

Bachelor's thesis
Information Technology
Game Development
2015

Pradip Neupane

ESSENTIAL ELEMENTS OF GAME DEVELOPMENT: A CASE STUDY

– EvilHuman Game



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

Pradip Neupane

ESSENTIAL ELEMENTS OF GAME DEVELOPMENT: A CASE STUDY

The game industries have been growing rapidly since the last decade. The main objective of this thesis was to develop a game from scratch to have a deeper look at the game development process. This thesis attempts to answer the following questions; what are the essential elements for game design, how to generate revenue through games by using different monetization models, what are the tools and software available today to create a game, what is game testing methodology and how it is implemented, and, finally, how to manage customer service. To know about whole game design process, one simple mobile game called "EvilHuman" was designed.

A fully functional game was completed using the Unity3D game engine and was published in Google Play Store to receive comments and feedbacks from the users. The game was downloaded between 100 – 500 times. The users who downloaded the game wrote reviews in Google Play Store, and the average rating was 4.6 out of 5. There were many challenges during the practical part of this thesis. The main challenge for this game was learning Unity3D, C# programming, game designing and modeling. However, some assets were used from the premade assets from the Unity3D asset store, which had made the design process easier.

The publishing of this game in Google Play has motivated the author to continue in the game development field because of suggestions and positive feedback, which were provided by the real users who downloaded the game and played it.

KEYWORDS:

Monetization, Game Engine, Steam, Google Play, Unity3D, Assets

CONTENTS

LIST OF ABBREVIATIONS (OR) SYMBOLS	5
1 INTRODUCTION	6
2 GAME DESIGN	7
2.1 Elements of Game Design	8
2.2 Ernest Adams' Views On Game Designing	10
2.3 Design Process of "EvilHuman" Game	13
3 GAME MONETIZATION	14
3.1 Paid Model	14
3.2 Freemium Model	15
3.3 Free-To-Play(F2P)	15
3.4 Banner Advertising	15
3.5 Combination Model	16
3.6 Monetization Model Used in "EvilHuman" Game	16
4 GAME DEVELOPMENT TOOLS	17
4.1 3D Modeling Software	17
4.2 Special Plugins Tools	19
4.3 Game Engines	20
4.4 Game Development Tools Used in "EvilHuman" Game	22
5 GAME TESTING	24
5.1 Planning	24
5.2 Analyzing Game	24
5.3 Testing technique	25
5.4 Testing Method	26
5.5 Implementation	26
5.6 Execution	27
5.7 Evaluation and Documentation	27
5.8 Game Testing For "EvilHuman" Game	28
6 USER MANAGEMENT	29
6.1 System Management Tools	29
6.2 User Acquisition Tools	30
6.3 Behavior Analytics	31
6.4 Engagement and Retention Tools	32
6.5 Monetization Tools	33

6.6 User Management of EvilHuman Game	34
7 CONCLUSION	35
REFERENCES	36

PICTURES

Picture 1. Bartle's Player Types	7
Picture 2. Rules of Good Game Designing	10
Picture 3. Freemium vs Paid Models	14
Picture 4. Blender ScreenShot	18
Picture 5. K-3D ScreenShot	18
Picture 6. Unity3D vs Other Game engines	21
Picture 7. Screenshot of main Menu of "EvilHuman" Game	22
Picture 8. "EvilHuman" Game Play	23
Picture 9. "EvilHuman" Game in Unity3D Scene Mode	23
Picture 10. Player Life Cycle in User Management	29
Picture 11. An Example of System Management Tools	30
Picture 12. Difference Between Heatbeat KPIs and Action KPI	32
Picture 13. Break Down of User Engagement on a Game	33
Picture 14. Facebook Page of "EvilHuman" Game	34

LIST OF ABBREVIATIONS (OR) SYMBOLS

DAU	Daily Active Users
NURBS	Non-uniform Rational B-Spline
KPI	Key Performance Indicator
ARPU	Average Revenue Per Users
Monetization	Way of generating money in a game
Freemium	offering products in a free of charge.
Avatar	Graphical representation of user's or user's characters
Steam	A community platform to distribute and play games
Optimization	to make it perfect as far as possible
Shaders	A set of software instructions used to calculate rendering effects on graphics hardware.
Occlusion Culling	Features in Unity3D which disable rendering the objects if camera does not see the objects
Assets	Components which are used in a game like textures, modeling, animation
AssetsBundling	These are the files which can be export from Unity3D to contain assets of developer's choice
Build Size Stripping	Features in Unity3D which reduces file size of the build apps

1 INTRODUCTION

Game Development is a software development process whose purpose is to design a game. It is a fast growing business and can generate huge revenues in a short period. The time and cost of money to create a game may be influenced by factors, such as scale, platform, and assets. Game development is the only business where the size of a company has no effect on revenue generation. However, large companies can generate more publicity for their games than small companies and this might affect somehow the success of the game.

The idea of this thesis was to design a game and have a deeper look on essential elements of game design, how to generate revenues from monetization models, what are the tools and game engines available today to create a game, what are the requirements of game testing and how it can be implemented and what are the ways to increase web traffic in game and how to manage customer service. To have a deeper look at those essential elements of game design, one simple mobile game called "EvilHuman" was created.

This thesis is structured into six main chapters: (1) Game design elements and Ernest Adams' game design views, (2) Monetization referring to different game monetizing models, (3) Game Development tools analyzing different game engines and modeling software, (4) Game Testing which analyzes different testing methodologies, (5) User Management discussing customer service and managing web traffic for the game and (6) Conclusion.

2 GAME DESIGN

Game design is one of the most important elements in developing games. It is an art of interaction between the players for fun, simulation, educational and many more. Game design is mainly used on games to create goals, challenges, rules and game play. It is a diverse field where the whole process of making a game starts from the game design. Game design needs some designers to create a game based on the players' interests. While designing a game, the target group and the player types need to be analyzed. The proper analyzing of player types of those targeted groups helps to design a good game for them. Bartle has divided player types in four categories: killers, achievers, socializers and explorers [1]. Below follows a description of these player types:

- Killers like to dominate and act to other player. They can dominate other player in a different ways, for example, by bullying them or pressuring.
- Achievers achieve something from the game. For example, they like to score and want to win all the challenges of the game.
- Socializers usually like to interact with other players. These players like to play multiplayer games where they usually spend time in chatting, sharing scores with other players.
- Explorers like to explore the game world. They usually spend time in finding the game environment topology and physics inside a game.

The following diagram shows a graphical representation of "Bartle's Player Types".

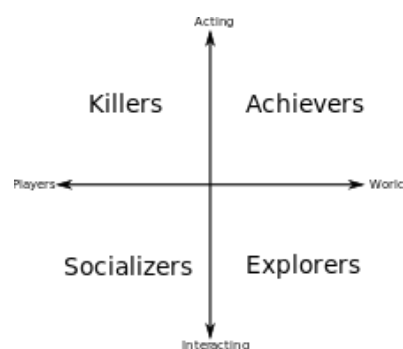


Figure 1. Bartle's Player Type Model [2]

The first section of this chapter describes the elements that are needed for game design process and the second part discusses Ernest Adams' views on game designing.

2.1 Elements of Game Design

Developers may have a good concept of the game which they are going to develop but if they don't focus on the elements of game design, then game might be a failure. There are essential elements of game design that have high impact on making a game successful. Each of them is discussed below [3]:

Space

All games have their own world where they exist which is called space. The game space which makes the game better-looking are lighting, audio, visual effects, colors, sounds, barriers, etc. Depending on the game space, developers can think about suitable characters which may attract the players or the targeted group. Space is also a place of escapism where players can move in a virtual environment to escape from reality.

Goal

A game should have something to offer to players at the end. Simply achieving a high score without any meaningful goal may not attract users to play it. Monotony in a game should be avoided. What this means is that the game looks more attractive when there is more than one-way to defeat the enemies or achieve the goal. Creating an appropriate goal is important in game development. Goals that feel like unbeatable missions, unavoidable traps, unsolvable challenges only frustrate the players and eventually they will stop playing the,. That is why it is better to consider these factors while making game.

Components

Another important element in a game is selecting components. Adding components in a game like characters, weapons, vehicles, etc. gives the players opportunity to play the game in their own way. For example, adding different kind of weapons in a game gives the player the opportunity to select the weapons of their own choice, which motivates them playing game repeatedly. All the components which are in a game should define properly their roles and purpose of existence. Meaningless components that have no purpose in a game play should be avoided as much as possible.

Mechanics

Game mechanics is a game element which connects the players' action with the purpose of the game [4]. This element allows the player to interact with the game during the game play. It describes what player can do, how they can do it, how they can reach achievements and what happens after they reach achievements. For example, in a simple game like shooting enemies when a player shoot the enemies, the game mechanics tells the player about the firing speed (for example bullet), its direction and what happens after the enemies have been shot. By changing the mechanics in a game, the developers can design different levels with different challenges and achievements.

Rules

Rules are one of the essential elements of a game design. Every game has their own rules. Playing a game is following the game rules. The rules define how the game functions, what the players can do, what they cannot do, and what are the ways to reach the achievement in a game. It helps players to play fairly by not letting them cheat. Thus, it increases the fairness in the game. Besides these, rules are also responsible for giving the right information about gameplay and warning if there is violence in a game. For example, children should not play games which contain too much violence and blood. The safety features should be available in game rules before the users play. [5]

The figure below illustrates the elements which are needed for good game design.

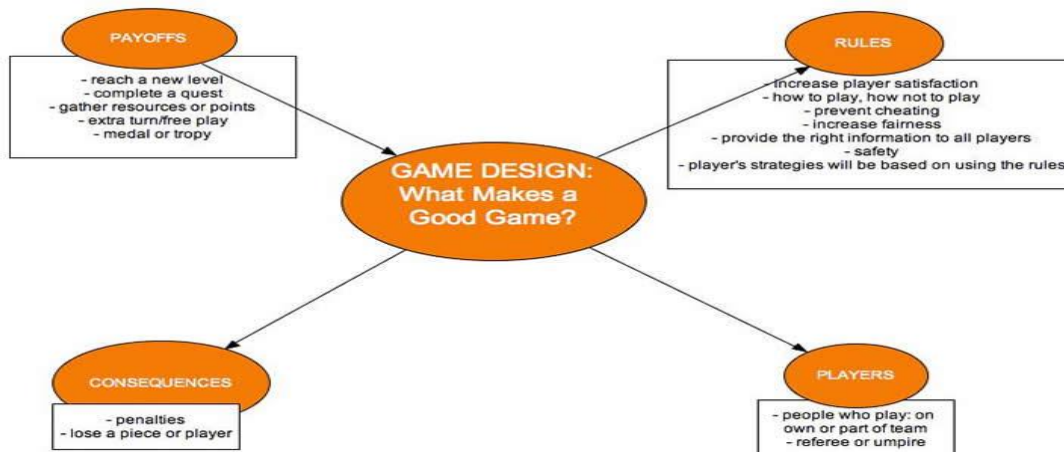


Figure 2. Rules of Good Game design process [6]

2.2 Ernest Adams' Views On Game Designing

Ernest Adams is a game designer consultant, author on game development and founder of the International Game Developers Association (IGDA). He worked in Electronics Art (EA) as a lead designer on a video game called “Madden NFL” from 1993 – 1999. His book on game design is famous and he has written about “Understanding the Players” in his book called “Fundamentals of Game Design”

“The player-centric approach that this book teaches demands, above all else, that you understand your player, not merely as part of an audience of consumers, but as an individual who has an emotional connection to your game and, indirectly, to you.”

Ernest Adam author of Fundamentals of Game Design

His five-factor model and player demographic are going to be described more in detail.

Five-Factor Model

The Five Factor model was introduced by Jason VandenBerghe and is based on human nature. Adams co-related this model with five domains of play, which are explain below.[7]

Novelty

There are two kinds of players, the ones who seek novelty and those who do not. Players who look for novelty in their game often want plenty of variety and unexpected elements in the game. On the other hand, those who do not like novelty often want a familiar environment and do not want much variety inside the game.

Challenge

Challenges mean desire in game. High-challenge players like difficult challenges and want to complete every difficulty that the game has. On the contrary, sandbox games, like GTA, "Assassin Creed II", etc. where the player is free to do and can have fun without required to achieve something, are usually liked by low-challenge players.

Stimulation

In this domain of play, players like to play online with others. This kind of players like to interact with other players and usually like party games. For example "Mario Party: Island Tour" is a good example of party games. On the other hand, those players who avoid stimulation games prefer to play alone.

Harmony

Game is not only about playing and having fun. Some video games can seriously affect psychologically the players. Affecting players in negative or positive way determines the type of video games. Video games can be categorized in to co-operative games, competitive games, etc. Co-operative games bring social harmony and can create the feeling of helping each other. On the other hand, competitive games can bring the feeling of competition and may affect social harmony.

Threat

This is one of the strange domains of play because the player's reaction can be the opposite of what the developers expect from their game. Usually, it is common to generate this kind of negative emotion or feelings in games with high scores in neuroticism in OCEAN (openness to new experiences, conscientiousness, extroversion, agreeableness, and neuroticism) tests. If the developers consider these factors, then it helps to know about what domains they need to focus to make the more entertaining in the game designing phase among those players and can prevent them from developing bad or negative emotions towards it while playing.

Player Demographics

Many people think that females play less video games or they do not like video games at all compared to males. This is the reason why most of the developers do not think about the mechanism inside the game which attracts females. One study carried out by Entertainment Software Association in 2013 shows that 31% of US teenage females play video games while only 19% of the teenage males play video games[7]. That is why the developers must consider females as likely customers for their video games. However, there are differences in the interests of men and women in video games and those differences are usually in their nature and emotions. To make video games attractive to females, there are some rules which developers should think about.

- The game must have a meaningful goal or gamification (simply getting high score for female is not enough).
- Colorlessness, dullness in the game should be avoided (This means that repetition of the same things again and again in the game does not attract females)
- Creating avatars (even though the game is for females it does not mean that developers have to make hypersexualized characters or female character).

However, designing a game for females is not an easy task because of aspects like different emotions and interests, their nature, playing patterns, etc. Females are more fragmented in their interests compared to males. However, the game does not have to be stereotypically feminine. Females pay much attention in visual designing, sounds, story line, characters etc. more than males. [7]

2.3 Design Process of "EvilHuman" Game

The "EvilHuman" game was designed with simple rules and game play. The game was created for learning purposes and that is why a proper analysis of player types before designing the game was not implemented properly. In this game, game components like tigers, guards, players, waterfall, rivers, day and night simulation, rivers, and vegetation are used. All the components in a game have their own roles and importance. The guards and tigers are responsible for guarding the forests. The game has only one level and the mechanism inside the game is that the player will try to cut down the trees and the guards and tigers will try to protect the forest. When tigers see the player cutting the trees, they will give signals to guards by roaring and after that, the guards will start to chase the player and try to kill him/her.

3 GAME MONETIZATION

Game monetization is the process of generating money through a game. Making money with a game used to be a simple and easy process in the past. Nowadays, it is important to know which part of the game will make money, approaching consumers in the right direction, to balance the money-grabbing with enough fun to keep the player onboard and that is before the game is even on sale [8]. To solve which part of the game is making money, proper monetizing is important. There are different monetization models which can make a game successful. Some of them are paid, freemium, free-to-play (F2P), banner advertising and combination of models. The pie-chart below demonstrates the market share of the freemium vs the paid game applications.

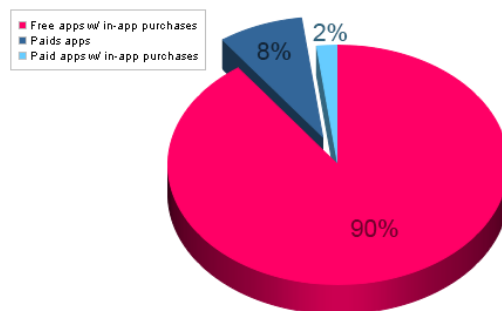


Figure 3. Freemium vs Paid models [9]

3.1 Paid Model

The paid model is the oldest and traditional way of monetization for game app developers. There are many successful games on premium pricing, for example, “Mine Craft”. This is an easy way of monetization and developers sell their apps at a fixed price in the app store. For this model, it is important to create interesting products and develop great game-play before selling it because in this model the users are actually paying money in the beginning. This kind of model is almost nonexistent and nowadays it is hard to be successful. According to research by Flurry, a mobile analytics company says that in 2013, 90% of apps that are used by mobile phone today are free and only 10% have a paid version. [10]

Today the paid model is mostly used for non-gaming apps, for example HD widgets. Here, the developers can offer a free version app with limited features

and the users can try it and when users like it, then they can pay for it. The challenge for this paid model is that there are many alternatives apps, which are similar to the paid version. The question why to purchase arises, That is why it is essential for developers to show the users what is on it and convince them to buy it. Unless the developers solve this challenge, paid monetization has almost zero impact on the success of the game.

3.2 Freemium Model

The freemium model is a pricing strategy where the users can download apps for free with limited features but have to pay extra money if they want to add extra features or gameplay. This is currently the most successful monetization model and this kind of apps usually make more money than the direct pay version but it might take longer time to build the profit.

Freemium apps allow users to download the game in full version but may allow playing with certain restrictions, for example, playing only one level a day, or the user can choose limited weapons, etc. “Candy Crush Saga”, one the most successful games, is a good example of the freemium model. [10]

3.3 Free-To-Play(F2P)

Free-to-play monetization means that the game is free to play at no cost as long as the users wants with all the features available. This kind of game usually has low system requirements because this is a free game and everybody should be able to play it. The entry barrier in this model should be low so that it can reaches the huge mass of players. In this model, the developers make money by selling certain components like extra weapons, power, etc. This model is popular among online multiplayer games, for example “Legion of Legends”. [11]

3.4 Banner Advertising

Ads are one of the effective way to generate money in a game in a fast and easy process [12]. Everything is free in this kind of app but users have to look at too many advertising banners, which are shown while playing the game or at the end

of the game or even at the start of the game. There is a large variety of ads networks available today which has made developers easy and quick to use this model with a minimum risk. While implementing this model, it is the developer's responsibility that the ads shown to users in a game are not annoying to the users. To have an insight into users, developers can use third party tools, like Flurry, unityAds, Chartboost, NativeX that delivers ads to the users according to their interest.

3.5 Combination Model

This monetization is where more than one monetizing models are used within a single app. Some of the good examples of this model are explained below:

- Freemium + paid model was used by Rovio developers in their game called "Angry Birds". Users were given a freemium version of "Angry Birds" with a limited number of levels and but had to pay for the full version of the game.
- Freemium+Free-to-play+Paid+Advertising was used by the "SIMS" game, which was developed by EA. In the start, it was released with a free and paid version of the game. It has free-to-play elements and the players need to develop and gain points, which can be used for home improvements. Alternatively, these points can be bought with money. The free version also contains ads.

3.6 Monetization Model Used in "EvilHuman" Game

"Evil Human" game uses the free-to-play model, so the game is free to play and users do not need to pay money to play this game. The logic behind monetizing this game on this model was that the game was designed for learning purposes and to get personal experience. However, the author of this thesis wanted to monetize the game on free-to-play model so that other people can try the game without paying money and give some feedback and suggestions.

4 GAME DEVELOPMENT TOOLS

Game development tools is software applications that help to create games (including 3D or 2D games). There are numerous game development tools available today that are used to create video games. These game development tools are responsible for creating textures, converting 3D materials, images, creating assets for the game and using the assets inside the game with less coding. Game development tools can mainly be divided into two categories: “3D Modeling Software” and “Game Engines”. This thesis focuses more on 3D game development modeling and game engine software than 2D game development.

4.1 3D Modeling Software

3D modeling software are used to create games assets. If a developer makes 3D games, visual effects like animations, modeling, textures play a great role in attracting players. 3D modeling software is used to create all the game assets, like 3D/2D characters, textures, game environment, skyboxes, etc. Modeling can be divided in to three basic phases: 3D modeling, animations and rendering. Creating 3D modeling is not an easy process and can cost a huge amount of time and money. The software that are used to make 3D models can be expensive. However, there are some good free software tools that are being widely used for modeling. The first part of this chapter describes some paid and free versions of modeling software that are being widely used and the second part describes some special plugins tools that can create 3D characters in a couple of minutes.

Free Versions

3D modeling software can be expensive. However, there are some free software, which have powerful features like paid versions and are being widely used in the modeling industry. Listed below are some of the open source software that are used widely and are free with decent features.

- Blender: it is the most powerful free version of open source 3D modeling software. 3D modeling, UV unwrapping, texturing, raster graphics editing,

rigging and skinning, fluid and smoke simulation, particle simulation, soft body simulation, sculpting, animating, match moving, camera tracking, rendering, video editing and compositing are the features available in Blender. Besides these, it has also its own game engine.[13]

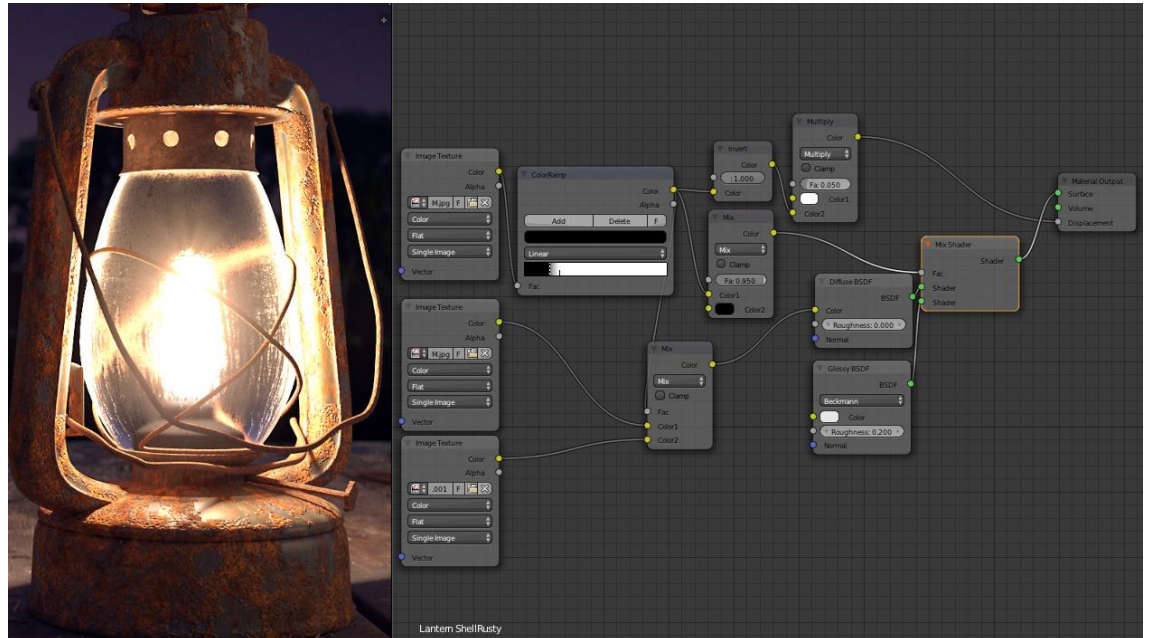


Figure 4. Blender Screenshot [14]

- K-3D is a plugin orientated 3D modeling software and is free to use. This is not as powerful as Blender but it is used for doing low polygon modeling and also contains basic tools for curve, animations, NURBS. [15]

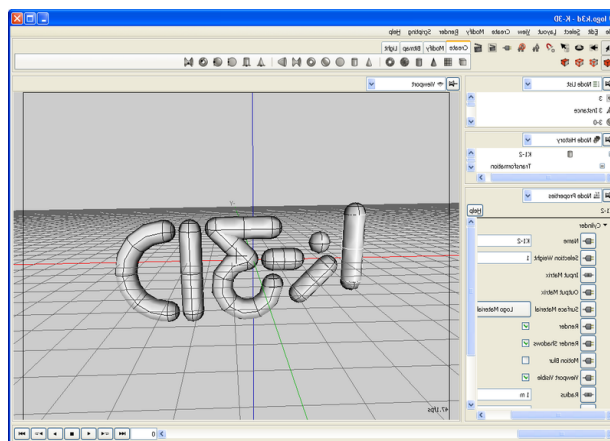


Figure 5. K-3D Screenshot [16]

Paid Versions

Even though there are several free modeling software available in market, they may not support all the features or plugins or they might not have any guarantee, which means that they may crash in the middle of the work. Especially when making 3D games, the visuals effects is one of the main features that attracts the players. Some of the widely used paid versions are 3DMax, Maya, etc. 3DMax and Maya are the products of Autodesk, one of the most powerful 3D software, and have small differences between them. The only difference between these products is that Maya is usually used in the film industry because of its scripting language; MEL is highly customizable and is used for character animation whereas 3DMax is mainly used in the game industry because of its powerful modeling tools. The price for each software is around 3900 euros [17].

Besides these, there are other popular paid versions like Adobe Photoshop, Rhino3D which are also widely used.

4.2 Special Plugins Tools

It is no doubt that creating a character in 3D software takes a huge amount of time and money. For example, 3D max software costs 3900 euros per license, which is expensive for a indie developer. Some special plugins with which we can create character in a couple of minutes are available in the market. Mixamo Fuse and Make Human are examples of those tools for creating 3D models of human characters .

Mixamo Fuse

This good plugin is directly compatible with the Unity3D game engine. The best part of this plugin is that the developer can not only create the character in this platform but also can do rigging within one minute and have many animations available to give life to 3D characters. It is extremely easy to use and has drag and drop features. The drawbacks of this plugin is that its free version has limited features and the professional version can cost up to 1500 euros per year. However, Mixamo Fuse have a student discount policy. Another drawback is that

the developer can create only humanoid character until a fixed date (7.04.2015) with this plugin. [18]

Make Human

Make Human is another tool for creating 3D models within minutes. This is open source software, so it is completely free. The drawback is that it allows to create only 3D humanoid characters with plugins like Mixamo fuse. [19]

4.3 Game Engines

A game engine is a collection of software tools which implements graphical assets that are made from 3D modeling software in a game with minimal coding. Usually, the game engine does common tasks like rendering, collision detection between objects, doing animation to models, physics, etc. User interface, physics, rendering, animating are the components of a game engine. There are many popular game engines available in markets. Some of the top game engines are Unity3D, Unreal, Cry Engine, etc. The first part of this section briefly introduces the Unity3D game engine, which author has used to create the "EvilHuman" game and the second part compares Unity3D and Unreal.

Unity3D

Unity3D is a multiplatform game engine developed by Unity Technologies. The first unity game engine was developed in 2005. Unity version 4, which was released in 2012, was a huge breakthrough for Unity with powerful features including the Mecanim animation system. Since then, it has been popular among the developers and has been established as the number one game engine of today. It has its own asset store where developers can sell and buy game necessary products like models, environment, animations, extensions and plugins, etc.

Until Unity4, there were limited features available in free versions to the users but since Unity5, all the game engine features are free and can be used for

commercial purposes without paying money. However, if the game earns more than 100 000 euros/dollars annually, then users have to buy pro versions.

Unity3D uses C# and JavaScript as programming languages for the game. Before, there was also Unity3D's own script called BOO but it has been depreciated from Unity5 and is no longer available. The new version of Unity5 supports 21 platforms, including Windows, Android, IOS, OSX, BlackBerry, PlayStation, WII U, Xbox 360, Oculus Rift, Web player, Gear VR, etc. The best part of this game engine is that developers develop a game and can export it to all the major platforms with a single click. This makes it easy for the developers and gives many options to publish their game on all the major platforms with optimization features like occlusion culling, asset bundling and build size stripping without wasting time. [20]

Unity3D vs Unreal

Unreal supports all the major platforms like Unity3D. Unreal uses a blueprint while Unity3D has its own asset store where numerous game-related materials can be found. Both Unreal and Unity3D have been free since March 2015. However, Unreal charged money if the game has collected a revenue of 3000 dollars per year while the free Unity3D version can be used until 100 000 dollars revenue has been generated. Here is the bar graph which compares the game engine market of today.

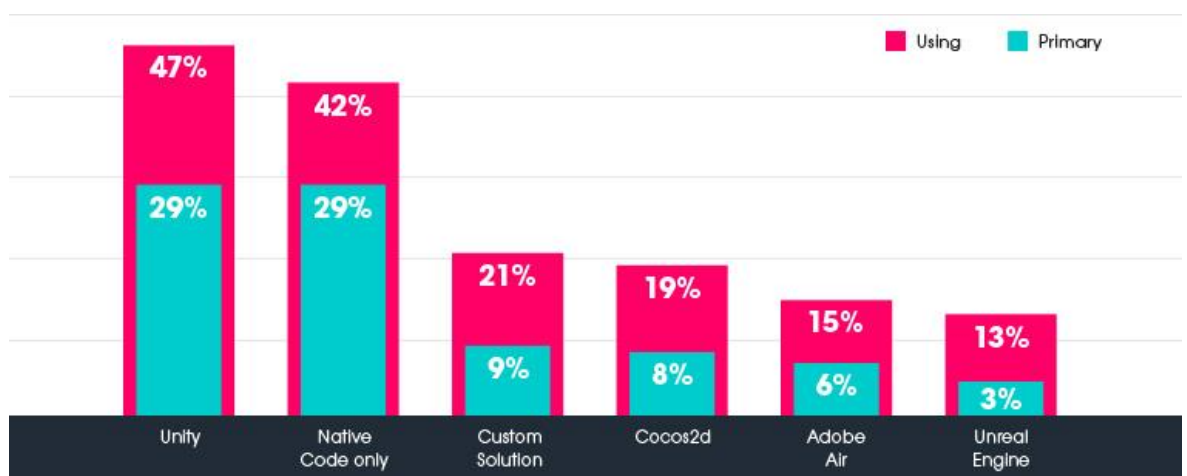


Figure 6. Unity3D vs other game engines market Q3 2014 [21]

The developers who like C++ may become interested in Unreal while C# developers might be interested in Unity3D. However, the needs of a developer also determine the use of game engine. Unreal is good for first person games with its optimization features and Unity3D has strong features on its new UI (user interface), physically based shaders and mecanim animation. [20] [22]

Besides these two game engines, there are other game engines, such as Cry Engine, Source, Xash3D, Rage, etc. which are also widely used.

4.4 Game Development Tools Used in "EvilHuman" Game

The "EvilHuman" game was created using the Unity3D game engine. The modeling of this game components were done using Blender and Unity3D. The game environment was created with the Unity3D Terrain toolkit, which is easy and flexible to use with powerful editing features. The author has also used some plugins like Mixamo Fuse to create the humanoid characters and for rigging. Mixamo Fuse is easy and is a directly compatible plugin with Unity3D. The reason behind choosing Unity3D in this case was because of its powerful features like animation system, user interface, and its asset store where game assets are available in paid and free versions. Some of the screenshot from this game are illustrated below.



Figure 7. Screenshot of Main Menu of EvilHuman Game



Figure 8. Screenshot of Game Play “EvilHuman” Game

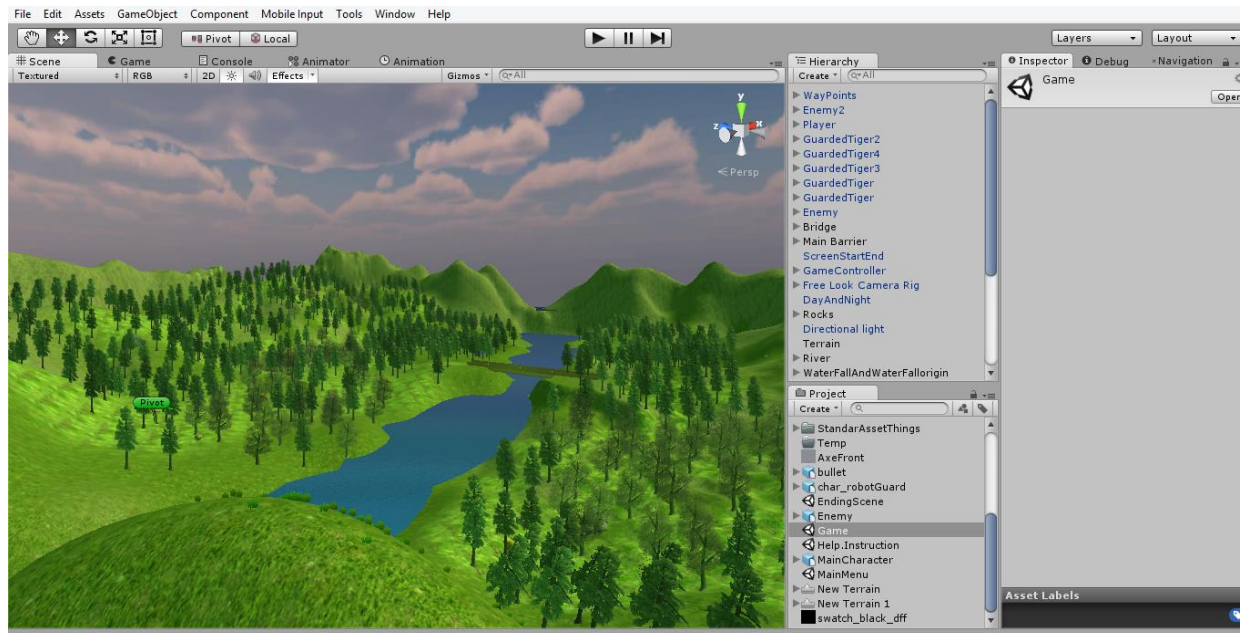


Figure 9. Creating of “EvilHuman” game In Scene Mode.

5 GAME TESTING

Game testing is a process whose main priority is to test the quality of the game, detecting game defects and fixing the bugs. It is important to test the game before releasing it to the users because even a small bug on a game may irritate the users and they may stop playing it. The game may not be tested only for the software bugs but should be tested for other fundamental issues like game play, rules, features, sounds, lights, etc. [23]. The undesirable features like unbeatable mission should be removed immediately during the testing phase. The process of game testing methodology is explained below.

5.1 Planning

In this phase of testing, there should be a plan of the game is requirements, game objectives, like target groups, complexity, surprise, entertaining, etc. The developer can create documents of all those game components, such as features of game, camera view, animation, physics of game, AI, user interface and then it may include the expected results after testing. [24] [25]

5.2 Analyzing Game

Before testing the game, it is good to analyze the quality of the game and decide on what criteria the game is going to be tested. Then the developers can break down game components into smaller testable elements. Some of the examples of breakdown of game components in to small testable components can be:

- Animation (checking the realism and frame rate)
- Sounds and its effect (checking audio drop, skipping, distortion, missing sound effects)
- Camera (checking zoom in, zoom out, replay)
- Scoring levels
- Levels challenges
- Models (like character, textures, game environment)
- User interface (menu functions and how easy to use)
- AI (for both enemy and players about their behavior in response to the game)

- Physics (checking the collision, gravity, etc.)

After this phase, developers should be able to make decisions about what components they are using and what they are removing from a game. [24]

5.3 Testing technique

In the analysis phase, the game components were broken down into small testable elements and the testing phase deals with the techniques and tips of testing to those components and features of game. Some of the techniques and tips for game testing are mentioned below: [24]

- The game should be tested for an entire game not only small part of components.
- Testing the physics in a game, for example, collision between two objects and their impact on one another.
- Testing the loading screen and ensuring that the loading indicator is shown and gives the correct information.
- Ensuring that the hardware does not run out of memory if game is mistakenly left open for a few days.
- The user interface should be simple and easy to use.
- Testing for localization, for example, culture and language.
- If the game is multiplayer, checking the networking and ensuring that the information shown about player in network is correct. For example, if one player kills another, it is good to give information like Player A killed Player B in history logs.
- Ensuring the game is up to date during testing. If developers are building too many versions of game in a day, it is often possible that they will test to wrong version of game.

5.4 Testing Method

After simplifying the game components and features into testable components, there are testing methods which developers can use to test their game. Some of the testing methods are mentioned below. [26]

- **Component-Based Testing:** This testing is more about testing components of game. This testing can be either code-based testing or specification-based testing of those components.
- **Code-Based Testing:** This testing is also known as white box testing and refers to the testing of the internal structure of game, mainly the source code. This testing requires the game tester to have enough knowledge of programming language of that game. This testing may be represented in the form of a graph.
- **Specification-Based Testing:** This testing method is also called black box testing and in this testing method, the game tester does not know anything about the internal structure of game. This testing is performed more from the player perspective. This testing performs a functionality test of game from the player's view.
- **Positive and Negative Testing:** In positive testing, the functionality of game is tested. Positive testing checks that all the functions of the game work perfectly. However, in negative testing, the testing is performed by breaking the game intentionally to see its durability and tolerance resistance.

5.5 Implementation

The implementation phase ensures that every component and feature is checked properly by running different test levels. Some of the test levels are mentioned below. [25]

- **Unit Test:** This test level ensures that all the components of game can run properly without depending on other elements of game.

- Integration test: This test will ensure that all the elements of the game are functioning together according to their roles.
- System Test: In this level, entire game is checked by playing from start to end. This test level ensures that all the components and elements are working perfectly and there are no bugs.
- Acceptance test: After implementing test levels, such as unit test, integration test, system test, the developer is ready to test the acceptance test level. In this test level, the game is shown to other players and ask for feedback and comments from them about the entire gameplay and functioning. In this test level, the developer can ask the users some questions to enhance the game play. Example questions can be: What was your favorite part? Did you have fun? Was it easy to play? Do you want to play again?

5.6 Execution

Having run different test methods and different test levels in a game, now game testers should be able to know what they are going to test, when they are going to test and how they will test it. The execution phase will execute the actual testing of game. While performing any kind of testing, it is important to create documents, for example date, what version of game, etc. for future reference. It is advised that the game tester should make complete tests rather than trying to fix issues immediately. After this phase of testing, below are some of the questions, which need to be answered:

- a. What was the cause of issues?
- b. How can it be fixed?
- c. D the issues occur frequently or rarely in game?
- d. How can we prevent this issues from appearing in the future?

5.7 Evaluation and Documentation

After completing all the testing and making ensure that the game is working fine and every bug and other issues are fixed, there is still work, namely evaluation

and documentation, which every developers have to do for future reference. Below are some of the questions, which help to developers to make good documentation and evaluation of their game:

- Was the requirement of game met and how?
- Was the game complete and what were the testing methods used to test the game?
- Can the game be used for other target group and community, and if not why?
- What part of the game did users like most and if they did not like, was it because of game itself or player preference?
- What could be done as next steps to ensure more quality of game?

5.8 Game Testing For “EvilHuman” Game

In “EvilHuman” game, there was not much done in game testing phase because of time limits. However, some online and face-to-face suggestions and feedback were asked from friends and teachers on rules, gameplay, and on some other fundamental issues, like characters, animations, physics and camera movement. The author has also implemented some small testing on user interface and on overall game play and rules with his teacher. Below are some of the questions which were asked with friends while doing testing for this game.

- How is the FPS rate ?
- Is the user interface friendly and easy to use ?
- How is the game play and is the level difficult to challenges ?
- How is the physics of game ?
- Is the character animation behaving in a correct way?

6 USER MANAGEMENT

When the game is ready to launch, user management worries game developers. Certain questions, for example, how to handle customer service, how to get more users, what kind of analytics tools should be used, how to integrate ads in game, etc. are often asked. There are five different life cycle steps, which can support mobile game development; they are system management tools, user acquisition tools, behavior analytics, engagement and retention tools, and monetization tools. [27] User management of a player's life cycle is explained in figure below.



Figure 10. Player Life Cycle [28]

6.1 System Management Tools

If the game is an online game or a multiplayer online game, it is required to have the server up and serve the players 24/7. Maintaining the server up all the time is quite challenging because the number of players will grow more and more eventually. The minor bugs like system performance, errors, and technical attacks in server are frequent in online games. However, if these problems are frequent, then they will seriously damage KPIs (Key Performance Indicators). To solve this problem, it is advised to have log management service which will monitor all the system performance, and developers can fix the problem easily in case the problem occurs. The picture below describes an example of organized system management tools.

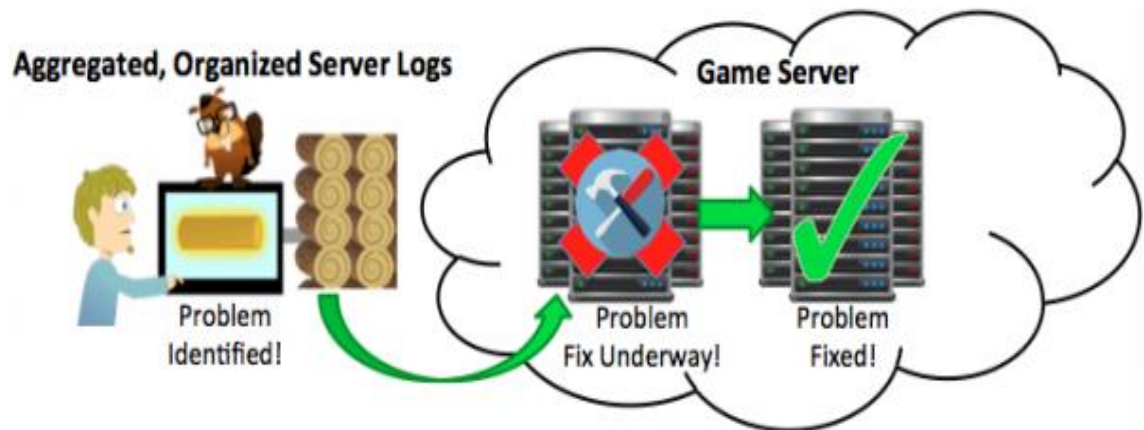


Figure 11. An example of System Management tools [29]

6.2 User Acquisition Tools

User acquisition is a challenging process because developers need to have an idea who is their target group and what is the way of attracting the players in their game. There are certain tools which can increase the web traffic of players in visiting the gameplay. Some of them are mentioned below:

Social Media

Social media is the favorite choice of all developers. Facebook, Twitter, Google Plus are some of the social media channels where developers can publish their game trailers, information, etc. However, simply registering in those social media sites and publishing the game is not enough. Below are some of the ways that developers can use social media to increase web traffic in their game: [30]

- Improving online profile by giving information about company and its goal and and keep updating it.
- Taking advantages of available social media application like hootsuite, hubspot, socialBro, etc. to analyze the performance and promote the game.
- Creating more resource material like tutorials, case studies, blogs, etc. and sharing them

- Finding the right audience is important. There are some monitoring tools like SocialMention, Delicious, Feedly, GatherContent which help to find the right audience for the game.
- While posting in social media on the behalf of game, developers should never mix professional and personal matters, ensure they will never share any confidential information, and should not create any dispute while commenting on posts.
- Establishing good communication with customers, listening to their feedback and suggestions and monitoring their reviews on game for example, developers can use Google Analytics tools helps to increase web traffic to the game.

Advertising Tools

For mobile games, there are many advertising tools that can offer advertisements to well-targeted users. Some of the advertisement offers include reward ads, direct deals, ad mediation, native ads, rich media ads, targeting segmentation and attribution and game specific ad offering.

6.3 Behavior Analytics

Most of the developers may not have a clear idea about what they should track in the players with what kind of tools, what kind of data they should collect, why they should collect data, and how they should use those data. This behavior analytics helps to solve the problem what to collect and how the data should be handled. Different users have different ways of responding to the game, its content, and game play. It is important for developers to work hard to bring the positive behaviors on users towards their game. There are two kind of KPIs for game userbehavior analytics, Heartbeat KPIs and Actionable KPIs.

Heartbeat KPIs are those tools which gives information about DAU (Daily Active Users) and ARPU (Average Revenue Per Users). These KPIs do not give deep information and only give the basic overview of general game health situation. On the contrary, Actionable KPIs are specific on gameplay and give the information

of key user behavior and this KPIs is more important than Heartbeat KPIs. [27]
The figure figure describes the difference between these two KPIs.

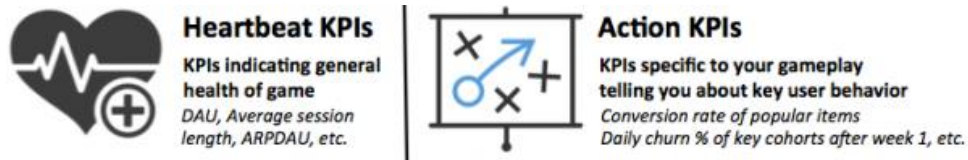


Figure 12. Difference between Heartbeat KPIs and Action KPIs [31]

6.4 Engagement and Retention Tools

To make a successful game among the users, engaging and retaining is important [32]. Creating a player segment and deciding what features would be the best is quite challenging. Honeylizer is a tool which the developers can use to decide how to segment the players and what features should be used so that the users can be attracted to the game.

There are some ways to attract players to engage in their games. They are explained below: [33]

- It is good to give some hints to the players to solve the challenges.
- Every game may not have a story narrative behind them. However, telling story at the start of the game gives the background information about the game to the users, which might interest the players.
- It is a good practice to reward the players by giving some game coins or in some other way. For example, giving more points when they complete challenges is another way to engage players in the game.
- Creating uncertainty in the game is another good way to attract players. What this means is giving surprise rewards, more rewards for greater risk, etc.
- The user interface (UI) is another element that plays an important role in attracting players. UI should be easy to use. UI should also give all the information about game, achievement, players rewards, etc.

A good example of user engagement breakdown in a game is explained in the figure below.

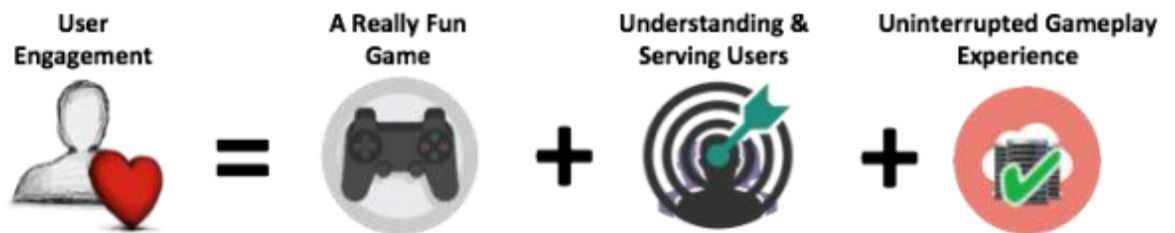


Figure 13. Break down of user Engagement on a game [34]

6.5 Monetization Tools

All the model of monetization has been already discussed in Chapter 3. However, this section discusses how to add advertisements in a game without making the players feel annoyed. Usually developers only think about using the ads in their game but they do not care much whether the ads are relevant to the users or not. The ads that usually come in the middle of game are not interesting for the players anyway. Below are some of the points, which are focused on the integration of ads in mobile games.

- Embed the native ads directly on the UI so that they become fluid to the screen size, can fit in the screen, and do not disturb the players.
- Give reward to the users when they click the ad.
- Provide ads in which users is interested. For this feature, developers need to integrate ads with third party tools like Flurry, unityAds, Chartboost, NativeX, etc.. These third party software deliver ads to the users according to the users' interest which makes it less annoying to the users.

6.6 User Management of EvilHuman Game

As the “EvilHuman” game was designed for learning purposes, the user management was not implemented at all. However, a Facebook page was created for the users to tell them about the coming game updates, and solve their problems occurring while playing game. Below is a screen shot of the Facebook page which was used to manage and solve the users problem for “EvilHuman” game



Figure 14. Screenshot of the EvilHuman Facebook Page.

7 CONCLUSION

During the last decade, there has been a huge development in the game development industry. Plenty of available game engines, modeling software and freely accessible tutorials have made it easy to develop a game. Nowadays, thousands of games are published every day in game stores, like Google Play, IOS, Windows, Steam etc. This has given more options to users, and that is why many users are not playing the same game for a long time.

This thesis also compares some of the modern tools and technologies that are being widely used in game development and it demonstrates with some examples of tools that are used to make a game. Even though there are many tools and elements available today to create a game, still all the players, are not similar and they may respond in a different way to the same game.

The thesis also describes the case study of the HumanEvil game that was designed for learning purposes and demonstrates the essential elements in a game development process. That is why, only an idea is not enough to design a game. It needs proper target player analysis, monetization for generating revenues, managing users, and proper game testing to make it a success.

REFERENCES

- [1] Luimula, M., Abdulghani, B. and Smed, J . 2014. Bartle's Player Types. [ONLINE] Available at:
https://moodle2.utu.fi/pluginfile.php/318494/mod_resource/content/1/Gamification_141124.pdf
 [Accessed: 18 May 15].
- [2] Ramiromagalhaes. 2013. Bartle's Player Types [ONLINE]. Available at:
http://en.wikipedia.org/wiki/Bartle_Test#/media/File:Character_theory_chart.svg [Accessed: 18 May 15]
- [3] Make School. 5 Basic Elements of Game Design. [ONLINE] Available at:
<https://www.makeschool.com/gamernews/298/5-basic-elements-of-game-design> [Accessed: 5 May 15]
- [4] Schell, J. 2014. The Art of Game Design: A Book of Lenses. 2nd ed. Florida: Taylor & Francis.
- [5] Salen, K., Zimmerman. E. 2004. Rules of Play - Game Design Fundamentals. 1st ed. Massachusetts London, England: The MIT Press Cambridge
- [6] Glendale School, 2008, Game Design [ONLINE]. Available at:
http://projects.cbe.ab.ca/glendale/showcase/normal09/images/Game%20Design-0_1.jpg
 [Accessed: 13 April 15].
- [7] Adams, E. 2014. Fundamentals of Game Design,. 3rd ed. San Francisco: New Riders Press.
- [8] Rose, M, 2013. Understanding the realities of video game monetization. [ONLINE] Available at:
http://www.gamasutra.com/view/news/205412/Understanding_the_realities_of_video_game_monetization.php [Accessed: 25 March 15].
- [9] Hailay.L. 2013, Freemium vs Paid Model [ONLINE]. Available at:
<http://www.lukehailay.com/blog/wp-content/uploads/2013/12/freemium-vs-paid-app-strategy.png> [Accessed: 09 April 15].
- [10] Rudenko, E. 2014. Understanding Your Options When Monetizing Mobile Games. [ONLINE] Available at: <http://appdeveloper magazine.com/1275/2014/4/1/Understanding-Your-Options-When-Monetizing-Mobile-Games/> [Accessed: 18 May 15].
- [11] Weidemann, T. 2014. Beginnes's Intro to F2P, Aalto Game Monetization Design. [Online Video]. 23 July. Available from: <https://www.youtube.com/watch?v=fdzk7x3W3kE> [Accessed: 18 May 2015].
- [12] Earn Revenue from AdMob Ads | Android Developers. [ONLINE] Available at:
<https://developer.android.com/distribute/monetize/ads.html> [Accessed: 29 May 2015].
- [13] Blender. 2015. Features. [ONLINE] Available at: <https://www.blender.org/features/> [Accessed: 5 April 15].
- [14] Blender. 2012, Blender ScreenShot [ONLINE]. Available at: http://www.blender.org/wp-content/uploads/2012/11/materials_screenshot.png [Accessed: 18 May 15].

- [15] K-3D. 2010. About. [ONLINE] Available at: <http://www.k-3d.org/node/1> [Accessed: 7 April 15].
- [16] K-3D community, 2009, K-3D screenshot [ONLINE]. Available at: http://commons.wikimedia.org/wiki/File:K3d_software_screenshot.png [Accessed: 18 May 15].
- [17] Autodesk store. 2015. Autodesk store. [ONLINE] Available at: http://store.autodesk.eu/store/adsk/en_IE/html/pbPage.All-Product-Listing_en_IE [Accessed: 7 April 15].
- [18] 3D Character Creator, Fuse - Mixamo. 2015. 3D Character Creator, Fuse - Mixamo. [ONLINE] Available at: <https://www.mixamo.com/fuse> [Accessed: 7 April 2015].
- [19] Makehuman | Open source tool for making 3D characters. 2015. Makehuman | Open source tool for making 3D characters. [ONLINE] Available at: <http://www.makehuman.org/> [Accessed: 7 April 2015].
- [20] Unity - Multiplatform - Publish your game to over 10 platforms., 2015. Unity - Multiplatform – Publish your game to over 10 platforms. [ONLINE] Available at: <http://unity3d.com/unity/multiplatform> [Accessed: 1 May 2015].
- [21] Unity3D, 2014, Unity3D vs Other Game Engines [ONLINE]. Available at: <https://unity3d.com/profiles/Unity3D/themes/unity/images/company/pr/vision-mobile-graph.jpg> [Accessed: 28 April 15].
- [22] Unreal Engine 4 Release & Royalty Tracking Guidelines., 2015. Unreal Engine 4 Release & Royalty Tracking Guidelines. [ONLINE] Available at: <https://www.unrealengine.com/release>[Accessed: 1 May 2015].
- [23] SmartBear. 2013. Testing Your Mobile Games: Introduction. [ONLINE] Available at: <http://www2.smartbear.com/rs/smartbear/images/Mobile-Game-Testing-eBook-Smartbear.pdf>[Accessed: 18 May 15].
- [24] Scribd / api-3698888., 2008. Game Testing Methodology. [ONLINE] Available at: <http://www.scribd.com/doc/6555758/Game-Testing-Methodology> [Accessed: 24 May 15].
- [25] Hjelmberg, N. 2014. Testing Methodology Applied to Games. [ONLINE] Available at: <https://boardgamegeek.com/thread/1228786/testing-methodology-applied-games> [Accessed: 24 May 15].
- [26] Redavid, C. and Farid, A., An Overview of Game Testing Techniques.[ONLINE] Available http://www.idt.mdh.se/kurser/ct3340/ht11/MINICONFERENCE/FinalPapers/ircse11_submission_15.pdf [Accessed: 27 May 15].
- [27] Gamesauce. 2014. a comprehensive analysis of the tools that support mobile game development (part 1). [ONLINE] Available at: <http://gamesauce.org/news/2014/09/10/a-comprehensive-analysis-of-the-tools-that-support-mobile-game-development-part-1/> [Accessed: 18 May 15].
- [28] Industry Contributions, 2014. Player's Life Cycle [ONLINE]. Available at: <http://gamesauce.org/news/wp-content/uploads/2014/08/14-600x161.png> [Accessed: 13 April 15].

- [29] Industry Contributions, 2014, An example of system management tools [ONLINE]. Available at: <http://gamesauce.org/news/wp-content/uploads/2014/08/6-600x190.png> [Accessed: 13 April 2015].
- [30] Six ways to raise your game on social media | Marketing Donut. [ONLINE] Available at: <http://www.marketingdonut.co.uk/marketing/online-marketing/six-ways-to-raise-your-game-on-social-media> [Accessed: 20 May 2015].
- [31] Industry Contributions, 2014, Heartbeat vs. Actionable KPIs [ONLINE]. Available at: <http://gamesauce.org/news/wp-content/uploads/2014/09/16-600x98.png> [Accessed: 13 April 15].
- [32] Engage & Retain Users | Android Developers. [ONLINE] Available at: <https://developer.android.com/distribute/engage/index.html> [Accessed: 29 May 2015].
- [33] Karl, M. K., Five Elements that Make Games Engaging. [ONLINE] Available at: <http://www.scogatt.eu/files/Supporting%20Documents%20for%20Teachers/10%20Five%20Elements%20that%20make%20SCG%20Engaging.pdf> [Accessed: 21 May 15].
- [34] Industry Contributions, 2014, Break down of user Engagement on a game [ONLINE]. Available at: <http://gamesauce.org/news/wp-content/uploads/2014/09/211-600x125.png> [Accessed: 13 April 15].

