have allowed an in-depth analysis of the emotional responses from the participants.

Tested users are gamers

An issue that was not taken into account before the test was carried out, was the fact that many of our test participants were not only digital natives but also gamers.[1] This meant that our target group was not well represented. They may have had too many skills than we anticipated which may have affected some outcomes in the card sorting test. E.g. that the buttons were boring, that they are used to using a joystick and have a smaller knowledge gap than the digital natives, as the current knowledge point are significantly closer to the target knowledge point for gamers. It should also be noted that while our test participants found the button controls the most familiar, a digital immigrant who has no preconceptions about how navigation should work, might find the gyroscopic controls easier to learn, as it relates more to the real world. This hypothesis is backed up by anecdotal evidencefrom when we showed the prototypes to older people.

Reference to appendix

Anomalies in the test data

When looking at the data gathered from the tests, the data of two test subjects stands out as anomalies. While the joystick is generally slower than the other two control schemes, there was one test participant who spent nearly twice as long as the second slowest test participant. The deviation from the norm in this test was so large that it single handedly increased the average time by close to 25%. Another anomaly we noticed was on the gyroscope. One of our test participants ran into some technical issues during the test, to which we could attribute his overwhelmingly negative comments. This test participant accounted for 40% of the gyroscopes negative comments. One must note, that the first of these anomalies can be eliminated since it's part of our quantitative data, while the other can not since it's qualitative, even though it has a big impact on the data.

reference to appendix?

[1] people who play video games on any platform

Competition

The test was designed as a competition in hopes of motivating the user to do their best i.e. as fast as they could. This worked well during the test but it later showed that the test participants might have been too focused on the competition rather than the navigation. This could also be because they were not instructed properly before the card sorting. It affected the card sorting results, as many chose cards like "fun" or "engaging" and reasoned it with the fact that it was fun because it was a competition. We quickly noticed a trend in this and made sure to instruct the subsequent test participants that the comments should be related to the control and the controls only. We also chose to disregard the comments related to the test's premises.

Test data to redesign

In order to test whether the button controls are actually the most efficient, another test should be done where the test participants get a chance to play around with the controls before being timed. This should eliminate, or at least reduce the time needed to learn the controls, allowing the test to focus only on efficiency instead. While the joystick controls received mostly negative feedback there were some positive comments though, particularly on the fact that it alone allowed the user to move sideways instead of just back and forth. This feature is something that shall be brought into the next iteration. The fact that the users would like to be able to move sideways can relate to the fact talked about earlier, that a lot of the test participants are gamers as mentioned on page 6. In most games with a first person control scheme being able to move sideways is the standard.

Control schemes choices

Analysis of State of the Art (**??**) gave an understanding of which controls are most common and should be most familiar to users with digital knowledge.

Limited Control Designs

The control scheme designs were limited to three which were implemented, as they were the ones that initially seemed to answer the minimizing of the knowledge gap through familiarity.

All three designs were based on familiarity. Either through the user's previous experience in navigation in a 3D environment, or through what would seem familiar with a natural way of moving in real life. However, design alternatives should not be overlooked, as possible combinations of the initial control schemes could bring new interesting outcomes. There is a wide variety of possible alternative ways of controlling a 3D environment. In the Redesign section(**??**) possible alternative control designs will be mentioned.

Joystick - not calibrated nor adaptive

Testing showed that there is one main problem with the joystick control's implementation. Graphically the joystick is displayed in the center of its boundaries which are represented as a white circle. But it was implemented as though the actual center is in the corner of the joystick. Additionally, some of the users were focusing on the task rather than the controls. The current state of the joystick controls only works when the fingertip is placed in the center and dragged from the center. Some of the users placed the finger on the side of the control field assuming that the camera would still turn. There are different approaches in solving this problem and few of them will be discussed in the redesign chapter (**??**).

Gyroscopic approach and the alternatives

When designing the gyroscopic control scheme, different approaches were discussed. The approach taken was to eliminate all the navigation from the screen that could be done in an alternative way (by moving the device). This control scheme was the one with least amount of navigation that required design elements on the screen, instead the rotation of the camera had taken a completely different approach. This approach has led the project in an interesting direction. Only from rough general observations, it could be seen that the users were engaged in this way of navigating in the 3D environment. As much as this control scheme could be a working approach to navigate in virtual space, some general things have to be considered. For instance, moving around for a longer period of time or being in an environment that is not optimal for moving around (e.g. sitting). Although the most of the test participants did not point that out, it may be an issue in the future when the problem area expands. Alternatively, an additional version of a control scheme that uses the gyroscope sensor, could go into the next iterations is discussed in section **??**.

Consideration to alternative control schemes

To further emphasize on the familiarity that this project is aiming for, additional alternative control designs could be considered that aim to convey familiarity to the one that represents real life, or the navigation controlling in a virtual environment that the user has engaged with before.

When it comes to intuition and familiarity, the research in the field of navigating in virtual environment is limited. Therefore general guidelines for navigation in such environment are hard to come by. When it came down to designing different control schemes, the ones created seemed like obvious choices, as they fit the current knowledge point that was aimed within the target group. To create a deeper approach on how the new alternative control schemes should be initiated, there is a need for a better definition of what characteristics the control scheme has to contain to be defined as familiar to the person. This could be done by further analysis of the target group and how they interact with different control schemes.

Technical Difficulties

When the team decided on implementing the three control schemes there was no real way of estimating how difficult implementing each control scheme would be. This meant that in order for the prototypes to be ready for testing, the implementation was not planned as rigidly as it could have been. This lead to the joystick having more technical difficulties than the other two control schemes. Furthermore, the joystick implementation is composed of both the teams own assets and some of the standard assets provided from Unity. This means that for the team to be able to modify the code, the provided code from unity had to be analysed and understood. This difficulty has caused some negative test results for the joystick as mentioned in the evaluation chapter (**??**).

3D Level design

The test level was effective and helped to make tests with useful results. However there was one small aspect that needs to be addressed. The plan was that participants would go through the level as fast as possible. However, the level was designed for further possible design iterations where each improved control scheme would be tested for how efficient the test user could move around e.g. walking sideways and focus on one point and other test goals mentioned in **??**. This affected small amounts of test participants, even if they were told to ignore the text and other signs and just go through the test as fast as possible.

Other aspects that caused discussion was that some users had a problem with hitting the obstacle and loosing some time with every control scheme. This was because each scheme had its own little disadvantage in implementation that caused inaccurate user movement. This might cause some negative feedback in the card sorting as mentioned in evaluation. However this does not affect the comparison between the different schemes, since all the controls were calibrated to match each other, speed-wise and big a 3D area they require. reflabel

Iterative design process

The process that was used for this project was focused on information gathering and creating a product based on this information. Instead of focusing all the effort on finding research the design process could have been done more user centred. 2.2 This would mean creating low-fi prototypes and doing quick and dirty testing from the beginning. This would allow us to go through several iterations where we heighten the fidelity and eliminate errors while the focus of the project is being narrowed down with each iteration. Doing several iterations would have allowed us to catch technical issues like what we had with the joystick controls early on in the process, allowing for several tests with a decreasing amount of error.

Bibliography

[car, a]

[car, b] Card sorting. http://usabilitynet.org/tools/cardsorting.htm.

[per,] Performance testing. http://usabilitynet.org/tools/testing.htm.

[Ciotti, 2013] Ciotti, G. (2013). The psychology of colour in marketing and branding. http://www.helpscout.net/blog/psychology-of-color/.

[Csikszentmihalyi, 2005] Csikszentmihalyi, J. N. . M. (2005). The concept of flow. http://myweb.stedwards.edu/michaelo/2349/paper1/ConceptOfFlow.pdf.

[Developers,] Developers, A. Motion sensors. http://www.smashingmagazine.com/2010/11/04/best-practices-of-combining-typefaces.

[Examstutor.com, 2015] Examstutor.com (2015). Psychographic segment. http://www.examstutor.com/business/resources/studyroom/marketing/market_analysis/8_psychographic_segmentation.php.

[Greenfield,] Greenfield, S. Modern technology is changing the way our brains work. Daily Mail website. http://www.dailymail.co.uk/sciencetech/article-565207/Modern-technology-changing-way-brains-work-says-neuroscientist.html/,.

[Gube, 2010] Gube, J. (2010). What is user experience design? overview, tools and resources. Smashing Magazine. http://www.smashingmagazine.com/2010/10/05/what-is-user-experience-design-overview-tools-and-resources/.

[Nielsen, 2012] Nielsen, J. (2012). Usability 101: Introduction to usability. http://www.nngroup.com/articles/usability-101-introduction-to-usability/.

[Norman, 1988] Norman, D. (1988). *The Design of Everyday Things*. DOUBLEDAY.

[Preece et al., 2015] Preece, J., Sharp, H., and Rogers, Y. (2015). *Interaction Design: Beyond Human-Computer Interaction, 4th Edition*. Wiley.

[Prensky, 2001] Prensky, M. (2001). Digital immigrants , digital natives. http://www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf.

[Prensky, 2009] Prensky, M. (2009). From digital immigrants and digital natives to digital wisdom.

[Raskin, 1994] Raskin, J. (1994). Intuitive equals familiar. Technical report, Communications of the ACM.

[Robinson, 2013] Robinson, K. (2013). 3 keys to develop a faster, more intuitive user experience. Openview Labs. http://labs.openviewpartners.com/intuitive-user-experience-design/.

[Sardo, 2009] Sardo, G. (2009). Building mobile user experiences.

[Spool, 2005] Spool, J. M. (2005). What makes a design seem 'intuitive'? http://www.uie.com/articles/design_intuitive/.