KYMENLAAKSON AMMATTIKORKEAKOULU

University of Applied Sciences

Degree Programme in Information Technology / Game Development

Obi Ikemefuna Marizu

MOBILE GAME USABILITY AND HEURISTICS

Bachelor's Thesis 2015

ABSTRACT

MARIZU, OBI IKEMEFUNA   Mobile Game Usability and Heuristics

Bachelor's Thesis   33 pages + 5 pages of appendices

Supervisor   Paula Posio, Principal Lecturer

Commissioned by   Romar International

March 2015

Keywords   Usability, Mobile Gaming, Heuristic Evaluation

A new area of research is the playing of games on mobile devices in a mobile context. It is necessary to create a new set of heuristics specifically targeted to increase the usability of mobile game designs. Several researchers have developed playability heuristics to cover the mobile context aspects, which are normally used along with an evaluation method specifically designed to identify the probable playability difficulties in the user interface. However, this has to be done early on in the game design process. Korhonen and Koivisto (2006) created a few heuristics that cover general mobility, game play and usability issues of the game. Their method aimed primarily at *pre-production and production phases* of a game project, however with development of the heuristics they could also be applied in post-production phase.

This thesis is an attempt to understand the game engines and platforms and the usability of game especially Unity 2d and 3d engines and platforms and the principles of game design currently being used to create high quality games.

Thesis includes a usability case study on playing games developed on Unity engine on different mobile platforms; Andriod OS, Apple IOS and Google chrome browser for smartphones.

The results analyzes how different users play mobile games, develop habits and how less instruction each individual users needed in playing mobile games.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Aims

The aim of the thesis is to study usability of games implemented with different engines and playable on a single or multiple platforms, and how that affects the general usability and playability of the game. The aim of the usability test carried out is to determine usabilty of games playable on different platforms and implemented on a single game engine, in this case Unity engine. In addition, an aim is to understand how users interact with all of these platforms both under supervision and unsupervised.

1.2 Contents

The first part is an introduction to the topic. This is followed by a detailed description of game engines and platforms, namely the Unity 2d & 3d game engine and HTML5. The third chapter defines various models involved in assessing the usability of a system. The fourth chapter is concerned with the principles of game design, creation, balancing and troubleshooting. The last set of chapters focuses on the usability case studies in different platforms, usability test results and discussions. The last chapter focuses on the conclusion.

1.3 Company

Romar publishings is a subsidiary of Romar International, established in 1995 solely for the purpose of publishing educational materials. The company in late 2010 started to develop and build mobile applications for private clients.

1.4 Mobile Game Market

Newzoo game market research specialists are certain that the mobile games can generate higher profits than console games in 2015. The company projected in their quarterly global game market update that revenues from tablet and smart phone titles will grow to $30.3bn in 2015 from approximately $25bn in 2014, a 42% increase from 2013 (Fig 1) . They predicted expansion in the emerging south-east Asian markets

and explosive 86% growth in China. They also foresee steady growth in the mature European and US markets. They also predict that in Japan the decline of the previously prevalent feature phone games will be offset by the increased revenues of iOS and Android (Stuart 2014.)
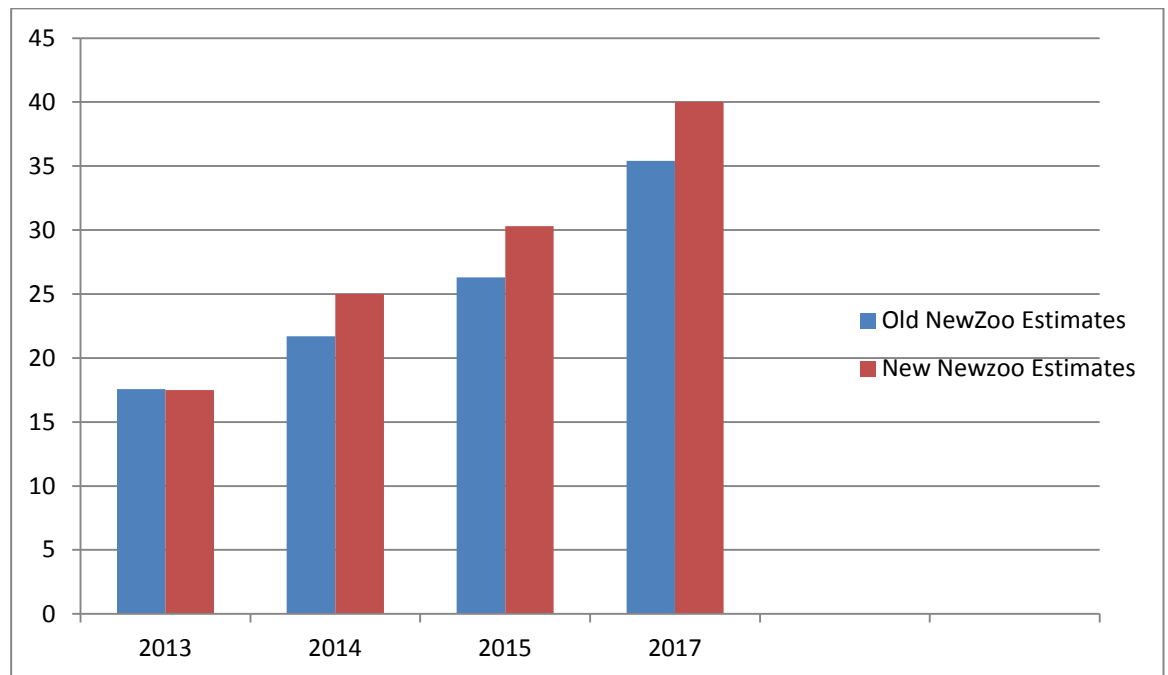

Fig 1. The mobile game market growth and expectations according to Newzoo

There are potentially 20-30 times more customers playing mobile games than on any more traditional gaming platforms, meaning that mobile games have the potential to increase their lead over consoles in terms of revenue. There has been a dramatic change in the present day mobile games market compared to merely a few years earlier (Fig 2). Presently, over one third of the monthly spending of digital gamers in the United States alone goes to mobile games. (Kelly 2014.)

However, after a period of initial growth, the mobile games industry is reaching a mature stage. Thousands of established game developers are competing with each other, making it difficult to stand out from the pack. An increasing cost of user acquisition makes it harder to reach out to and attract new customers. At the same time, higher customer demands and expectations are increasing development costs. Small scale de-

velopers face a large challenge as they try to build and protect their own intellectual property against the larger brands in the market (SuperData Research, 2014).

Worldwide mobile game revenue
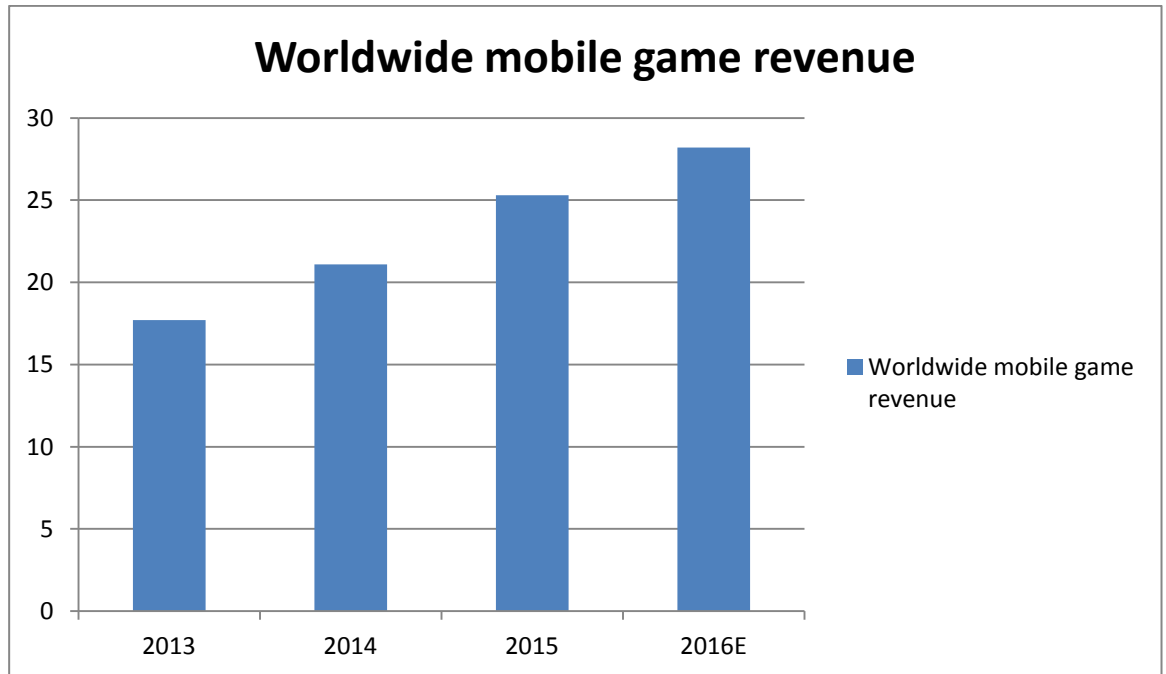
Fig 2 The worldwide mobile game revenues from 2013, according NewZoo

2 PLATFORMS AND GAME ENGINES

2.1 Platforms

A platform is a set of technologies on which other technologies and applications are built on. (Techopedia.com, 2014) A computing platform is the software and hardware that identifies a computer system. It is on the whole acknowledged as a synonym for all the present and available machines that are qualified to be considered as a Computer. Examples of common platforms for games are Playstation, Xbox, Sony PSP, Wii, Facebook, iOS, and Android. Many platforms are designed for mobile games.

A mobile game is a video game that can be played on any hand-held gadget like a cell phone, PDA, smart phone, tablet, portable media player or even a calculator. This also includes video games played on handheld devices, such as PlayStation Vita or Nintendo 3DS. Mobile games have created a revolutionary transformation to the relationship between humans and machines. This has consequences that go beyond the

gaming experience to affect the design of interaction structures and interfaces. (Consumerreports.org 2012).

According to Takahashi (2014), Mobile is a fast growing platform with the previous year growth surpassing 35 percent. The competition is increasing on a worldwide basis. However, creating mobile games is not without its own challenges. According to Fiksu *mobile gamers are fickle*; a game is opened hardly once by 19 percent of new players whereas after the first 24 hours 66 percent of players do not play a game ever again. (Takahashi 2014)

2.2 Game Engines

A game engine is a software developed and designed for the creation of video games. The most significant aspects that game engines offer are interoperability between existing, diverse gaming systems, and a slew of components provided to developers to speed up and simplify game creation. Video games are developed by using these game engines for mobile devices, video game consoles, and computers. The essential core functionality of a game engine consist of a rendering engine for two-dimensional or three-dimensional graphics, sound, coding, collision detection and collision response (physics engine), networking, animation, artificial intelligence, memory management, streaming, threading, localization support and a scene graph. Reusing and adapting the same game engine to develop an array of games can often economize the process of game development, and make it simple to "port" games to a number of different platforms. (Ward 2008.)

Game engines provide a great number of sophisticated components and important features for making games. Their components are optimized specifically for making games, to the detriment of various different types of applications; unlike general development frameworks, like the .NET used for building Windows applications or Cocoa Touch used for building iOS applications. Game engines have graphics engines incorporated that are designed to be as fast-paced as possible in order to compensate for the lack of simple tools for creating menu bars and widgets and they include sound engines which place sounds in 3D space in place of system sounds and default popup windows. (Michaelenger 2013.)

Game engines provide reusable components that are manipulated by the developers to bring a game to playable level. Loading, displaying, and animating models, physics, graphical user interfaces, collision detection, and even parts of the artificial intelligence in a game are all components provided by the game engine. The components belonging to the actual game, on the other hand, are the content of the game, textures and specific models, the way objects interact with nature and the meaning of object collisions and input. (Ward, 2008)

One of the greatest advantages of game engines is the provision of tools to developers for building games so that they do not really have to reinvent the wheel. Game engines essentially perform the thankless and menial work of game developing; from taking care of the graphical optimizations necessary to obtain a nice frame per second rate to trading common asset formats, so the story, atmosphere, and other factors essential for creating a great game becomes the focal point of the developers.

One of the greatest disadvantages of game engines is their tendency to homogenize the games that are created on them. A game engine made for first-person shooters cannot be the top choice to utilize for a racing RPG, but the engine that gets chosen can end up weighing down a game's creative expression. For example, due to technical limitations on Unreal Engine 3, all games made with it have texture pop, similar sound effects, and identical visuals, despite their different styles.

Despite this disadvantage, game engines offer an unquestionable necessity for the contemporary world; platform interoperability between desktops, gaming consoles, and mobile devices. Depending on the number of platforms the game engine supports, the central idea is that the developer builds the game on an engine and can then export the game to an array of different platforms. The advantage provided to developers by this option is difficult to understate, as the developer only needs to build a game once and then make it accessible on a wide number of platforms by merely pressing a button. However, developers will need to consider whether their game requires additional adjustments when changing to touch, controller, or keyboard.

Chapple (Chapple 2014) identified 16 major game engines for the year 2014. Pricing models range from free to subscription and even to the top pricing levels and the engines are interoperable across multiple platforms. These are:

1. Unreal Engine 4
2. Unity
3. CryEngine
4. GameMaker
5. Havok Vision Engine
6. Project Anarchy
7. ShiVa
8. BigWorld
9. Lead Werks
10. App Game Kit
11. FPS Creator Reloaded
12. Reach 3dX
13. Hero Engine
14. Marmalade
15. Turbulenz
16. Game Salad

2.3 Unity 2D/3D

Unity is one of the most powerful game development tools to appear in recent years. It is a quality game engine that developers use to create video games for a number of different platforms. Unity is not only a professional development tool applied by thousands of seasoned game developers on a daily basis; it is the most easily accessible modern tool for rookie game developers as well. Until recently, a newcomer to 3D game development would have faced a number of difficult barriers at the beginning of their careers, but Unity has now made it simple for them to start working right away. (Hocking, 2014)

A matchless toolset, effective workflows and swift iteration attributes of Unity can be advantageous to any developer creating 2D or 3D games, visualizations or simulations. One can create anything, be it a 3D console title or a 2D mobile app, with the help of this versatile game engine. Unity offers physics simulation, screen space ambient occlusion (SSAO), normal maps, dynamic shadows, and many other features, yet similar features are offered by several other game engines as well. Nevertheless,

two main advantages provide Unity a competitive edge over other identical cutting-edge game development tools (Hocking, 2014); these are an extremely greater level of cross-platform support and a highly productive visual workflow.

There are several other advantages of Unity. The biggest is easy importing. Unity is able to import animations and models from any 3D application that exists today. The importing is as simple as dragging and dropping them like a file into the concerned folder. The creation and population of 2D and 3D scenes (game levels in Unity) is extremely effortless. Unity is powerfully optimized and gets the excellent visuals and performance from the game. In order to ensure the best results, the developer can use the dedicated Profiler window to maneuver memory usage, CPU & GPU performance, draw calls and much more. Unity works well with both 2d and 3d. In 2D games, sprites are automatically atlassed to reduce the download size. Another large advantage is that two industry-standard physics engines; NVIDIA® PhysX® and Box2D; are both incorporated into one workflow. Developers can apply the same system of joints, rigid bodies and colliders whether they are developing 2D or 3D game content.

Unity has a sophisticated, lifelike animation system, which is fully integrated with the engine. A wide variety of realistic motion can be created from just a few clips thanks to a hierarchical state machine that blends trees and retargets automated. Shapes can be blended for facial animations. Unity can create automated animations from multiple sprites. With the animation curve views and window's dopesheet the developers exercise complete control over 2D animations. It is extremely effortless to blend the 2d and 3d scenes in Unity rendered using flat planes. It is the highest quality engine available with bringing baked lighting and real time shadows to the 3D games. Furthermore, a lot of extremely exceptional shader packages are presently available from the Unity Asset Store. (Hocking, 2014)

However, Unity does have a few disadvantages. Game development in Unity can quickly become extremly complex. In sophisticated scenes (game levels), the developer can easily lose track of the objects which have particular components attached to them. Another great disadvantage is that Unity does not support linking with external code libraries. A manual copying of the several libraries available in every project becomes tediously must. (Hocking, 2014)

2.4 HTML5

Compared to many other technologies, writing code in HTML5 is easy to learn. HTML5 is guaranteed to do a lot of things and one of those is to *make the world a far more interesting place* (Rettig, Karlins and Wilson, 2012). HTML5 is a modernized version of the language HTML, having new attributes, elements, and behaviors, plus a bigger set of technologies that permit further development of diverse and powerful Web sites and applications. HTML5 is normally used as an umbrella term for a lot of diverse standards. A few of those standards were already a part of previous versions of HTML, but have been reorganized with HTML5. Other standards are new additions to HTML. HTML5 has richer media elements with more descriptive semantics and permits direct embedding of mathematical formulas. The canvas element now supports the HTML5 text API.

Mobility is one of the greatest advantages of HTML5. With HTML5 the developers can deploy applications as local web applications on mobile devices like smart phones and tablets where applications can also be viewed in browsers. Moreover, the applications do not have any limitation by frames of the windows, and can run in full screen mode on compatable browser. The users have complete control of their screen space and the devices.

HTML5 allows developers a centralized way of deploying their applications on all platforms with browser accesibility. Mobile applications can use the same distribution and monetization channels as normal applications. Migration of development tools to mobile devices becomes simpler as HTML, JavaScript and CSS are the backbone of the web applications. (TECHNOSOFT, 2014)

However there are a few disadvantages to using HTML5. The efficiency of the code is dependent upon the translation engine as developers are normally working in languages that are not inherently incorporated with the platforms. As a result, it may happen that the code is redundant or coding techniques become inefficient. At times, mobile applications may take a lot of time to get ready as developers have to separately write codes for each single platform. The disparity in execution times of platforms can also create problems in the mobile application development platform. The HTML5 standard was developed in a rather short time span hence inconsistencies were created

while implementing the tags, attributes, and JavaScript APIs. Although tools are available to sort out such inconsistencies, at times these features behave entirely differently on different platforms.

However, HTML5 continues to be "the in thing" in the world of mobile game development as the advantages are in greater number than the disadvantages overall.

3 USABILITY

In 1990's usability was recognized to be a growing and important trend in software creation. In 1997 Lin, Choong and Salvendy stated that usability was becoming an increasingly significant software standard. The usability of software is the measurement of how easily it can be used to perform a given task. It can have a profound impact on millions of everyday computer users who have never done any formal study of computer science, and even those who have.

3.1 Definitions

Shackel (1991,19) defined usability as `. . . *the capability in human functional terms to be used easily and effectively by the specified range of users, given specified training and user support, to fulfill the specified range of tasks, within the specified range of environmental scenarios'*. A more utilitarian definition can be found in ISO (1993): `. . . *the quality of use: the effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments'*. Put another way by Preece (1993, 23),  `*the goals of HCI [Human Computer Interaction] are to develop and improve systems that include computers so that users can carry out their tasks: safely, effectively, efficiently and enjoyably. These aspects are collectively known as usability'*.

Measures of effectiveness relate the goals or sub-goals of using the system to the accuracy and completeness with which these goals can be achieved (Bevan and Macleod 1994). Effectiveness is highly related to the completeness of the functionality of the software and the effectiveness of each subpart of the functionality. There are several approaches which can be used to evaluate interface usability. The most commonly used approach is usability testing. Usability testing is based on the principle of data

analysis and capturing of how users interact with a system or a system prototype. Many user tests are carried out in specially equipped usability laboratories that are set up to capture a user's verbal and non-verbal reactions to the system.

3.1.1 Shackles Usability Model

Shackel (1991, 25) is the foremost creator of this operational method. He described usability as the "*artifact's capability, in human functional terms, to be used easily, effectively and satisfactorily by specific users, performing specific tasks, in specific environments.*" The crux of the working description is that it unambiguously sets usability close to that of the interface amongst operators and the object. This surpasses the characteristic aspects-based descriptions popular in the field. Moreover, in establishing benchmarks for evaluating usability, this method supports the assessment of every tool and the consequent clarification of the test outcomes. Usability consequently refers not to a set of interface characteristics, but to a situation-dependent yardstick of human-computer interface.

The study refers to the advantages of tight usability benchmarks; for instance, setting precise, solid times for job accomplishment and limits on how many mistakes can be tolerated. Shackel presents the case that these standards are crucial in informing the design method as well as assessment. He stressed the need for a consistent method to assess the likeability of features. Such a method, Shackel argued, would be more effective than other larger, more expensive systems. However, the nature of the usability essentials recommended by this study requires experimental validation.

3.1.2 Nielsen Principles

Usability inspection (Nielsen 1994) has seen growing use since about 1990 as a technique to assess user interfaces. Usability check is the univerisal name for a set of systems that are established on having assessors inspect the interface. Heuristic assessment is the best informal process and includes having usability experts evaluate whether every interchange component follows well-known usability values the "heuristics" (Nielsen 1994)

Nielsen developed a number of heuristics that should be considered when designing a user-friendly system. They are:

- Perceptibility of system condition

- Understanding bridge between system and nature

- User independence and control

- Consistency and adherence to standards and conventions

- Elimate error tendencies

- Understanding as opposed remembering

- Flexiblity in usage

- Easy recognition, diagonistic and recovery of errors

- Comprehensible and accesible documentations for assistance

3.1.3 Gerhardt Heuristic Evalution

A Heuristic assessment is a procedure where somebody skilled in usability values evaluates an application, website, or software. He or she rates the solution in regards to a set of rules or values, known as heuristics. Following these heuristics will lead to the creation of more usable software. Gerhardt takes an all-inclusive method to assessment. Gerhardt Powals' values (Gerhardt-Powals, 1996) are itemized below.

- Automate non-essential tasks:

  - When non-essential tasks are automated, mental resources are available for essential tasks.

  - Eradicate tedious calculations, estimations, comparisons, and unnecessary thinking.

- Decrease vagueness:

  - Display data in a way that is understandable and apt.

- Merge data:

  - Reduce mental load by adding data classified to be of lower class into a higher-class.

- When presenting new information, its meaning should be easy to understand:

  - Utilize a recognisable framework, making it easier to perceive.

- Use everyday language and figure of speech.

- Use names that are related to the context :

  - Context-dependent.

  - Endeavor to better recollection and recognition.

  - Index data in appropriate way to decrease search time.

- Reduce data-driven tasks:

  - Reduce the time spent processing raw data.

  - Make appropriate usage of color and graphical reprensentations.

- Display only essential information at a given point in time.

- Provide more than one summary of data when appropriate.

- Practice reasonable repetitions.

3.2 Defining Mobile Game Usability Heuristics

Usability methods are a valuable tool for software creators, but few currently existing techniques are especially applicable to computer game design. Heuristic assessment is an inexpensive usability technique that has been employed with a great deal of success in the designof non-game software, and its benefits, such as elasticity and low operating costs, would make it a good addition to game design procedure (Korhonen and Koivisto, 2006).

Games vary from service software in several key features. In games, the goal of the software is to be entertaining and provide pleasure in playing the game. Learning to play the game, unraveling complications, and seeing new things are all aspects of those traits. Furthermore, in a game, the participants are not aware beforehand of what to anticipate. Game creators have provided the game content and well-defined objectives that are essential for the players to accomplish. Thought provoking complications as the player works toward these objective are part of the game's experience. Consequently, employing general usability heuristics in game assessments is not enough and trying to apply them directly to game design would leave several vital fea-

tures of the game unrefined (Federoff, 2002) Efficacy, capability and fulfillment are significant to each game. One cannot say the same for several non-usability features of games for example story, game behavior and personality growth.

The motivating energy behind usability techniques is to boost artifact standard (Lindholm, Keinonen and Kiljander, 2003). The benefits of these enchanced features are not only to create a superior game, but correspondingly improve the probability of the game's success. While employing usability techniques in any design procedure is an added expense, these techniques can assist in reducing costs later in the game's development. Initial usability assessment can aid in recognizing glitches earlier in development, when they are more inexpensive and easy to repair (Bias and Mayhew, 2005). The great thing concerning usability techniques is they can be adapted in a variety of ways.

The benefits of this flexibility are twofold. Primarily, a basic design procedure can be created that can be made to fit for any game's budget and development timeline with only minor adjustments. Furthermore, when a new category of games emerges, creators will have a template for the tools that can be used to appraise it. Possibly the paramount benefit of heuristic scrutiny for game appraisal is that it does not include explicit user testing. This lack of reliance on users has several advantages for game design. For one thing, it makes it easier to control introduction of the game to the general public. This secrecy is significant for marketing reasons, which frequently regulate the success, or lack thereof, of a game. Not relying upon user testing also helps to lower expenses, as there is no need for user testing laboratories, the hiring of multiple participants, or the scrutiny of a large amount of statistics.

4 FUNDAMENTALS OF GAME DESIGN

An elegantly designed game is simpler to create, more enjoyable to play, and more likely to succeed in the market. Several philosophies are important to keep in mind while designing a game, but the first and foremost concern should be keeping the player engaged. One way to do this is to provide well-defined plot points and goals to the player throughout the game. Another way to keep players engaged is to make use of their expectations given by the story or UI. A player's anticipation for game elements like prizes or payouts can keep them motivated, and occasionally going against

their expectations can be a pleasant surprise. It is also important to make sure that the player never feels lost. Elements such as visual cues to mark where to attack an enemy or an area they should explore can help the user feel in control and remain motivated. (Adams and Rollings 2010.)

4.1 Game Innovation

Peter Drucker (Drucker 2006, 78) wrote, *"Effective innovations start small. They are not grandiose."* He is correct. An insignificant idea followed meticulously is significant. Apple made an impact with the Macintosh in large part by taking advantage of inventions that Xerox ignored. Desire and persistence are vital to prodigious game inventions. Designers must revolutionize as they aspire to earn revenues and create the best game.

4.2 Game Creation

Game creation is the procedure of creating a game, from the first idea to the culmination of the Alpha, Beta, and Final stages. Game designers are responsible for the vision and the overall feel of the game. Their focus is primarily on levels, gameplay, and interface design. This includes the mechanics of the game and the story behind the game. Game developers need to translate these elements into a programming language.

The Unity 3D engine allows the developer to employ three different programming languages: a version of Javascript, Boo, and C#. The best thing about this is that regardless of language (C# or Javascript), the codes are compiled by theUnity engine with a press of a button any time the developer runs the program.

Another important game element is art. If the game is a 3D game, the developer needs animation and modeling software. If it is a 2D game, a vector-based software solution, for instance Photoshop, Flash, or Illustrator or even 2D Paint are adequate.

4.3 Game Balancing

In game design, balance is the idea and the practice of modifying game rules, typically with the objective of preventing any of its module structures from being unproductive or unwanted compared to their equivalents. An unbalanced structure signifies squandered development reserve at the minimum, and most seriously it can compromise, the game's complete set of rules by making significant parts or tasks extremely difficult to execute. (Newheiser 2009.)

The most important goal of balancing is to keep the game difficult yet possible to master at the same time. It should be complex, but still comprehendable. The objectives of balancing can change radically as players compete with the game's setting and/or non-player characters. Such player versus environment games are typically balanced to walk the fine line of frequently challenging players' capabilities without ever creating undefeatable or one-sided complications. This changes balancing into the administration of dramatic arrangement usually referred to by game creators as "pacing". (AI intelligence 2014).

Balancing does not always mean making a game fair. Replication games can be balanced deceitfully so as to be true to life. A war game may possibly put the player into the character of a general who was conquered by an overpowering force, and it is usual for the capabilities of sides in a sports game to reflect those of the actual squads they represent, regardless of the implications for players who select them.

There are numerous methods for balancing games, and impacting how exciting they are to play. Video games frequently permit players to impact their balance by providing a selection of "difficulty levels". Other balancing methods make use of symmetry, statistical analysis, randomization, feedback loops, or a game master.

4.4 Game Troubleshooting

Developers face additional difficulties when designing mobile games. For example, mobile devices frequently have to deal with a slow network response time. Compared to devices like desktop computers and laptops, interface design for mobile devices is more challenging due to the hardware sacrifices made by hand-held gadgets for the sake of portability. Frequent focuses of troubleshooting are the affordance cues, the buster principle, the dominant strategy and Fitt's law. These work on several diverse values like the social gratification of the players.

5 USABILITY TEST PLAN FOR CASE STUDIES

5.1 Introduction

The report documents the usability and playability test for Robot run 1.0 which was developed for Romar Publishings a subsidiary of Romar International. The game play is centered around a robot collecting rings to get points, jumping and climbing barriers to avoid obstacles. Game uses only touch sensors.

The parts which were evaluated in the usability test, were the logical flow of the game on all selected platforms, the accessibility and location of the buttons, general appearance look and feel of the game, and finally users interaction to playing the game .

The game is intended for smartphone users on different platforms. The target group are young adults, male and females between the ages of 15-25 years, who uses their devices for gaming purposes. The general user demographs tested was 12-45 years.

The environment where the application can be used is not specific. There is no restriction set on the application so therefore the application can be used as long as it is installed on the device and there is enough power to run the device.

Application runs both in online and offline mode, and data received during the use of the application is not sent to any server and cannot be shared online. Data collected are stored on the device memory.

5.2 Test Objectives

The objectives of the test are; to monitor user group response to the logic, flow and playability of the game. Also, Habits users have developed over the short period of the test phase.

The users interacted with all the features of the application, launching from Android, IOS dashboard and launching on browsers, starting new game, playing game using fingers, as the game uses capacitive technology, navigating back to main menu after end of game session and viewing top scores and navigating other menu items like how to play. Users were also asked to exit the whole application.

5.3 Development Methodology

5.3.1 Participants

For the test to be carried out successfully, test participants are required. Highlights of participants characteristics are highlighted below

- The number of participants tested was 10 for the general testing on selected platforms and then 5 for the playability testing on selected platforms.
- For the playability testing the target was the young adults and teenagers, participants were between the ages of 15-25. For the general testing the target group was any smartphone device users between the ages of 12-45 years.
- Selected users were familiar with mobile computing platforms, and the use of smartphones.
- The participants were approached in no particular order, only essential characteristics was the knowledge of using a smart phone.

5.3.2 Context of the Product Use in the Test

It is important to have a guidline of how the test will be carried out, what the test process will include and what expectations are before the commencement of the test.

- Any known differences between the evaluated context and the expected context of use, evaluated contest being the result of the usability test and expected context being the researched theoritical aspect of the thesis.
- The first part of the testing covered all the tasks in the application, which are: Starting a new game, game play , navigating back to main menu after game end, navigating the highscore tab, navigation the how to play tab, navigating the about app tab and exiting the application.
- The scenarios followed were user led, with the test conductor available to answer questions or assist when needed.
- The tasks were selected on the basis that all the operational functions of the game had to be checked.
- For the first testing phase participants were given a simple description of the application, and then asked what they would do when presented with each view, when an action was taken the conductor would then present the appropriate new view to the test participants.
- During the 2nd phase of the testing, participants were given the phone and allowed to navigate through themselves.
- The test criteria for each task were relatively understandable by the participant.

5.3.3 Test Facility and User Devices

All of the application testing was carried out at Dunstable public library, while playability testing was carried out at different locations, all in the United Kingdom.

Used display devices were:

- Samsung galaxy S5 running Android OS 5.0 Lollipop
- HTC running on Andriod 4.4 Kitkat
- Iphone 5s running IOS 8.1
- Google chrome browser on Samsung galaxy S5

5.3.4 Test Administrative Tools

For effective data collection, administrative tools play a major role in how data is collected and how collected data are stored for analysis.

- A simple questionnaire was used to find out how the particpants reacts to the game application and what kind of changes the conductor expects.
- Data was collected by observations, which were marked down with pen and paper.

5.3.5 Experimental Design

For the logical flow of the game usability testing, the control variables were that the same starting point was given to each test participant, after which they decided the path taken through the app.

For the icon testing, the control variables were the icons presented, which were the same, the independent variables were that one group of test participants were told what the icons were meant to represent, while the other group were not told, and were in fact asked to offer what they thought the icons could be used to represent.

For the playability testing, the control variable was making the conductor use their fingers to control the game object on different testing platforms. The independent variable was letting the test participants move their fingers on the object as it pleases them.

Written notes were taken for all the testing that was carried out and different individuals were quoted in the test results

6 USABILITY TEST RESULTS AND DISCUSSIONS

The findings are categorised using this symbols:

✚ Positive approach

💡 Suggestion made; Good idea

➖ Negatives/ problems that made participants hesitant

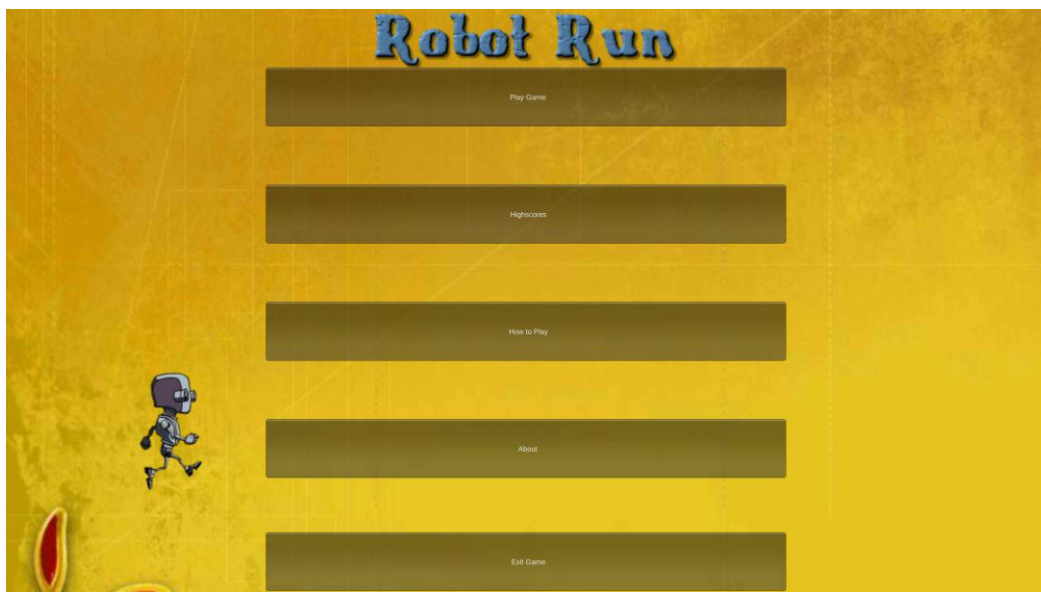6.1 Result of general Usability testing of Robotrun

Fig 4 Home screen of the game

This test phase consist of the "general look and feel" of the game which includes, testing of texts, Colors, icons and general navigation style.

✚ Most of the participants found the background theme appealing

✚ All participants where happy about the menu navigation style of the application

💡 Half of the participants suggested that the font size of the menu text should be incresed.

✚ Participants found relatively easy to start a new game

▬ Two of the participants stated the application should also be available in portrait format.

Fig 3 Highcore page view of the game

This phase of testing involves the users navigating the highscore of the game and giving an opinion to what they believe should improve or removed.

▬ 7 participants believed the highscores section should not only have one highscore but numerous highscores in ascending or descending order.
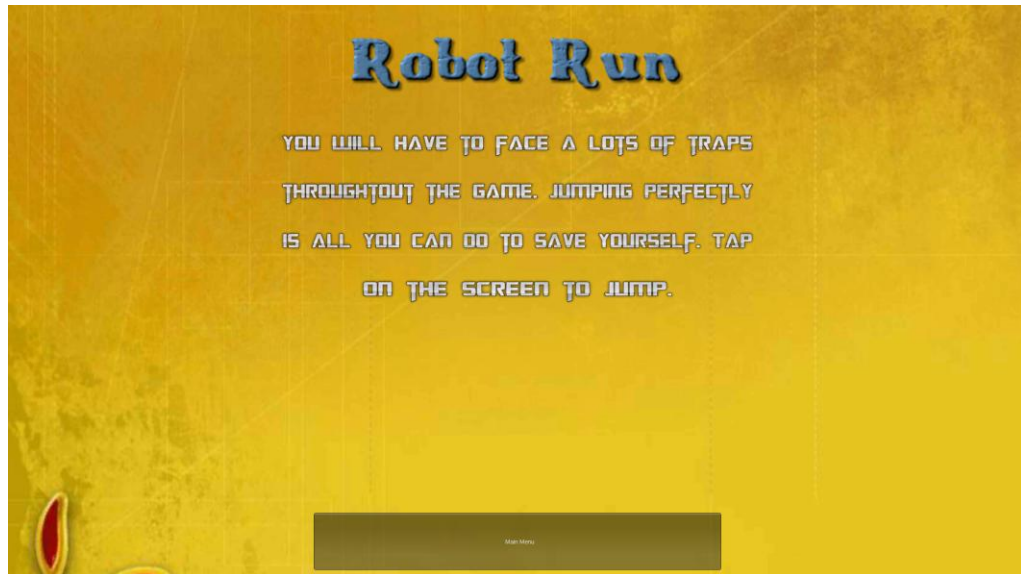
Fig 5 The game manual page

Testing the game manual is an important test aspect as it the guidelines for users on how to play the game, Fig 5 gives an insight to how users should go about playing the game. Readablity of the text and how descriptive the text was, was tested.

▬ Most of the participants also found it difficult to play the game base on the instructions given on how to play the game. They have to count on their experience of mobile gaming.

🔦 "How to Play" is not descriptive enough and more details on how to play the game should be added.
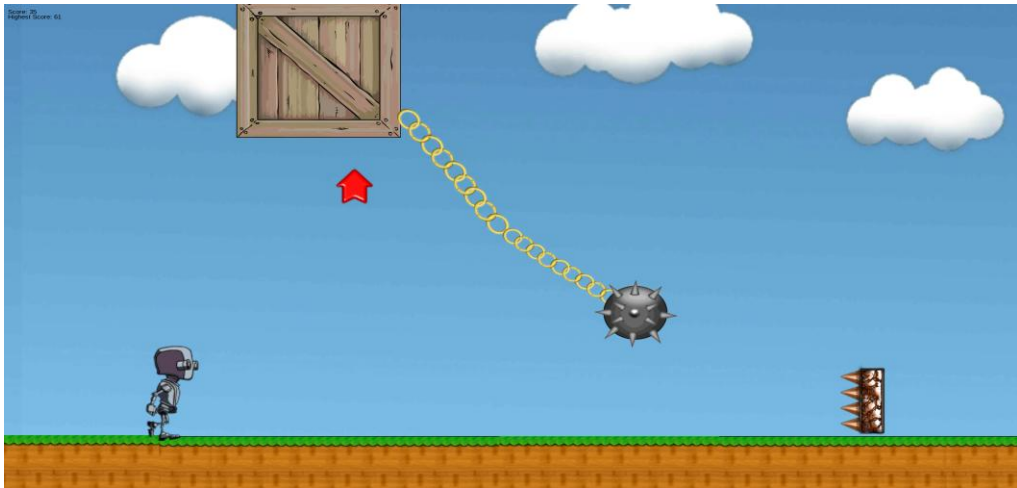
Result of Game play experience

Fig 6 Inplay session of the game

An inplay view of the game (Fig 6) was a test where users were asked to play the game by themselves using information they read from the game manual. The outcome shown below are the results of the game play experence and interactivity testing on different test platforms.

▬ On the web browsers, users find it difficult to get the game object to jump the barriers in time. The response time was slow.

💡 Game was relatively easy to play as no external forces like sensors were needed to play the game.

▬ 3 of the test participants on IOS complained of a second lag.

▬ Participants could not pause game during game play

▬ Participant could not return to game's main menu during game play

6.2   Results Discussion

From the results, the users strongly suggest that the user interface of the game, most importantly from the main menu should be modfied. The colour schemes and the text fonts and sizes also needs to be further developed. Given instructions on how to play the game was not comprehensive enough and needs to be further developed.

For the the playability and interaction testing, playing the game on the web browser seem the most challenging as users complained of the touch responsiveness to be a slow. Some users found the IOS version to have a second lag during the jump of the game object, and also during launch of the application.

7   GENERAL CONCLUSION

Usability heuristics has been a standard for determining and asserting quality in software development for few decades.

With the emergence and growth of smart device technology over the years, there has been increase in need for game development on mobile platforms. Different vendors have succeeded in building cutting edge game engines.

This thesis looks into different mobile game development engines with emphasis on Unity engine and HTML5 development platforms. The thesis also researches the different usability concepts and its importance to mobile game development.

A usability testing was done for a mobile application developed using Unity game engine and tested on IOS, Andriod OS and on a browser, the test attempted to compare which mobile platform do users believe has the best user experience.

SOURCES

Adams, E. and Rollings, A. (2010). *Fundamentals of game design*. Berkeley, CA: New Riders.

AI intelligence. (2014). [online] Available at:
http://www.valvesoftware.com/publications/2009/ai_systems_of_l4d_mike_booth.pdf [Accessed 27 Dec. 2014].

Bevan, N. and Macleod, M. (1994). Usability measurement in context. *Behaviour & Information Technology*, 13(1-2), pp.132-145.

Bias, R. and Mayhew, D. (2005). *Cost-justifying usability*. Amsterdam: Morgan Kaufman.

BrainyQuote, (2014). *Game Quotes at BrainyQuote*. [online] Available at:
http://www.brainyquote.com/quotes/keywords/game.html [Accessed 27 Dec. 2014].

Burigat, S, Chittaro, L, & Parlato, E, (2008) Map, diagram, and web page navigation on mobile devices: the effectiveness of zoomable user interfaces with overviews. In proceedings of the 10th international conference on Human computer interaction with mobile devices and services (pp. 147-156). ACM. ISO 1993, ISO CDS 9241 ± 11: *Guidelines for Specifying and Measuring Usability*. [Accessed 27 Dec. 2014].

Chapple, C. (2014). *The top 16 game engines for 2014*. [online] Develop-online.net. Available at:
http://www.develop-online.net/tools-and-tech/the-top-16-game-engines-for-2014/0192302 [Accessed 26 Dec. 2014].

Consumerreports.org, (2012). *Video: The Sony PlayStation Vita handheld game system*. [online] Available at: http://www.consumerreports.org/cro/news/2012/02/video-the-sony-playstation-vita-handheld-game-system/index.htm [Accessed 26 Dec. 2014].

Drucker, P (2006). *Essential Wisdom of Peter Drucker from the Pages of Harvard Business Review*. Boston, MA: Harvard Business School Publishing [Accessed 23 Dec. 2014].

Crowshaw, B. (2010). *On Difficulty Levels*. [online] The Escapist. Available at:
http://www.escapistmagazine.com/articles/view/video-games/columns/extra-

punctuation/7820-On-Difficulty-Levels [Accessed 27 Dec. 2014].

Federoff, M. (2002). *Heuristics and Usability Guidelines for the Creation and Evaluation of FUN in Video Games*. Masters. University Graduate School of Indiana University.

Gerhardt • Powals, J. (1996). Cognitive engineering principles for enhancing human€ • computer performance. *International Journal of Human-Computer Interaction*, 8(2), pp.189-211.

GitHub, (2014). *Game Engines*. [online] Available at: https://github.com/showcases/game-engines [Accessed 26 Dec. 2014].

Harrison, R., Flood, D. and Duce, D. (2013). Usability of mobile applications: literature review and rationale for a new usability model. *J Interact Sci*, 1(1), p.1.

Hocking, J. (2014). *MEAP Edition Manning Early Access Program Unity in Action Version ô€€˜' No books found. Search ag*. Manning Publications Co.

Kastelle, T. and Kastelle, T. (2010). *Innovation for Now and for the Future*. [online] Tim Kastelle. Available at: http://timkastelle.org/blog/2010/08/innovation-for-now-and-for-the-future/ [Accessed 27 Dec. 2014].

Kelly, T. (2014). *Is Mobile Gaming The New Core Gaming?*. [online] TechCrunch. Available at: http://techcrunch.com/2014/11/02/is-mobile-gaming-the-new-core-gaming/ [Accessed 26 Dec. 2014].

Korhonen, H. and Koivisto, E. (2006). Playability heuristics for mobile games. *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services - MobileHCI '06*.

Lin, H., Choong, Y. and Salvendy, G. (1997). A proposed index of usability: A method for comparing the relative usability of different software systems. *Behaviour & Information Technology*, 16(4-5), pp.267-277.

Lindholm, C., Keinonen, T. and Kiljander, H. (2003). *Mobile usability*. New York: McGraw-Hill.

Listoid.com, (2014). *Best Modern Game Platforms - Listoid*. [online] Available at:

http://www.listoid.com/list/73 [Accessed 26 Dec. 2014].

michaelenger, (2013). *Game Engines: How do they work?*. [online] Giantbomb.com. Available at: http://www.giantbomb.com/profile/michaelenger/blog/game-engines-how-do-they-work/101529/ [Accessed 26 Dec. 2014].

Molich, R. and Nielsen, J. (1990). Improving a human-computer dialogue. *Commun. ACM*, 33(3), pp.338-348.

Mozilla Developer Network, (2014). *HTML5*. [online] Available at: https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5 [Accessed 26 Dec. 2014].

Mozilla Developer Network, (2014). *HTML5*. [online] Available at: https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5 [Accessed 26 Dec. 2014].

Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, *Proc. ACM CHI'90 Conf.* (Seattle, WA, 1-5 April), 249-256.

Nielsen, J. (1994b). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*, John Wiley & Sons, New York, NY. [Accessed 27 Dec. 2014].

Nielsen, J. and Molich, R. (1990). Heuristic evaluation of user interfaces. Proc. ACM CHI'90 Conf. 249 - 256. [Accessed 27 Dec. 2014].

Nielsen, J., and Phillips, V. L. (1993). *Estimating the relative usability of two interfaces: Heuristic, formal, and empirical methods compared*. Proc. ACM INTERCHI'93 Conf. 214-221. [Accessed 27 Dec. 2014].

Newheiser, M. (2009). Playing Fair: A Look at Competition in Gaming. [Blog] *Strange Horizon*. Available at: http://www.strangehorizons.com/ [Accessed 27 Dec. 2014].

Rettig, P., Karlins, D. and Wilson, B. (2012). *Professional HTML5 mobile game development*. Indianapolis, IN: WIley Pub. Inc. [Accessed 23 Dec. 2014]

Romar International (2014). *Robot Run game for private client* [Accessed 9 Jan. 2015].

Shackel, B. and Richardson, S. (1991). *Human factors for informatics usability*. Cambridge: Cambridge University Press.

Step.unwto.org, (2014). *Tourism and Poverty Alleviation | Tourism and Poverty Alleviation*. [online] Available at: http://step.unwto.org/content/tourism-and-poverty-alleviation-1 [Accessed 26 Dec. 2014].

Stuart, K. (2014). *Mobile games revenue to overtake consoles in 2015, says analyst*. [online] the Guardian. Available at: http://www.theguardian.com/technology/2014/oct/23/mobile-games-revenue-overtake-consoles-2015 [Accessed 25 Dec. 2014].

SuperData Research, (2014). *Mobile games market*. [online] Available at: http://www.superdataresearch.com/market-data/mobile-games-market/ [Accessed 26 Dec. 2014].

Takahashi, D. (2014). *The DeanBeat: If you blinked, you don't know how big the mobile gaming industry has become*. [online] VentureBeat. Available at: http://venturebeat.com/2014/05/09/the-deanbeat-if-you-blinked-you-dont-know-how-big-the-mobile-gaming-industry-has-become/ [Accessed 26 Dec. 2014].

TECHNOSOFT, P. (2014). *Advantages and Disadvantages of HTML5 tools for cross platform mobile application development - WhaTech*. [online] Whatech.com. Available at: http://www.whatech.com/mobile-apps/press-release/32132-advantages-and-disadvantages-of-html5-tools-for-cross-platform-mobile-application-development [Accessed 26 Dec. 2014].

Techopedia.com, (2014). *What is Platform? - Definition from Techopedia*. [online] Available at: http://www.techopedia.com/definition/3411/platform [Accessed 26 Dec. 2014].

Ward, J. (2008). *What is a Game Engine?- GameCareerGuide.com*. [online] Gamecareerguide.com. Available at: http://www.gamecareerguide.com/features/529/what_is_a_game_.php [Accessed 26 Dec. 2014].

Appendix 1. Pre-Test procedures, pre-test interviews and pre-test tasks

1.1 Pre test Procedures

- Installed game on devices (IOS and Andriod)
- Upload game to a web server (For Browser testing)
- Clear App caches and browser history
- Run application before start to make sure it is fully functional

1.2 Pre-Test Interviews

- Are you a smartphone user?
- Do you know what Operating system runs on your device?
- Do you play mobile games on your phone?
- How often do you play mobile games?
- Any genre of game in particular?
- How long time do you spend playing mobile games on your phone on average?
- Will you pay for a mobile game?

Appendix 2. Test task, Interviews; General testing, Icon testing, Game play experience testing

2.1 Test Task- General testing/Iconography testing

- Launch game on dashboard.
- Start a game
- Return to game's main menu
- Navigate the highscores tab
- Navigate the 'How to play' tab and return to main menu
- Navigate to the about game page and return to main menu
- Read text on main menu's button
- Exit the application

2.2 Test task for gameplay experience Testing for all platforms

- Make the robot on the main menu jump
- Start a new game
- Collect ring points
- Jump barriers
- Find game score during In-play
- Pause game
- End game
- Exit game

2.3 Interviews after the Test

- Did you have the RobotRun game easy to use (On all platforms)
- Will you purchase the game
- Do you know more time to learn how to play the game
- Will you play the game in the future
- State one or two possible improvements to the game
- Do you think more should be do the graphics of the game

- Will you participate in a test like this again.

Appendix 3. Test task results

3.1 Test task results for game play experiences (Playability testing on IOS, Andriod and Chrome on Andriod)

	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5
Make the robot on the main menu jump	✚	✚	✚	━	✚
Colect ring points	✚	✚	✚	✚	✚
Jump Bar-riers	✚	✚	✚	✚	✚
Find game score dur-ing In-play	━	━	✚	━	✚
End Game	✚	✚	✚	✚	✚
Start a new game	✚	✚	✚	✚	✚
Exit Game	✚	✚	✚	✚	✚

3.2 General Usability testing

	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5
Launch a game on the dashboard	✚	✚	✚	✚	✚
Start a new game	✚	✚	✚	✚	✚
Return to game main menu	▬	✚	✚	✚	▬
Navigate the high score tab	✚	✚	✚	✚	✚
Navigate how to play and retun to main menu	✚	✚	✚	✚	✚
Navigate about game and return to main menu	✚	✚	✚	✚	✚
Read text on main	▬	▬	✚	▬	✚

menu's but-ton					

	Participant 6	Participant 7	Participant 8	Participant 9	Participant 10
Launch a game on the dashboard	✚	✚	✚	✚	✚
Start a new game	✚	✚	✚	✚	━
Return to game main menu	✚	✚	✚	✚	✚
Navigate the high score tab	✚	✚	✚	✚	✚
Navigate how to play and retun to main menu	✚	✚	✚	✚	✚
Navigate about game and return to main menu	✚	✚	✚	✚	✚
Read test on main menu's but-ton	✚	━	✚	━	✚