Multiple Platform First

Design guidelines for multiple platform games

Emelie Andersson
Abstract

This thesis investigates the two gaming platforms PC and Console and how the interfaces of games on these platforms could be designed more efficiently making it easier to release games on multiple platforms. In other words, how could the Multiple Platform First method look. Little previous work exist on this problem so this thesis work gathers information from other industries and also research on user interfaces in games in general. By looking at games running on both platforms different best practises and common solutions were discovered. A study was conducted testing different in-game components on users. The components were selected to test if the users would accept non-traditional components since users detect when playing on an interface not intended for the platform. This makes the study very complicated since the "best" solution might not work if the users does not accept it for the intended platform. Concepts were designed to combine the testing of solutions with the users opinion of the solutions. The chosen concepts were researched both in literature and by looking at present implementations in games. To be able to user test the solutions they were iterated from low fidelity prototypes on paper to high-fidelity prototypes that were playable in Unity. The prototypes were tested on users and data gathered through Think Aloud comments and questionnaire answers. This study presents a first draft of how a multiple platform approach can be achieved.
I would first like to thank my external thesis advisor Linus Dilén at Paradox Interactive for his constant feedback and will to aid in the work, it was much appreciated to always have someone to turn to. Also all the "Good job" compliments really lifted my spirit, so thank you for being here during this whole process.

I would also like to thank my thesis advisor from the university Håkan Gulliksson for always having a kind word to say during our meetings. It lifted my spirit and made the report writing much more fun, thank you for that. And also a special thanks for the very fast e-mail replies when the panic took over.

A big thank you to all the staff at the Arctic studio for supporting my thesis work by cheering me on and participating in my user study. You are all awesome! A special thank you to my fellow thesis worker at Paradox, Olle, it was so nice to have someone in the same situation to talk to and get feedback from. Our thesis therapy sessions was much appreciated.

Then there was my peer-review group. What a relief to have you to discuss with during the hard times of this thesis. Your kind and supporting words along with your invaluable feedback has made this thesis what is it today.

Finally, I must express my very profound gratitude to my loved ones for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Emelie Andersson
Contents

1 Introduction
   1.1 Objective
   1.2 Paradox Interactive
   1.3 Scope

2 Method
   2.1 Iteration one
      2.1.1 General Literature study
      2.1.2 General Field study
      2.1.3 Definition of the problem
      2.1.4 Concept generation
      2.1.5 Low Fidelity prototypes
      2.1.6 Low Fidelity test
   2.2 Iteration two
      2.2.1 Concept literature and field study
      2.2.2 High Fidelity prototypes
      2.2.3 High Fidelity test
   2.3 Iteration three

3 User interfaces in games
   3.1 User interfaces
   3.2 Game genres
      3.2.1 Controls in different genres
      3.2.2 Strategy games
      3.2.3 Action Games
      3.2.4 Role-Playing Games
   3.3 The PC platform
      3.3.1 User interfaces for the PC platform
   3.4 The Console platform
      3.4.1 User interfaces for the Console platform
4 Design approaches for multiple platform games

4.1 Experience from other industries
4.2 Multiple platform games
  4.2.1 PC first
  4.2.2 Console first

5 Evaluation and inspection methods

5.1 Definition of heuristics
5.2 Heuristic Evaluation of Playability (HEP)
5.3 Principles of Game Playability (PLAY)
5.4 Game Approachability Principles (GAP)
5.5 Game Approachability Issue Definition (GAID)
5.6 Critique against the heuristic evaluation

6 Result

6.1 General Findings
6.2 Summary of previous work
  6.2.1 Components
  6.2.2 Complicated situations
6.3 Iteration one
  6.3.1 Concept generation
  6.3.2 Concept selection
6.4 Iteration two
  6.4.1 Component literature study
  6.4.2 Hi-fi prototypes
  6.4.3 Test environment
  6.4.4 Test group
  6.4.5 Cursor prototype test
  6.4.6 Menu prototype test
  6.4.7 Think Aloud comments from the test
6.5 Iteration three
  6.5.1 Heuristic Evaluation of Multiple Platform Issues (HEMPI)
  6.5.2 Test of HEMPI

7 Discussion

7.1 Result Discussion
7.2 Multiple Platform First
  7.2.1 Insights in Multiple Platform First 38
  7.2.2 Methods in Multiple Platform First 39

8 Conclusion 40

9 Future Work 42

References 42

A Interview Questions 47
  A.1 After each cursor prototype 47
  A.2 After both cursor prototypes 47
  A.3 After all radial menu prototypes 47
  A.4 After each menu prototype 48
  A.5 After both menu prototypes 48
<table>
<thead>
<tr>
<th><strong>Glossary</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Console</strong></td>
</tr>
<tr>
<td><strong>D-PAD</strong></td>
</tr>
<tr>
<td><strong>Hi-fi</strong></td>
</tr>
<tr>
<td><strong>HUD</strong></td>
</tr>
<tr>
<td><strong>Lo-fi</strong></td>
</tr>
<tr>
<td><strong>PC</strong></td>
</tr>
<tr>
<td><strong>UI</strong></td>
</tr>
<tr>
<td><strong>UX</strong></td>
</tr>
</tbody>
</table>
1 Introduction

Gaming is a common leisure activity for people worldwide. It refers to playing electronic games via Personal Computer (PC), Console or through some other medium. It can be performed alone or together with others and is often a source to relaxation and distraction. The principles of gaming has spread to other industries as well and games can today be used in, for example, education. Gaming is a nuanced term that could refer to both casual and hardcore gaming. Overall in 2018 there was 2,3 billion gamers worldwide [1]. There are so many different ways to game and there exists so many gamers all over the world. There is a demand for games for different platforms, in different genres, to play alone or with friends and games with an unique story and environment. A large industry of studios developing games has continuously grown with the demand for games. In fact the turnover of the gaming industry is estimated to be about over 180 billion US dollars in 2021 compared to 70 millions dollars in 2012 [2].

There are many unwritten standards a game has to follow, typically an ordinary game has to be fun or provide some sort of pleasure for the user. Digging deeper, each genre has some similarities that the user expects to find when playing a game in a certain genre. Besides these standards a game also has to follow the standards for the intended platform, a game for PC has to work in a similar way to other PC games.

Today the gamers exist all over the world and they use a variety of platforms. Computers, different console and mobile phones are the most common. Since the market is divided between these platforms increasingly more companies are exploring the opportunity to launch their games on several platforms, so called multiple platform games [3]. New technologies like the SteamLink and the Google Stadia challenge the very way Console and PC gaming is separated today and is leading the development towards platform independent gaming, which faces the same challenges as the multiple platform game User Interface (UI) design is dealing with today. Launching a game on multiple platforms expands the available market for the game but since the different platforms present different prerequisites it makes developing multiple platform games a challenge. It is a trade-off between the revenue from the multiple platforms and the cost to develop for several platforms. Today there are some unavoidable differences between platforms that will create work but could the UI design be done in a more effective way?

Traditionally a multiple platform game will follow one of two approaches, it will either design PC first meaning it will target the PC audience as its primary target or Console first targeting the console audience primarily. When attempting to design a common UI or creating them in a smarter way for several platforms it is important to have the intended platform in mind since the users seem to detect when they play on a UI not intended for the platform. Chunky interactions, readability issues, tactile feedback issues and navigation problems are just some of the situations which could occur and should be able to be avoided in multiple platform designed UI’s.
How could the *Multiple Platform First* approach look and what would it focus on? How can the problems with multiple platform interfaces be detected in a structured way?

### 1.1 Objective

The objective of this thesis is to research the design of UI’s for multiple platforms games. Furthermore to investigate guidelines on how to work around the trade-off between the cost of creating double UI’s and if only one interface could work for multiple platforms. The thesis aims to answer the following research questions:

- Can traditional evaluation and inspection methods be modified to identify and evaluate multiple platform issues?
- How affected is the playability of games when using traditionally console components on PC interfaces?
- How affected is the playability of games when using traditionally PC components on console interfaces?
- How could interfaces be designed to mitigate the difference between console and PC based on the result from this study?

### 1.2 Paradox Interactive

Paradox Interactive is a game development studio founded in 1999. Paradox is one of the leading publishers of strategy games for PC. The studio has released a bunch of successful games in this genre including Crusader Kings, Europa Universalis, Hearts of Iron and Cities:Skylines to mention a few. Today the company have more than 2 million active users each month and studios in Stockholm, Umeå, Malmö, Delfs and Seattle [4]. The development platform is primarily PC, but the company has released games on console and mobile platforms as well. Paradox is interested to see how the multiple platform development process could change from how it is done now and if there are some guidelines that could be implemented in the design process.

### 1.3 Scope

This thesis will limit the multiple platform game scope to games for PC and Console. For the consoles traditional hand controllers will be prioritized.
2 Method

The overall structure of the method in this thesis was inspired by Design Thinking 101 by Nielsen Norman Group [5]. This means that the method will follow the structure of: Empathize, define, ideate, prototype, test and implement. This is an iterative method, in this thesis the phases have been adjusted removing the implement phase and deciding to iterate twice over the chosen steps. In figure 1 the different phases and their connections is shown as used in this thesis.

**Figure 1:** Design Thinking 101. [5]

**Empathize:** The phase empathize aims to create an understanding of the field and foremost of the user. The tools needed to create this understanding was a literature study, field study of games and study of gaming related forums were the tools were used.

**Define:** Using the material from the empathize phase, different conclusions or further definition of the actual problem was drawn.

**Ideate:** In the ideation phase brainstorming was used to generate concepts that could be potential solutions

**Prototype:** The different concepts was prototyped in different stages. The low-fidelity prototypes were created with pen and paper and the high-fidelity prototype created in Unity.

**Test:** The prototypes were tested in different phases. The low-fidelity prototypes were tested in workshop sessions with colleagues while as the High Fidelity (Hi-fi) prototypes were tested with a usability test with potential end users.
2.1 Iteration one

To take a first stab at the multiple platform first method a literature review, a field study of games and a look over gaming forums was conducted. Which are the key differences between PC and Console UI? Which previous conclusions exist on UI design that is or can be applied to the multiple platform issue?

2.1.1 General Literature study

To create an understanding of the field both a literature study and an field study of games was conducted. The literature study aimed to find what research existed concerning multiple platform user interface design. The field study had as a goal to find existing strategies and also to understand the problem better. The literature study took off from sites such as:

- Google Scholar
- Research Gate
- Diva-Portal

The articles were selected based on their overlap with UI in games or with evaluation methods for games. The literature study had a goal to find previous research on multiple platform game interface design and/or research that could be applied to that field. In addition research on evaluation methods for games was included.

To find out what the gaming community thought on the subject another litterateur study was conducted in parallel. This study aimed to find both professional articles from discipline experts as well as thought from the gamers. This study took off in different forums such as:

- GamaSutra
- Saphirenation
- GiantBomb

The forums were included if the post or article concerned multiple platform game issues or opinions.

2.1.2 General Field study

A field study was conducted to look at some interesting games from a multiple platform perspective. This included both active multiple platform games together with PC and Console specific games. The games were chosen by insights from the forums and also by collecting knowledge from within Paradox. The study resulted in a list of games to look at as examples of different design ways.

- Europa Universalis IV
- StarCraft 2
• Halo 2
• Zelda: Breath of the Wild
• Fortnite
• The Sims 4
• Dishonored 2
• Dark Souls

2.1.3 Definition of the problem

Based on the previous research and the understanding of the problem it is very unlikely that there exist any one answer to how multiple platform UI’s should be designed generally. A hypothesis was constructed that general guidelines still could exist on a lower level. When digging deeper and looking at individual parts of the UI it is more likely that conclusions will be able to be drawn.

2.1.4 Concept generation

After deciding on which level to investigate on, a concept generation was conducted with a brainstorming session which took off from the literature and field study. The brainstorming session was conducted alone. The layout of the session was inspired by McCarthy’s article concerning brainstorming alone [6]. Instead of post-its, as suggested in the article, a white board was used to write the ideas on. Instead of doing two iterations the session went on until one minute had passed without any new ideas had been written. After the brainstorming the ideas were grouped together and the nonsense ideas thrown away. This method writes solutions on a very basic level with simple sketches or words which is cohesive with Nielsen and Normans suggestions for the ideate phase [5].

2.1.5 Low Fidelity prototypes

The fidelity of a prototype refers to how it conveys feeling of the final product. It is about visual design, content and interactivity. When creating a Low Fidelity (Lo-fi) prototype only a few visual attributes are presented such as shapes and visual hierarchy. The content is very restricted and only the key elements of content is included. Even a Lo-fi prototype should have some interactivity were a simple simulation can be preformed by a human [7]. These prototypes were very simple sketches with the different stages sketched to achieve a simple simulation of the function. The layout and content were kept at a minimum level to convey the situation and functionality.

2.1.6 Low Fidelity test

The Lo-fi prototypes were tested through workshops with one User Experience (UX) designer and one Game Designer from Paradox. The prototypes were presented for the same amount of time, then discussed and finally a voting took place selecting the top ideas. Which ideas were selected as top depended on the situation which is why the discussion is important to agree on how to evaluate the ideas.
2.2 Iteration two

After selecting the two final concepts to be prototyped in to hi-fi prototypes a new iteration of the method started.

2.2.1 Concept literature and field study

The aim of the concept literature study was to find which previous research existed in just those fields and which games standards, if any, existed. The study was done in the same manner as in the previous iteration.

2.2.2 High Fidelity prototypes

Taking the chosen concepts further a Hi-fi prototype was created. The Hi-fi prototype should be more detailed and closer to the end product. The interactions and design should be realistic and real content used [7]. In this case the the focus was on the interactions and less focus on the exact content and design. This means that there could be arguments that this prototype is mid-fidelity which means it is somewhere between Hi-fi and Lo-fi but it will be referred to as a Hi-fi prototype in this thesis.

The Hi-fi prototypes were created in Unity in an A/B test manner. An A/B test is a method for comparing two versions of a design or function against each other. The A/B test was chosen since it is very good at deciding which version is the most successful one. On UXPlanet it is recommended even in a pre-launch state which is more similar to this situation [8]. The layout of the prototypes were decided into a city with clickable houses since it was appropriate in both situations for both the concepts.

2.2.3 High Fidelity test

As suggested by Nielsen and Norman the prototype was put in front of real users to verify it achieved its goal. Based on the research by Nielsen and the graph of issues discovered versus the number or participants the number of users that got to participate in the tests of the Hi-fi prototypes was set to ten users [9]. To add additional value to the tests the method of Think Aloud was applied. According to Holzinger this method only needs a minimum of three users [10].

The usability tests was performed with Think Aloud since it doesn’t require as many users to give results [10] in order to get some comparable data this will be completed with a questionnaire but there will not be enough users to draw a significant conclusion. Think Aloud means that the user in encouraged to think loud during the test and say what they are feeling and experiencing. Using Think Aloud gives access to how different people experience the same task which is generally a very qualitative result. Think aloud can give very varied result which is why triangulation with other methods is suggested by Charters [11]. The test was a situation based test were the user got to perform different tasks which were designed to expose the participant to the thing it was supposed to test. Each participant got to try both versions of both prototypes but the order was randomized between participants to minimize any eventual bias between the order of the prototypes. The prototypes were tested on ten participants at the office of Paradox Interactive. The participant
got to fill out a short questionnaire about experience with gaming before starting the tests.

An example of a task was:

- Navigate around in the world and test the controls, try selecting at least one house from each part in the city

After completing the tasks on one prototype a short questionnaire was filled out by the participant. After completing testing both prototypes a comparative questionnaire was filled out.

### 2.3 Iteration three

Based on the general literature study different evaluation methods for games had been discovered. One was chosen as a predecessor to a new concept. Taking off in the research on existing methods a new version was developed. It modified and removed points from the previous version trying to adjust it to identify multiple platform issues. The evaluation method was hypothetically tested on games in different stages to evaluate how it potentially could work in reality. This test meant going over the points in the evaluation with the constructed Unity prototypes as a source.
3 User interfaces in games

User interfaces for multiple platform games are complicated and is affected by the standards correlating with the genre and platform the game is intended for. The individual gameplay of each game also affects which possibilities exist when wanting to design the UI. This chapter will look into UI for both the PC and Console platform. Which are the most important platform standards to consider? Furthermore some of the bigger genres will be explained with their corresponding standards.

3.1 User interfaces

User interface is according to definition the means in which a person controls a software application or hardware device [12]. In games the user interface is the means in which the player experiences the game. In the word of Desi Quintans:

Put simply, the role of a good UI is to provide relevant information clearly and quickly and to get out of the way once it has done its job.

[13]

User interfaces for games is not a pure science with only proved concepts and rules but a combination of art and the science by the lessons learned. Even though it is not a science, some people have done work trying to categorize the elements in the UI to draw conclusions about how they are designed. A categorization of the elements in a UI into four categories is suggested by Lorentzon and Fagerholt [14]. UI elements are categorized into the four categories: Diegetic, Spatial, Meta Non-diegetic. Diegetic elements can be seen by the players avatar in the game. Information is provided in a way that tries to make the player think that the avatar himself is viewing that very same information, is diegetic. Spatial UI elements tries to keep the player immersed in the game as much as possible, even though the information they provide is clearly for the player’s eyes only. An example of this is in tutorials fine spatial elements tries to fool the player to think the avatar is receiving the training even though the text and directions are not visible for the avatar. Meta elements are closely related to spatial ones, but with the distinct different of sitting on a clearly different plane from the game world. Generally meta elements sit on a 2D plane above the game screen and provide player information on what is going on, for example blood spatters on the screen when the player is hit. The meta elements tries to keep in line with the visuals and aesthetics. Non-diegetic element are there to purely provide usable information, visualization and clear controls for the players. They sit on a 2D plane above everything else like meta elements but they also usually clearly stand out from the main view. In figure 2 the four different categories for UI elements are presented.
Turunen conducted a study of UI’s in video games with the conclusion that the most important lessons was: Customization options, Genre consistency, Visual appearance, Balance between efficiency and effects [15]. When it comes to looking at common flaws in game UI Yordanov also contributed with his post on Sapphire-Nation listing common mistakes. His list and Turunens findings overlap on many points. Yordanov lists slow to navigate UI and game immersion which is not mentioned in Turnunes study [16].

When looking into designing a control scheme for a game one has to take the comfort of the user into account. Dotsenko [17] writes from his experience as a designer in game control design. Both common PC and common console controls today depend on the use of hands. Human hands have limitations which makes controls using the thumb and index finger more comfortable than those using the pinkie. With this in mind Dotsenko suggest "Accessibility Tiers" for both standard PC and console controls. These have arranged the different buttons on the console and keys on the keyboard into tiers based on how comfortable and accessible the button/key is, this is shown in figure 3.
3.2 Game genres

Since there exists genre standards when it comes to the UI of a game it is interesting to look into some of the most common genres. Paradox Interactive is most known for the Grand Strategy genre. There could be some things used in other genres that could be applied to this genre or vice versa. Some genres are equally represented on each platform and some are favoured on one platform. To follow the genre standards it is important to make the game accessible to users.

3.2.1 Controls in different genres

On PC it is the keyboard and mouse that is the standard input control. These can be used in many different setups to allow different interactions. Some of these are more common in certain genres. Moving a character or the view is very common and is a key control in almost every game. On a keyboard there are a few common ways to move. One is to use the WASD-keys to move forward, left, backward and right. This can be utilized both for camera and for moving character, in third person and in first person view. The arrow-keys can be used in the same way as the WASD-keys. It is also common to use the mouse to move. One example of this is when the background moves when the cursor hits the edge of the screen, edge scrolling. One other way is to drag the background by holding down a key and then moving the background by dragging the mouse. The mouse can also be used to rotate the camera or a character. The mouse is often used as a cursor allowing for components like psychological buttons in the UI and more information heavy menus. When playing console games with a controller, the controller is your only interaction method. Common practise is to use the left joystick to move and the right joystick to look around. The D-PAD can also be used for movement but is more common to use to navigate menus. The bumpers are also a common navigation tool in menus.

3.2.2 Strategy games

With gameplay based on traditional strategy board games, strategy games tend to give players a godlike access to the world and its resources. These games require players to use carefully developed strategy and tactics to overcome challenges. More recently, these type of games have moved from turn-based systems to real-time gameplay in response to player feedback. Some subgenres include: 4X, Artillery, Real-time strategy (RTS), Real-time tactics (RTT), Multiplayer online battle arena (MOBA), Turn-based strategy (TBS), Turn-based tactics (TBT), Grand strategy wargames [18]. Strategy games are more common on PC, especially in the TBT, TBS, RTS and Grand strategy subgenres. In strategy games on PC it is common to have a lot of information easy accessible and often in many menus. In console menus the navigation is a little trickier due to the lack of the cursor. The bumpers and sometimes the use of a button could help to make navigation in menus possible.

3.2.3 Action Games

Action games are just that—games where the player is in control of and at the center of the action, which is mainly comprised of physical challenges players must overcome. Most early video games like Donkey Kong and Galaga fall into the action category. Action games have many sub genres including Shooters and Survivals
Shooters let players use weapons to engage in the action, with the goal usually being to take out enemies or opposing players. Shooters are categorized by the player perspective: First-person shooters (FPS) are played from the main character’s viewpoint; Call of Duty, Half-Life, and Halo are good examples. With third-person shooters like Fortnite and Splatoon, the action takes place from a viewpoint where the player can see the main character, usually from slightly above and behind. Survival action games have really come into their own over the past few years. The survival horror game Resident Evil was one of the earliest (though a linear game), while more modern survival games like Fortnite take place in open-world game environments and give players access to resources to craft tools, weapons, and shelter to survive as long as possible. In FPS it is common practise to use the WASD-keys to move and then use the mouse to rotate and aim, left click to shoot. In console the left stick handles movement, the right joystick handles rotation and aim and A-button or triggers is clicked to shoot.

### 3.2.4 Role-Playing Games

Probably the second-most popular game genre, role-playing games, or RPGs, mostly feature medieval or fantasy settings. This is due mainly to the origin of the genre, which can be traced back to Dungeons and Dragons and other pen and paper role-playing games. Still, hardcore RPGers don’t discount sci-fi fantasy-themed RPGs like Mass Effect, Fallout, and Final Fantasy, which have helped put unique spins on the genre. RPG also have a bunch of subgenres including: Action RPG, MMORPG, Tactical RPG, Sandbox RPG. RPG’s exist on both console and PC and with both first person and third person perspective. The common controls is to move the character with WASD or arrow keys or by moving the character by clicking with the mouse on the new location on PC. On console the joysticks usually do this but in some cases the D-pad could be used. On PC the mouse is clicked to select stuff while on console the A-button does this.

### 3.3 The PC platform

The PC as a platform means that the user uses a computer to play games. This means that traditionally the user is close to the screen and interacts with the game using a keyboard and mouse. The keyboard gives access to many keys that can be used in different ways in the game and the mouse provides accurate and fast navigation over the screen. The computer today is used for so many different activities and gaming is one of them. The computer started as an advanced calculator and has from there evolved to the computers we have today. Overmars describes the major milestones in computer game development in his text “A Brief History of Computer Games” [19]. One of the first usages of a computer as a gaming platform was the game Space War that was run on a PDP1 computer. Up until the 90’s the PC gaming was very limited. By the 90’s the computers were getting more powerful. PC had better resolution, more memory, hard disks to store game data and a much higher processor speed. Many games were released during this period in time including: Sim City, Grand Theft Auto and Half-Life to mention a few. Over the years the hardware of the computer became better and allowed for more advanced
graphics and more powerful games. The change in hardware, mostly the ability to store large files opened up to saving game and creating recurrent games that didn’t really end. As the development of computers went on the market extended beyond PC and today there are other type of computers as well. The most popular non-PC computer is the Apple Mac. When it comes to gaming, PC computers are still the most common computer to use.

3.3.1 User interfaces for the PC platform

The PC platform and its controls allows for rather information heavy and menu heavy UI. Since the keyboard has very many keys, hotkeys in a common usage to allow easy access to information in PC UIs. In PC UI it is also common for the user to be able to "drag and drop" windows, which means that the user can move the windows, often menus to a location on the screen of their choosing. Hovering items with a cursor to get more information, in the form of tool tips is very common. The general layout is often compact since the elements in the UI are usually close to each other. One example of a menu and information heavy PC UI is Europa Universalis IV [20], shown in figure 4.

![Figure 4: Screenshot of Europa Universalis IV UI](image)
Europa Universalis has a map where one clicks to move the people which are symbolising armies. In every menu there are a lot of information and clickable stuff to get deeper into the menu. Tool tips arrive on hover. This type of UI uses the mouse mobility and ability to hover over and click in menu, the user can move the mouse freely and move to the desired area of the screen easily. It also uses the fact that the user is sitting close to the computer. On a TV screen this much and small text would be unreadable. Europa Universalis IV is an example of a Grand Strategy game.

An example with the usage of hotkeys on PC is StarCraft 2, seen in figure 5.

![Figure 5: Screenshot of StarCraft 2 UI. [21]](image)

StarCraft 2 is an example of a real time strategy game (RTS). The menus are actually pretty deep but with usage of hotkeys and the mouse it is possible to play this very intense game. Generally FPS games have much more minimalistic UIs and are less information heavy.

To summarize some of the main points on PC UI:

- Keyboard allows for many hotkeys
- Closeness to screen allows for a heavy amount of text
- Mouse cursor makes movement on the screen easy
- Standard for tooltip is on hover location
- Windows can easily be dragged and dropped with mouse
- Buttons are a common element

### 3.4 The Console platform

The Console platform was in the beginning only intended for gaming and has in its later days been developed to support other activities as watching Netflix on your Xbox for example. This is almost the opposite from the computer that was originally
developed for other usages before it became a gaming device. The first consoles or “video game consoles” appeared in the 70’s. The Magnavox Odyssey was released in 1972. Then a rapid development started with new improved consoles releasing continuously. Consoles have so far existed in eight generation, with the Magnavox Odyssey being part of the first generation. In the second generation we find the Atari 2600 mostly known for the game Pac-Man. In the third generation Nintendo’s NES was released in 1983-1986. In the late 80’s more actors came into the market and Nintendo’s Super NES got competition from Genesis, both with over 40 million units sold. In the fifth generation the Playstation became the most popular console, released in 1994-1995 it sold over 100 million units compared to the Nintendo 64 which only sold 32 million units. In the sixth generation the Playstation 2 was the bestseller with 155 million units sold, the best selling console in history. In his generation the first Xbox was released. The seventh generation started with the release of the Xbox 360 and followed by the release of PlayStation 3 and the Nintendo Wii. The Nintendo Wii was the best selling console of the seventh generation. The eighth generation which is still ongoing contains consoles like: Wii U, Playstation 4, Xbox One and Nintendo Switch [22].

The controls for the consoles are different depending on which console is being used. Starting from one analog stick with only one button, to today’s controllers with many more buttons and options. There is a large variety in the controllers of today, for example the Wii and Wii U have a very different controller from PlayStation and Xbox. The common denominator is that all controllers is a combination of buttons and analog sticks, but some consoles have motion sensors and/or touchscreens. According to Techspot [23] the Xbox 360 controller is the most common used by Steam players and also the most liked among players. A picture of this controller can be found in figure 6.

![Figure 6: Xbox 360 controller](image)

### 3.4.1 User interfaces for the Console platform

The console platform and its controls has joysticks which makes rotation movement very natural. The console controller has fewer buttons than the keyboard which creates some limitations. The D-pad combined with the joysticks gives many navigation options. In console UI it is common to have very few buttons and very little information presented at once. This often create more airy interfaces. In menus it is common to let the user step through it with joystick or the D-pad. Hovering is not as common in console UI as in PC UI so tool tips for example could appear next to the menu or in a designated area. Due to the stepping in menus more
grid layouts are found in console games. Button hints are common to open different menus in different views. In console interfaces haptic feedback in the form of vibrations are used to enhance the experience and also to clearly communicate with the user the effect of their actions. The effect of actions are also sometimes communicated with more animations or rather more powerful animations.

An example of a very simplistic in-game UI is Halo 2. It has very few elements on the screen and they are pretty sparse. There are basically no in-game menus in Halo, to switch weapons you throw away your current for the new one. The Halo 2 UI is shown in figure 7.

![Halo 2 UI](image)

**Figure 7:** Screenshot of Halo 2 in game interface. [24]

There are more menu heavy games on console as well. One is the very popular Zelda franchise. In the game Breath of the Wild menus plays a big role in the gameplay. When you find equipment, food or weapons you have to go to the inventory to choose them. The inventory is also limited which means you may have to go in and throw something old out to fit the new equipment or food. When changing weapon there is a smaller in-game menu. The in-game UI and menu UI are shown in figure 8.
To summarize some of the main points on Console UI:

- Fewer buttons to work with
- Readability more challenging due to screen distance or screen size
- Joysticks making rotating movement very natural
- More haptic feedback
- Button combinations to reach some elements
- More animations
4 Design approaches for multiple platform games

In this chapter the different routes a multiple platform game can take are sorted out as well as the two main approaches when designing games today. Generally a game is either PC first or Console first.

4.1 Experience from other industries

In a similar way as a game is either developed PC first or Console first website design has a similar way, a website can be Mobile first or Desktop first. Desktop first is the traditional way where a website is designed to work for desktop and then scaled down to work on a mobile device. Some aspects and features are optimized for desktop websites and do not scale down very well. In the mobile first approach the design for this smaller more limited screen is done first and then scaled up. Both these strategies are equally valid but have different advantages. Gonzales lists the pros and cons with the mobile first strategy in his article. The main pros include that content is prioritized and creates a standardized experience over platforms. The downsides is according to Gonzalo mostly the cost of this way of development that it generally requires a larger budget [26].

4.2 Multiple platform games

There are many examples of games that today run on multiple platforms. Some are developed for multiple platform from the start and some games are ported. This means that they were first released on one platform and then redesigned to work on the other platform. Porting a game provides many challenges, not only the design challenge to make a comfortable UI. The hardware in each console is different and generally doesn’t have the same power as the computer. This makes it a risk that ports can have worse performance than a game intended for the platform. When looking into ports it is the UI design issues that are interesting in this context but it is not the only challenge with ports and multiple platform games.

4.2.1 PC first

PC first is a design methodology in game design targeting the PC users as the primary players of the game. One example of such a game is Fortnite. The menu system is designed with the direct input as a priority. This does however not mean that it has to be a bad console UI just because it is PC first. Sinclair [27] analyzes some UI’s in his article on Gamasutra. He is hesitant to call Fortnite PC first and
instead he wants to say it is PC leaning since it is still good and playable on console. He identifies some solutions to make the UI work on console, one is to augment the existing array of buttons peppered around the screen with icon prompts. One critique is that the button prompts are spread all over the screen making the player have to scan the entire screen. All in all this port works fine, the UI is shown in figure 9.

(a) PC UI

(b) Console UI

Figure 9: Screenshot of Fortnite menu UI. [28]
Fortnite has a well working UI on console even though the design is PC-leaning. An example of a PC first approach on console that is not working as well is the console version of The Sims 4. This is an example of a console game with an extreme PC first approach. The console game is basically a copy of the PC game which creates a bunch of control issues when playing on console. The Sims 4 has a mouse-style cursor controlled by the left joystick while the right controls the camera. The Sims 4 depends on context switching for the user to be able to move around in the buttons around the screen. The console UI of the Sims 4, the same as the PC UI is shown in figure 10. The Sims 4 demonstrates the most of the negative effects a PC first game can have on console.

![Figure 10: Screenshot of The Sims 4 UI.](image)

### 4.2.2 Console first

Console first targets the console users as the primary players of the game. All console exclusives are examples of this methodology but some ports also uses this concept as well. Sinclair describes his motivation to why Dishonored 2 is an example of the console first method. The UI looks identical apart from the button hints at the bottom of the screen. The main giveaway that this is a console-first design is in the favored interaction method-specific button/key presses on both platforms, where a PC first game would emphasize direct input with the mouse, thus leaving the player with only the two mouse buttons to worry about. The downside of this approach is according to Sinclair the UX burden that is put on the PC player to read over a bunch of key cues instead of encouraging a more natural direct input. The PC UI is shown in figure 11.
Dishonored 2 has a well working UI on PC even though the design is console first. An example of a console first approach on PC that is not working as well is Dark Souls. This game was ported to PC more or less directly without changing the UI. The largest problem with the PC port though was the resolution and overall quality.
5 Evaluation and inspection methods

When looking to evaluate and/or investigate UI’s there are several methods to try. Jeffreis compares four different approaches in his article. The heuristic evaluation finds the highest amount of problems according to the studies conducted by Jeffreis. Therefore heuristics will be investigated as a inspection method in this thesis.

5.1 Definition of heuristics

Heuristics is any approach to problem solving or self-discovery that employs a practical method, not guaranteed to be optimal, perfect, logical, or rational, but instead sufficient for reaching an immediate goal. Nielsen developed a list of heuristics aimed to evaluate software. These heuristics have been widely acknowledged as a useful tool in software development. When looking into video games, they have some things in common with software in general but is still a different field. This means that the heuristics developed by Nielsen is not directly applicable on games. Instead a further development of heuristics, called game heuristics has emerged.

5.2 Heuristic Evaluation of Playability (HEP)

The evaluation method Heuristic Evaluation of Playability (HEP) is introduced in an article published by Desurvire et al in 2004. The article is based on previous research on game heuristics by Federoff and heuristics developed by Nielsen. Guidelines developed by Malone and Clanton cover some parts of the HEP. Desurvire presents four categories of game heuristics: game play, game story, game mechanics and game usability. The study conducted aimed to compile a comprehensive list of the heuristics and verify them since this had not been done at the time. A prototype consisting of screen shots from a game in development was used for the HEP. The HEP guidelines were concluded from existing literature and reviewed by experts prior to the test. In order to get a result for how the method performs a standard user test was performed on the same prototype. The issues found by both methods were compared and analysed. The conclusion was that the HEP proved effective in uncovering playability issues. The number of issues found by the HEP was greater than the ones found by the user study. The user study issues were more specific to the interface and the issues from the HEP more specific to the game. To note is that the prototype was from the beginning of a process and thus the game play and the game mechanics was not fully developed. The article written by Desurvire et al points out the use of HEP as a tool to predict some of the issues from a regular user test. The HEP should be considered a complement to regular user studies.
5.3 Principles of Game Playability (PLAY)

Game design is becoming a critical area in the field of User-Centered design. HEP has been found useful but only in limited circumstances. Principles of Game Playability (PLAY) [37] is a broad list of Heuristics developed to be of use as a generalized foundation that could then be modified for each specific game. This set of heuristics are developed especially to help developers during the entire design process. The PLAY principles were created by using current and past research on game usability Heuristics and design principles as well as the most up to date information on superior game design from working top-level game designers. Unlike HEP [33], PLAY recognizes game design as an art and a science [37].

5.4 Game Approachability Principles (GAP)

Desurvire and Wiberg present an extension to their previous work on heuristic evaluation including how to make games accessible to inexperienced players. The heuristics are intended to support usability testing by focusing on issues of accessibility within a game when evaluating and then doing usability testing as a checklist to refine the design. In addition, and most importantly, the accessibility principles can be used to design a good tutorial from the onset of game design. In many cases the lack of accessibility results in the failure of a game resulting occasionally in failure of studio that would otherwise have produced good and successful games. The principles in the GAP list were developed from previous research and based on current literature in relation to learning. Usability and playability evaluation was performed using the heuristic evaluation, focusing on how each accessibility heuristic was supported or violated and then defined the issue [38].

5.5 Game Approachability Issue Definition (GAID)

This article defines the key concept of the methods HEP, PLAY, GAP and later GAID [39]. These principles are: Usability, Accessibility, Approachability and Playability. GAID is an attempt to design a flowchart model based on the HEP, PLAY and GAP models. The goal of the flowchart is to be able to identify which one of six commonly encountered problems in game design that a given problem is most likely to be connected to. GAID is meant to be used as a tool to ease usability studies by taking the errors found in such studies and then identify the problem type. GAID will however not, at present, be able to give the designer any guidance in how to solve the problem.

5.6 Critique against the heuristic evaluation

The unstructured nature of heuristic evaluation has also led to several criticisms of the technique. For example, Cockton and Woolrych [40] point out that heuristic evaluation does not encourage people to take a comprehensive view of how software will be used, and that analysts must pick sample tasks and system features at random. Cockton and Woolrych also indicate that using heuristic evaluation,
and discount methods in general, can cause evaluators to miss problems and to identify false problems. Cockton and Woolrych conclude that discount methods are best suited to use in early design iterations rather than as a summative evaluation technique.
6 Result

Below the results from the study will be presented.

6.1 General Findings

When looking into multiple platform games and UI’s for these kinds of games it was quickly realized that no general rules applicable for all games on a high level considering the whole UI could be found. Since each game has so many dependencies from being a game in general, from genre standards, platform standard and the particular games own circumstances it would not work with general UI guidelines. Guidelines that can be used generally needs to be on a deeper level. After this conclusion a prototyping work began to create a proof of concept for some deep level general guidelines for a component level.

A component in general is defined as a part of the whole [41]. In this thesis a component is a part of the UI. To make sense components are organized into categories concerning different parts of the UI. Some of the main categories are:

- Navigation
- Menus
- Heads Up Display (HUD)
- Windows

From these an example of a component from each category is:

- Navigation - Cursor
- Menus - Radial menu
- HUD - Physical Buttons
- Windows - Dockable windows

6.2 Summary of previous work

Based on the games brought up in the theory a summary was done of the situations creating problems when wanting to design a good interface for both platforms. From existing multiple platform games some interesting components were discovered that could be used to prove that some conclusions on UI could be drawn on a component level.
6.2.1 Components

With the definition of components created after the litterateur study different components were generated with a brainstorming session described in the method section. The generated components are presented below.

- **Navigation - Tab navigation**
  Left and right button hints around top menu row on console

- **Navigation - Cursor**
  Cursor on console

- **Menu - Radial menus**
  Radial menus on PC

- **Button - Hotkey**
  Press button to access versus mouse click on PC

- **Navigation - Menu Navigation**
  Button to reach level in menu console

- **Menu - Grid layout in menus**
  More spacious layout of items, creates difficult navigation with D-PAD

6.2.2 Complicated situations

Based on the field study of games different situations were discovered which each presented an interesting problem. These situations are presented below.

1. Information heavy menus (Console)
2. Multiple windows (Console)
3. Too many layers in menu (PC)
4. Too many clicks for an action (PC and Console)
5. Navigation in world and in menu (Console)
6. Precision of 3D selections (Console)
7. Few available buttons (Console)
8. Context switching (PC and Console)

6.3 Iteration one

Based on the general findings the first iteration of the method the first concepts on components to test were created.
6.3.1 Concept generation

The result from the concept generation was suggestions of different components to investigate. Each component case was connected to one or several problematic situations. These are the Lo-fi prototypes ideas which very simply gives an explanation of the interactions and design of each concept. The list of components/cases are shown below.

- Cursor on console (5,6)
- D-Pad to reach HUD (2,4,5,7)
- Radial menu (1,3,4)
- Cursor in menu (1,4,5)
- Dockable windows Console (2,5,7)

After the concepts were generated they were sketched and further explained as a very low fidelity prototype.
6.3.2 Concept selection

A group with a game designer and one UX-designer got to vote on which cases they thought would be the most relevant ones to take to a prototype state. From the Lo-fi prototypes each case was discussed and valued. Based on this workshop the prototypes that will be further investigated on is Cursor on console and Radial menu on PC. These concepts will be taken in to high-fidelity prototypes.
6.4 Iteration two

To be able to investigate the deeper level approach some components had to be selected and prototyped to be able to test them on users. Based on the resulting list from the field study of games some cases were created, prototyped and tested with the goal to prove that some general guidelines could be drawn.

6.4.1 Component literature study

When selecting concepts a more specific literature study was conducted looking at what had been done previously with these specific components both in research and in already released games.

Radial Menus

Radial menus is a common menu type in console games especially in genres with traditionally heavy menus such as strategy or RPG. The concept originated from console RPGs, introduced by the Mana series, beginning with Secret of Mana (1993). The radial menu, used in conjunction with the analogue stick, allows quick access to a large number of menu options with a small number of button presses. Both left and right analogue sticks are used in conjunction with menus in different games but the left stick is more commonly used. Radial menus have often been used as a solution to the problem of porting PC-based genres to consoles, with prominent examples including Halo Wars, Command and Conquer: Red Alert 3, Tom Clancy’s EndWar, and Stormrise.

Samp conducted a study on different radial menus. He also did a comparative study between a list menu and a radial menu to measure the time it takes to search in the different menus. In his study comparing layered radial menus and layered list menus with his specific layout the radial menu was faster and more accurate throughout the test. Samp and Decker investigated how to populate layers in radial menus and how this affected the search time. Murano conducted a comparative study between a one layered linear and radial menu. The radial, in this article called pie menu scored higher in how pleasant the felt the appearance of the menu was, the comfort using the menu and in ease of search.

Free cursor

When looking into PC games most of them include a cursor controlled by the mouse in some way, this is a very standard interaction in a PC game. On console there is no mouse but analogue sticks that could be utilized to control a cursor. Cursor on console has been tried in the past which has created some strong reactions from console players of the opinion that cursor belong in PC games and not in Console games. On Reddit one user says that the cursor in Call of Duty: Black Ops, It is usually not a great experience.

One game that really put in an effort to go away from this truth that cursor is not for console is Destiny 2. In his GDC talk, David Candland explains the thought behind the cursor in Destiny 2. Candland et al have designed a cursor that is pleasant to use thanks to gravity and back scrolling. The idea behind this is to be able to design more freely and not be constricted to the classical grid layout in every screen for console games. When having a cursor that can move around easily over the screen elements can be placed more spacious which is an advantage for console games due to the distance between the player and the screen. Even though Des-
tiny invested into making the cursor work it was not well received by all. Some are very strongly into the opinion that cursors being on PC and not Console. Barker[47] writes in his article:

\[ \text{It is time to kill the cursor in console games - Sammy Barker [47]} \]

Cursors have many advantage even on console but is still generally criticized by console players.

### 6.4.2 Hi-fi prototypes

Two prototypes were created in Unity, version 3.4.1, each with two test cases. The cursor prototype tried a traditional PC component on Console. Cursor navigation is the standard way on PC and this prototype tries to take this to console in two different cases: A free and fixed cursor case. The free cursor is essentially a mouse cursor controlled with the left joystick of the controller with edge scrolling to move the screen. The fixed cursor is a centered cursor which remains centered all the time. Here the left stick controls the camera instead moving around the center of the screen. The Menu prototype tries a radial menu, more common in console games, on PC and compares it with the traditional list approach. The radial menu was implemented with three different navigation’s and the user got to try all three and then select a favourite used in the comparison test with the list. The prototypes are shown in figure 13.

![Figure 13: Hi-fi prototypes in Unity](image)

(a) Cursor

(b) List menu

(c) Radial menu
6.4.3 Test environment

The environment in Unity was built as a city with different houses and lands. Each house and land was clickable. The control used in the cursor prototype was a Xbox controller. In the cursor variant of the setting there were several pods, clusters of houses and each of the houses turned red when hovered by a cursor and showed a text for a few seconds when clicked to confirm the click. The setting was identical for both the fixed and free cursor test. In the menu prototype there was only one cluster of houses. Now a mouse and a keyboard was used as navigation tools and a menu was opened by right clicking on a house, no hover effect was present in this prototype.

6.4.4 Test group

The test group was between 20-69 years old with a gender distribution of 60 percent men and 40 percent women. In the group 60 percent dedicated over 10 hours to gaming each week. Everyone but one in the group had PC as the primary gaming device but more than half played console games occasionally. The group consisted of ten people.

6.4.5 Cursor prototype test

The cursor test gave a unanimous result, everyone who tested the two prototypes preferred the fixed cursor. When looking at the comments during the test and the scale answers from the questionnaire there are some interesting findings here besides the answer which cursor was best.

In figure 14, the summary of the participants rating of the experience of the navigation is shown. The scale was from 1 (Pleasant) to 7 (Frustrating).

![Figure 14: Experience with free cursor from 1 (Pleasant) to 7 (Frustrating)](image)

There was a strong correlation between the experience with console and struggle with the free cursor. The more experienced console gamers had less problems with the navigation. Most of the participants didn’t find the navigation hard but frustrating. When comparing the ratings of the tasks, which were rated from Very Easy to Very Difficult there was less difference than when comparing rating with the Pleasant to Frustrating scale. This was also caught during the test with exponentially more sighs and frustrating sounds during the free cursor test. The hardest thing in the free cursor test was to select the moving object, this created a frustration with almost all the users. The other thing was the edge scrolling which confused
some participants. When selecting still objects in center screen the free cursor was more appreciated. There were also some comments on the versatile of the cursor since it could work like a mouse on PC, which was the intention. Many of the participant mentioned improvements that would make the free cursor better. The most common suggestion was gravity toward the objects. During the test a few measurements were taken. The average time for selecting the moving object were 9.97 seconds with an click accuracy of 41.4 percent while the average time to select all buildings were 35.6 seconds with a click accuracy of 80.3 percent.

In figure 15 the summary of the participants rating of the experience of the navigation is shown. The scale was from 1 (Pleasant) to 7 (Frustrating).

The fixed cursor had a quicker start-up with all the participants, and the time it took before they got used to the navigation was considerably shorter. The hardest assignment, to catch the moving object, was still considered hard but the average rating was lower than the free cursor test. The moving around was considered easier than in the free cursor case. Several people commented that they felt they got more feedback to the movement of the joystick in this prototype. They liked it when the had to navigate to areas "outside the screen". During the test a few measurements were taken. The average time for selecting the moving object were 5 seconds with an click accuracy of 58 percent while the average time to select all buildings were 28.78 seconds with a click accuracy of 87.9 percent. When comparing experience between the free and fixed cursor statistically there is a significant difference in the opinion since the p-value is lower than 0.001 [48], it was calculated to 0.00000878. When combining this number with the actual opinions it can be interpreted to that the fixed cursor had a significantly better experience than the free cursor.

### 6.4.6 Menu prototype test

The menu test had two results, which radial navigation was preferred and then if a list or that radial menu was preferred by the participants.

The participants were not in agreement of the navigation within the radial menu. The fixed navigation was considered "consolly" which was disliked by many. Some felt that the limitations with the fixed navigation made the navigation very clear and easy to understand. One suggestion that came from many participants was to have a fixed navigation but to be able to drag the mouse over the circle and end up at that part in the menu. A type of shortcut navigation. The motivation for choosing the free cursor was that it felt very natural on PC to be able to reach the whole
screen. Some pointed out that this however could create confusion if you are in a menu and still is able to navigate the rest of the screen. The limited pointer was equally appreciated as the free pointer, which is showed in figure 16.

![Pie chart showing preferred radial menu navigation]

**Figure 16:** Preferred radial menu navigation
When the users got to compare the radial with a list most chose the list as the preferred menu type. When comparing the individual results from the list and the radial test the participants thought both menus were easy to use but when it came to a selection the list was still preferred. There were many interesting comments about what is important in a menu that the radial was more fun and had many more graphical and design options. The list was considered more efficient and many commented that this was also the most common menu type on PC so it felt more comfortable. The distribution of the opinion from the test is showed in figure 17

![Pie chart showing the preference between list and radial navigation.](image)

**Figure 17:** List VS Radial navigation

The participants got to rate the ease of use after each menu. Looking at those rating of the list menu and the radial menu both of them averaged an easy navigation. The scale of the rating went from 1 (Very easy) to 7 (Very difficult). The radial menu averaged a score of 2.1 making it land in the middle of the easy scale. The list menu averaged a score of 1.3 landing in the very easy spectrum. The ratings for each menu is shown in figure 18
When looking into the statistical analysis of the difference in opinion it is not as certain as the circle diagram above shows. The p-value is calculated to 0.087 which only proves certain trend but not a statistical significance \[\text{(48)}\]. The list got better ratings so it is preferred but there is not a statistical difference between the opinion of a radial and a list.

### 6.4.7 Think Aloud comments from the test

As mentioned in the method section Think Aloud was used in this test which means that besides the data presented above with times and graphs there are comments and insights from the participants gathered during the elapse of the test. The most common comment areas will be presented here along with the ones giving the greatest insights to the method.

- The edge scrolling affected my opinion of the free cursor
- The cursor could be controlled with the left stick and the camera with the right stick.
- Being able to take a shortcut in the fixed radial approach would have improved my opinion of it
- The radial was much more fun to navigate in
- The list feels easier

One of the participant said the following about the cursor test:
Before testing the prototypes I was more negative about it but it felt surprisingly natural - Participant 5.

6.5 Iteration three

Based on the research done of heuristics in games [33][37][39][38] a modification has been made in this thesis focusing only on multiple platform UI problems leaving the game story, game design and game play outside the evaluation. The purpose is to find the situations where the UI gets in the way of the game instead of helping it. Since the PLAY method is an extension of the HEP method it will be the foundation of this smaller multiple platform evaluation. The evaluation is focused on only identifying where a multiple platform issue with the UI exists. This extension will be called Heuristic Evaluation of Multiple Platform Issues (HEMPI). Some points from the Usability and Control section has been extracted. The usability and control contains 9 heuristics: Documentation and tutorial, Status and Score, Game Provides Feedback, Terminology, Burden on Player, Screen Layout, Navigation, Error Prevention and Game Story Immersion. In some of these heuristics there are subcategories connected to the multiple platform issues these heuristics will be included in HEMPI. HEMPI is a suggested modification of PLAY focusing only on identifying multiple platform issues without finding general issues with the game.

6.5.1 Heuristic Evaluation of Multiple Platform Issues (HEMPI)

The points are extracted from the previous method of PLAY [37] with some additions which are marked with italic text to separate it from the previous work.

1. Status and Score
   (a) B1. Game controls are consistent within the game and follow standard conventions for genre and platform.
   (b) B2. Status score Indicators are seamless, obvious, available and do not interfere with game play.
   (c) B3. Controls are intuitive, and mapped in a natural way; they are customizable and default to industry standard settings.
   (d) B4. Consistency shortens the learning curve by following the trends set by the gaming industry to meet users’ expectations. If no industry standard exists, perform usability/playability research to ascertain the best mapping for the majority of intended players.

2. Game Provides Feedback
   (a) C2. Provide appropriate audiovisualvisceral feedback (music, sound effects, controller vibration).

3. Burden on Player
   (a) E1. The game does not put an unnecessary burden on the player by for example an extensive amount of clicks to preform an action.
   (b) E2. Player is given controls that are basic enough to learn quickly, yet expandable for advanced options for advanced players.
4. Screen layout

(a) F1. Screen layout is efficient, integrated, and visually pleasing, *readable on console and accessible on PC*.

(b) F2. The player experiences the user interface as consistent (in controller, color, typographic, dialogue and user interface design).

(c) F3. The players experience the user interface/HUD as a part of the game.

5. Navigation

(a) G1. Navigation is consistent, logical and minimalist.

6.5.2 Test of HEMPI

Since HEMPI is drafted from the points in PLAY [37] it shares its problems. It is not really applicable early in the design process but is more suited to test a finished product. It could be very applicable when trying to port a game. Looking at the original version but with the other platform in mind, this could give a pointer to which interfaces will struggle most when changing the platform.

Using HEMPI early didn’t result in any findings. The points aren’t written in such a way that it is applicable on a prototype and especially not a prototype of a component as in this case.
7 Discussion

In this section the result from the study are interpreted into additions into the Multiple Platform First design guidelines. The results are then discussed and possible improvements or other solutions compared with the chosen ones.

7.1 Result Discussion

Based on the results gathered from the component study several conclusions can be drawn. The test group preferred the list menu over the radial menu. Among the different radials there were no concise result in which one was preferred. Even though it seemed like a landslide for the list menu both the comments and the statistical analysis showed that it was not so. In the comments many said it was more fun and creative with the radial and the statistical analysis could not prove a significant difference between the opinion of the list menu and the radial menu. Comparing the different cursors the fixed cursor was preferred over the free cursor in this test group which could be proven by the statistical analysis. These results are interesting since they suggest that a radial on PC could be a viable option since there is no significant difference in the opinion proven in this study. This study also suggest that a when looking into cursor on console a fixed cursor is preferred by the users and is significantly better than the free cursor. This is within the frame of this study with prototypes. Based on the comments from the users cursor on console seems like a viable navigation method on console and based on the study a fixed cursor is the better choice of cursor.

One other very interesting thing that was discovered in this study was the very strong biases which existed in the gaming community. This was first discovered when investigating cursor on console, there was a bias that cursor on console was bad. This was concluded when reading forums about games and also when talking with the staff at Paradox. This bias will be a very important aspect to investigate further in order for the multiple platform first approach to be successful. It really got the mind spinning and asking why this bias is so strong and where it originates from? Since Console games and PC games have been more separated in the past this could have created standardization differences causing this bias experienced today.

From the user test many comments submerged and the most common ones was brought up in the result section. The most commented thing was the scrolling in the free cursor prototype. There came suggestions that it would have been a better comparison if the free cursor didn’t have the scrolling. But in order to conduct a comparison between the cursor both had to have the same ability, in this case to move over an area larger than one screen size. There were many suggestions to how this could be solved, maybe the free cursor would have worked better in a menu or a screen where no movement outside the screen was necessary was the opinion of
a few of the participants. One very interesting thing is that another participant also brought up that a free cursor in a menu would be a bad idea and that it was in the screen navigation it could have advantages. These conflicting comments show how difficult these problems are, even on such a level where different similar things are compared. Since so many commented on the scrolling and several suggested that the camera could have been moved with the right joystick on the controller while as the left joystick could continue controlling the free cursor. This is an interesting idea which could have potential but it would involve two movements instead of one which could be confusing. The other thing many commented on was that they felt very trapped in the fixed radial navigation where the pointer moved on a line around a circle, selecting objects on that line. On console it is not uncommon to have this sort of navigation but when imported to PC there were strong reactions. This could have something to do with that cursor are so common on PC that the players are used to being able to move across the entire screen. The suggestion from the users to this feeling was that shortcuts should be possible through the circle. This was an idea during the prototyping stage but was not implemented due to the scope of this thesis. This could also be something for future tests and implementations.

This thesis is the first attempt of creating the multiple platform approach. If it could aid in opening peoples mind and creating ideas on which further steps could be taken I am a happy camper. I hope the results from the cursor and menu prototype will make some game making teams consider using these for both platforms. The multiple platform industries is growing since it is becoming more common to play on several platforms. Technologies like the SteamLink and the upcoming Google Stadia makes this behaviour more common. When the conditions changes how the player play the games the industry needs to keep up. I think research like the one sampled in this thesis could aid in creating better games in the future just by being aware of things such as biases and which components work when. The study itself gave some really interesting insights and data to be used when developing and investigating games in the future. Some things could have been done better which would have made the study even better. The test group was also very PC heavy, 9 out of 10 had PC as the primary gaming platform which could have affected the results.

Since the web development business are experiencing a similar problem with the desktop versus mobile it could it smart to keep a watchful eye on their progress with the problem since their solution will probably provide some interesting insights for the game industry as well.

7.2 Multiple Platform First

Each part of this study has contributed with knowledge to this new approach, Multiple Platform First. Some content has turned into methods, other into libraries and some into plain insights.

7.2.1 Insights in Multiple Platform First

A few insights has come to light in this study. These insights are thing important to research further in order to develop the Multiple Platform First guidelines.
Biases

Biases turned out to be important to this study. There exists a bias between Console and PC gamers in how they expect their games to look. This became most obvious in the cursor prototype, both when reading forums and when talking to the staff at Paradox. Cursor on console is perceived as a cheap, not very good solution. This showed up in the test since people were surprised they did not hate it as much as they thought they would. Developing these guidelines further would be interesting to investigate further since an understanding of why there is such a strong bias could be a major part of solving the problems this causes.

General vs Deeper level studies

When attempting to create general guidelines this thesis fell short since it was a limited time and only one person doing the investigations. This does however not mean that such conclusions are impossible to find. In a larger study it could be possible and would be of immense use in the development of multiple platform games. This thesis investigated guidelines on a component level instead. This turned out to be more plausible considering the time frame of the thesis.

7.2.2 Methods in Multiple Platform First

Some results from the study has turned into methods that can be used when encountering Multiple Platform related issues.

HEMPI

HEMPI is today a draft of a method which could be used to detect multiple platform issues in finished games or maybe a beta at least. Moving on with it would be interesting to modify it to be useful in a prototype stage of a game as well. There might also be other methods that the heuristic approach used in this thesis.

Component Library

When doing the deeper level investigation on components it turned out to be a very useful addition to the guidelines to create a library of the components investigated that could be extended with more components in the future. This library could work as an aid when designing user interfaces to know which components could work and how they are perceived by users.
8 Conclusion

The purpose of this master thesis was to investigate a new approach as an alternative to the existing ways of designing games. The following problem statement was made: *How could the Multiple Platform First look?*. In order to investigate this the following research questions were stated:

- Can traditional evaluation and inspection methods be modified to identify and evaluate multiple platform issues?
- How affected is the playability of games when using traditionally console components on PC interfaces?
- How affected is the playability of games when using traditionally PC components on console interfaces?
- How could interfaces be designed to mitigate the difference between console and PC based on the result from this study?

The first research question was answered by previous research. Several different scientists have investigated heuristics and how it could be applicable for games. Nielsen’s heuristics have been the base of this research which has formulated points focusing more on how games work which is different from other software development. These methods are themselves rather new and still have not solved all issues. When adapting the latest version of this research to address only the multiple platform the issues from the previous version follows. This issue is the same as with Nielsen’s heuristics. It is hard to apply early in a design process on unfinished things, they need to be very Hi-fi or implemented in order for heuristics to come up with interesting findings. When testing HEMPI on the prototypes in this study it was quickly realized that they were to low-level to get any results. HEMPI turned out to be a better method for example when a game is to be ported. Going through the points of a PC game but imagining a console controller would certainly bring most of the problems into the light. This however needs to be tested in order to say for sure but it shows great potential.

Moving on with the other questions the first problem encountered was the difficulty of finding general conclusions. When starting the litterateur and field study the intention was to find these general solutions and suggestions. It was quickly discovered that this would not be possible within the scope of this thesis why a deeper level approach was chosen instead. This does however not mean that general conclusions are impossible to find but it will require more time and a very extensive study in order to draw such conclusions. The study conducted went down to a deeper level and investigated different components and how the could be used in multiple platform interfaces. In this thesis only two components were investigated but if a library were created with more components a multiple platforms interfaces design process could take off from the component studies and it could
aid in the selection of components. When deciding between two approaches involving the study’s components, the results can either help steer the decision, or outright settle it.

In summary all research questions was answered even if some of them were answered by asking more or new questions.
9 Future Work

There are so much that could be done within this field. The study conducted within this thesis was limited both in time, scope and participants. Looking into the general guidelines which this thesis failed to achieve could be very interesting for game development in the future. This would probably need much time and resources since investigating existing games thoroughly and conducting a large user study would be viable ways of getting the data for this type of conclusions. Extending the component library beyond the two components explored in this thesis could create a very useful tool when designing multiple platform games and maybe create a bigger variation in PC and Console games as well. If a radial is accepted by the users, why not use it when it fits our game? Taking the heuristic evaluation modification HEMPI further could turn into a helpful tool, maybe even in a ideation or prototype phase. A further investigation of how biases affect a gamers appeal to a certain component or UI could be very interesting for making these guidelines even more useful.
References


[34] Melissa A Federoff. HEURISTICS AND USABILITY GUIDELINES FOR THE CREATION AND EVALUATION OF FUN IN VIDEO GAMES. Technical report, Department of Telecommunications of Indiana University, Indiana, USA, 2002.


A Interview Questions

Below are the questions asked to the participants after completing different stages of the user tests. The rating questions was answered using Google forms while as the other questions were answered orally.

A.1 After each cursor prototype

- Please rate how the following worked for you: (1 Very Easy - 7 Very Difficult)
  - Selecting a house
  - Selecting the capsule
  - Moving around in the world
- How did you experience the navigation? (1 Pleasant - 7 Frustrating)

A.2 After both cursor prototypes

- Which cursor did you prefer?
- Why did you prefer that cursor?
- How would you rate the overall cursor on console experience (1 Pleasant - 7 Frustrating)

A.3 After all radial menu prototypes

- Please rate the ease of navigation from the different radial menus: (1 Very Easy - 7 Very Difficult)
  - Free pointer
  - Limited pointer
  - Fixed pointer
- Which radial navigation was best?
- Which radial navigation was worst?
- Why was the "best" radial best?
- Why was the "worst" radial worst?
A.4  After each menu prototype

- Please rate how the following worked for you: (1 Very Easy - 7 Very Difficult)
  - Opening a menu
  - Selecting an object in the menu

- How did you experience the navigation within the menu? (1 Pleasant - 7 Frustrating)

A.5  After both menu prototypes

- Which menu type did you prefer?
- Why did you prefer that menu?
- Could radial menu be a viable menu type on PC?