SAVONIA

Mobile game development, mobile market insights and animation methods.

Stanislav Belov

Bachelor's Thesis

___.___.______ ________________________________

Bachelor's degree (UAS)

SAVONIA UNIVERSITY OF APPLIED SCIENCES **THESIS**
Abstract

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Stanislav Belov			
Title of Thesis Mobile Game Development, Mobile Application Markets and Animation Principles			
Date	15 May 2016	Pages/Appendices	50
Supervisor(s) Mikko Pääkkönen, ohjelmistosuunnittelija, Jussi Koistinen, lehtori			
Client Organisation/Partners			

Abstract

It is not a secret that today almost every person possesses a mobile device that runs a variety of applications. Games are the dominating app category, hence, the mobile game development industry grows and develops every day. Because of that, the purpose of the thesis was to study mobile application markets, animation methods and the animation process in the mobile game development.

To begin with, some aspects of the mobile game development process are usually underestimated by the developers, therefore, the obstacles that a beginner mobile game developer faces were revealed based on personal experience. Secondly, the thesis investigated the mobile application markets by studying Apple App Store history and statistic data about it. Next, the study of 2D and 3D mobile game animating was done, including methods, principles, and techniques that are used in game development process. Finally, the methods of animation creation were implemented in a practical example.

As a result, the thesis brought the insights of Apple App Store market which represents a great value for mobile app projects. In addition, the animation process was thoroughly studied, implemented and carried out throughout the thesis. Thus, animation expertise was obtained.

Keywords
Apple App Store, Mobile Game Development, 2D, 3D, animation.

CONTENTS

1 INTRODUCTION

Today almost 60 percent of the population worldwide possesses a mobile phone or another mobile device. Each person is spending over 85 percent of their mobile usage time using various mobile applications. Consequently, the mobile application market becomes a billion-dollar industry where the mobile games occupy the biggest percentage of the market. Because of that, the mobile game development industry itself grows at an enormous speed. It becomes harder and harder for new developers to achieve success and earn recognition amongst the mobile users.

First of all, the thesis will try to cover the basic problems that a beginner developer is facing when starting his/her career based on professional experience. Then, another goal of this thesis is to research the Apple App Store mobile application market and collect valuable information what will be beneficial for the mobile game development.

Finally, a big part of this thesis is going to be dedicated to game animation. When developing a mobile game, the developer is always limited by a number of resources. The animation in mobile games occupies a core part when it comes to development. It requires a lot of technical skills and time, as well as the understanding of how to create a correct and plausible illusion in order to achieve best results. The optimization of the mobile game development process can be done through game animation. However, it is required to research and understand the animation principles and different types of animation methods. Furthermore, it is necessary to investigate animation methodology and implement these animation methods in practice.

2 IMPORTANT ASPECTS OF MOBILE GAME DEVELOPMENT

Today the mobile game development industry is one of the most profitable industries in the world. The amount of games that have been developed since the first launch of the App Stores could not be counted by any means. More and more companies and corporations are trying to step into the mobile game industry. There are a lot of small and big game development studios who try to succeed by creating innovative game designs and game ideas. The industry itself offers a lot of possibilities for unique ideas and new games that try to penetrate the market.

If an individual decides to create mobile games, it could be a challenging but at the same time an exciting project. However, based on personal professional experience obtained through the development process, there are core aspects that a person should know and understand when it comes to mobile game development.

2.1 Lack of knowledge about the target market

The first barrier in mobile game development that a person would hit is usually going to be a lack of education about the mobile game industry and the mobile game market trends. The market research is an important part of the development process because it helps to identify opportunities in the market, minimize risks, plan ahead and uncover potential problems for an ongoing or future projects.

2.2 Project management

The second obstacle that could become a major issue in mobile game development is project management. Project management is a specific field that is taught at the university level because it is a complex discipline that requires thorough understanding. A short interpretation of project management would be:

- Planning - Specifying the desired results, determining the schedules and estimating the resources needed.
- Organizing - Defining people’s roles and responsibilities.
- Controlling - Monitoring actions and results, addressing problems and sharing information.

Project management always becomes a big problem for beginner game developers because there is a big chance that a studio would start without a qualified project manager and learn by mistakes throughout the development process. It is necessary to underline that team management and communication problems also affect project management processes.

2.2.1 Team management

The size of the team always matters. The bigger the team the faster the project could be finished since it has more resources and work power. The benefit of working in a big team is that the most experienced colleagues will always provide proper help and advice on a problem that person is struggling with. However, big teams always have communication problems and growing deadlines, all because of the size, but with a good project manager, these problems could be avoided.

Small sized teams, however, are relatively easier to manage. Sometimes they even do not have a project manager behind. The communication goes very easily because most of the time small team works together and do not require regular meetings.

2.2.2 Communication problems

The next issue that strikes a beginner mobile game developer down is the communication problem. Sometimes communication within the team is ignored and not took into consideration seriously. Consequently, people could bring new features into the project without discussing them amongst the colleagues. The result of that could have a really big influence on the project. That is why there are regular meetings and team discussions in order to keep everything intact. There could be other communication problems within a team when there is a conflict between individuals. That would be the job of the project manager to solve that kind of conflict and bring the team together.

2.3 Skill barrier

Then there is a skill barrier in mobile game development but it is the least of the issues that should concern a beginner developer. Skills are obtained through experience, as well as through studies and regular training.

2.4 Mobile game development process optimization

A crucial part of the development process is an optimization of that process. Projects usually do not possess enough resources or time to fit inside the work plan. Exploring new methods of development or processes that can beneficially affect the project are always a big priority for the development team. Even experienced professionals are constantly in search of new techniques, methods and ways to reduce the resources needed to execute a project.

3 INTRODUCTION TO MOBILE APP STORES

When working as a game developer one of the tasks inside the team includes researching the markets and bringing the insight to the client or the team. As mentioned above the lack of knowledge about mobile game market could drastically affect the project success.

Today in the world there are two leading mobile app stores: Google Play and Apple App Store. Google Play is the official android application market for mobile devices with a big variety of digital content such as mobile apps, books, music and video streaming services. Apple App Store is the official iOS application market for Apple mobile devices featuring the same type of content as mentioned above for Google Play.

It is worth mentioning that the mobile application market is actually highly saturated by a variety of third-party app stores which have different revenues, publishing policies and their target markets. They include manufacturer-specific app stores such as Samsung Apps, LG Smart World, Motorola Shop4Apps, Dell Mobile App Store, CISCO App HQ, Docomo Market and Lenovo App Store. More than that there are Operator/Carrier app stores which are Verizon App Store, Vodafone AppSelect, China Mobile, T-Mobile Mall, Sprint, Aircell PocketApps, Airtel App Central and many others. Nevertheless, they still fade near Google Play and App Store. (Cooper 2012)

Apple App Store is the actual target market for future projects, so that is why the current research was compiled. The chapter below will present Apple App Store market insights.

3.1 Inside the Apple App Store

No one ever imagined that the mobile applications would occupy such a big part of human life. In a matter of years, Apple App Store became the leading mobile application market in the world. In order to understand how the Apple App Store functions, the history of Apple App Store was studied thoroughly.

On July 11, 2008, Apple launched App Store with total of 500 apps available for iPhone. By June 8, 2015, the amount of apps available for download exceeds 1.4 million with over 100 billion app downloads. The CEO of Apple gave a public speech at The Apple Worldwide Developers Conference.

> “It’s hard to believe that the App store was launched only 7 years ago, it’s hard to remember the day without it. I am happy to announce that the App Store recently passed a major milestone. The App Store has passed 100 billion apps downloads. The rate of growth and the momentum is absolutely staggering; the industry has never seen anything like this before. The App store has forever changed software and software distribution and it’s also being an economic boon. We’ve now payed 30 billion dollars to developers. The App Store continues to be the most profitable App Marketplace on the planet. “(Cook 2015)

On July 11, 2008, Apple App Store had 500 apps available for download, according to Mr. Jobs, 25 percent of 500 applications would be free and 90 percent of commercial applications would be sold for $9.99 or less. The initial revenue split deal for developers was 30 to 70 where Apple would take 30 percent of the profit and provide marketing and distribution for developers. (McChan 2012) Early metrics showed that the prices for apps were dropping rapidly. The reason for that was the classic low cost strategy in new highly competitive market. It was a trend that is still used in many industries. (Porter 1980, 3 – 44)

July 14, 2008, only 3 days from the launch had passed and the total number of the apps increased by 40 percent, from 500 to 800, as well as there were already 10 million app downloads. (Bowcock 2008) August 2008, within the month users had downloaded 60 million apps. However, big amount of these apps were free but Apple still sold an average of $1 million per day. (Wingfield 2008) The CEO of Apple commented that:

> “If sales stay at the current pace, Apple stands to reap at least $360 million a year in new revenue from the App Store. This thing's going to crest a half a billion, soon," he added. "Who knows, maybe it will be a $1 billion marketplace at some point in time. I’ve never seen anything like this in my career for software.” (Jobs 2008)

By October of 2008, the number of downloads was 200 million with only 7 500 apps available for download. (McChan 2012)

In the beginning of 2009, Apple App Store reached half million downloads having 15 000 apps available in the store. (McChan 2012) April 23, 2009, the number of half billion app downloads was reached and 35000 apps were available in 19 different categories. In July of 2009 was the one-year anniversary of Apple App Store. A number of downloads were already 1.5 billion and 65 000 apps were available in Apple App Store. Furthermore, by that time, more than 100 000 developers joined Apple’s developer program. (McChan 2012) The growth that Apple achieved within one year was tremendous. A new era of applications and smartphones had begun. In September of 2009, Apple App Store achieved the point of 2 billion downloads and the total of 85 000 apps that were present in Apple App Store. The Apple App Store growing speed, user acquisition and development was extraordinary. (McChan 2012) In November of 2009, 100 000 apps were available for download in Apple App Store.

During that time Pinch Media, the company which created analytic software for app developers gave a presentation at the New York iPhone Developers Meeting where they showed metrics from more than 30 million Apple App Store downloads of apps which were using Pinch Media’s analytics libraries. The analytic data from Pinch Media's showed that if an application lowered its price it resulted in a 130 percent rise in sales demand, however, increasing the application price resulted in 25 percent decrease in demand. More than that, Pinch Media discovered the fact that only about 1 percent of downloaded apps remained consistently used and only 20 percent users of free apps used it more than one day. As for the paid apps they had better user retention but still they tended to get less and less app usage. Also, the data showed that game apps engaged users to spend more time within the app than other types of apps. (Slivka 2009)

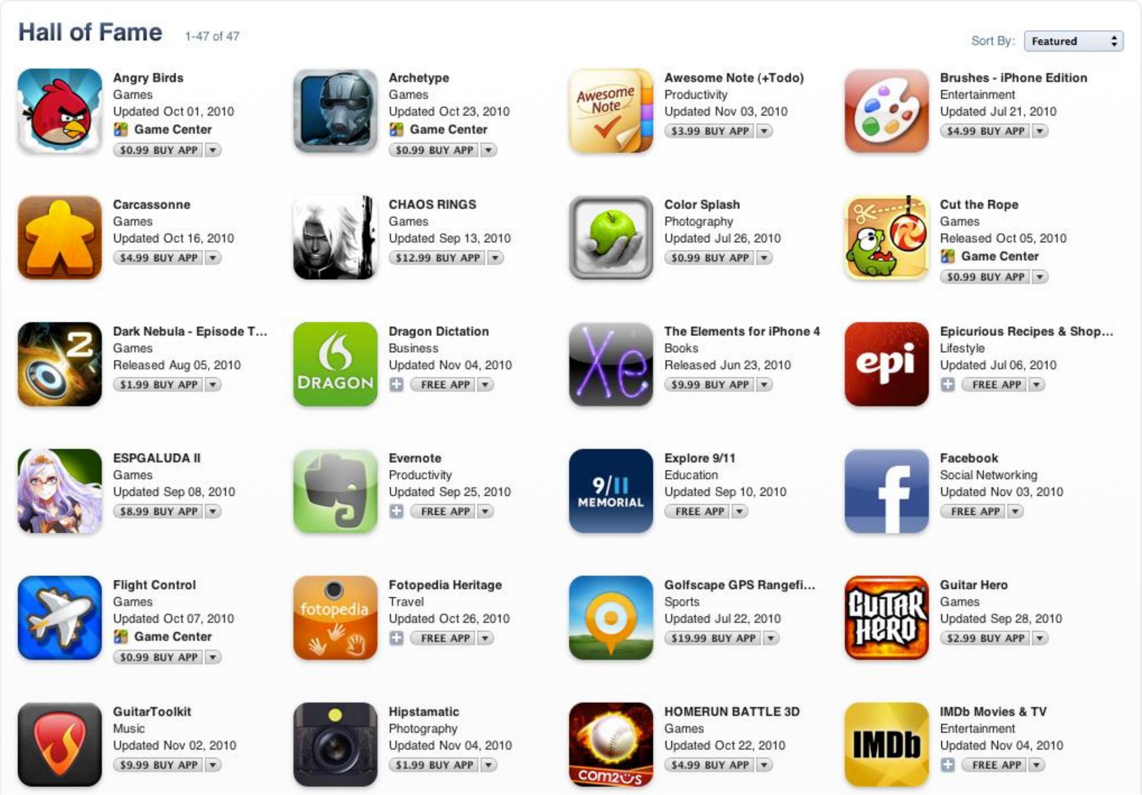
On January 5 in 2010, Apple announced that over 3 billion apps had been downloaded from the app store.

> “Three billion applications downloaded in less than 18 months—this is like nothing we’ve ever seen before. The revolutionary App Store offers iPhone and iPod touch users an experience unlike anything else available on other mobile devices, and we see no signs of the competition catching up anytime soon.” (Jobs 2010)

In April of 2010, Apple App Store hit 4.5 billion downloads and 200 000 apps were available for download. Plus, Apple released iPad (the first Apple tablet) with over 5000 apps available for it. Within only one month iPad generated 12 million downloads of apps. (McChan 2012, 4) In September of 2010 downloads were constantly rising, App Store reached 6.5 billion app downloads and provided 250 000 apps for download. (McChan 2012, 4) This was the year when Apple released their app review documentation “App Store Review Guidelines”. This was done in order to decrease the number of low-quality apps and to restrict the store from inappropriate apps. In their guidelines Apple openly stated:

> “We have over 250 000 apps in the App Store. We don’t need any more Fart apps. If app doesn’t do something useful or provide some form of lasting entertainment, it may not be accepted”. (Apple App Store Review Guidelines 2012, 1)

In November of 2010 Apple introduced a new element to their Apple App Store called “App Store Essentials: Hall of Fame”. The purpose of it was to show the best of the best apps for iPhone and iPad. Free and paid apps were both in that section. (Ritchie 2010) Image 1 presents the first apps that were included in the Apple App Store Essentials: Hall of Fame”.



Hall of Fame 1-47 of 47

Sort By: Featured

Angry Birds
Games
Updated Oct 01, 2010
Game Center
$0.99 BUY APP

Archetype
Games
Updated Oct 23, 2010
Game Center
$0.99 BUY APP

Awesome Note (+Todo)
Productivity
Updated Nov 03, 2010
$3.99 BUY APP

Brushes - iPhone Edition
Entertainment
Updated Jul 21, 2010
$4.99 BUY APP

Carcassonne
Games
Updated Oct 16, 2010
$4.99 BUY APP

CHAOS RINGS
Games
Updated Sep 13, 2010
$12.99 BUY APP

Color Splash
Photography
Updated Jul 26, 2010
$0.99 BUY APP

Cut the Rope
Games
Released Oct 05, 2010
Game Center
$0.99 BUY APP

Dark Nebula - Episode T...
Games
Released Aug 05, 2010
$1.99 BUY APP

Dragon Dictation
Business
Updated Nov 04, 2010
FREE APP

The Elements for iPhone 4
Books
Released Jun 23, 2010
$9.99 BUY APP

Epicurious Recipes & Shop...
Lifestyle
Updated Jul 06, 2010
FREE APP

ESPGALUDA II
Games
Updated Sep 08, 2010
$8.99 BUY APP

Evernote
Productivity
Updated Sep 25, 2010
FREE APP

Explore 9/11
Education
Updated Sep 10, 2010
FREE APP

Facebook
Social Networking
Updated Nov 03, 2010
FREE APP

Flight Control
Games
Updated Oct 07, 2010
Game Center
$0.99 BUY APP

Fotopedia Heritage
Travel
Updated Oct 26, 2010
FREE APP

Golfscape GPS Rangefi...
Sports
Updated Jul 22, 2010
$19.99 BUY APP

Guitar Hero
Games
Updated Sep 28, 2010
$2.99 BUY APP

GuitarToolkit
Music
Updated Nov 02, 2010
$9.99 BUY APP

Hipstamatic
Photography
Updated Nov 04, 2010
$1.99 BUY APP

HOMERUN BATTLE 3D
Games
Updated Oct 22, 2010
$4.99 BUY APP

IMDb Movies & TV
Entertainment
Updated Nov 04, 2010
FREE APP

IMAGE 1 The best apps in Apple App Store in 2010. Picture by Rene Ritchie, 2010.

On January 22, in 2011 the download count reached 10 billion downloads, as well as a total number of apps available for download rose to 350 000.

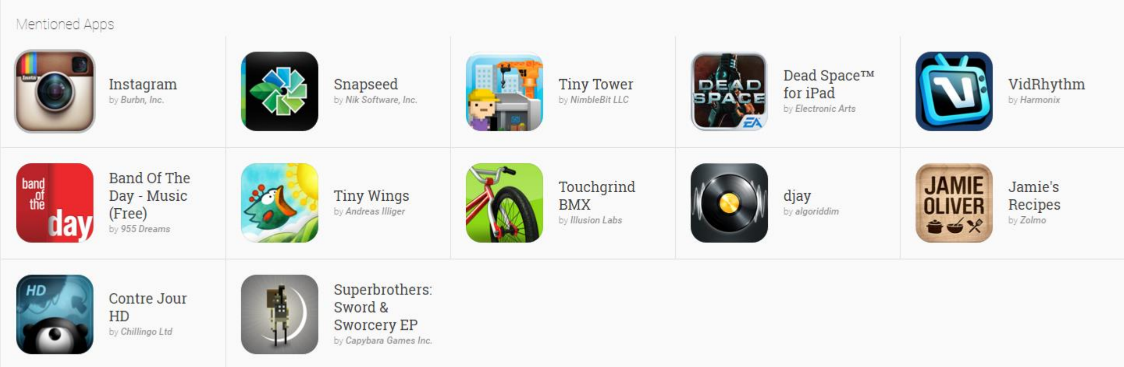
> “With more than 10 billion apps downloaded in just two and a half years—a staggering seven billion apps in the last year alone—the App Store has

> surpassed our wildest dreams. The App Store has revolutionized how software is created, distributed, discovered and sold. While others try to copy the App Store, it continues to offer developers and customers the most innovative experience on the planet.” (Schiller 2011)

On July 7, 2011, Apple announced that more than 15 billion apps were downloaded from Apple App Store with over 425 000 apps that were available on the market. Within three years Apple App Store became the most successful mobile software marketplace which was available in 90 countries worldwide. Furthermore, Apple had paid app developers more than $2.5 billion. (Pope 2011)

In October 2011, during the Apple Special event, Apple claimed to have over 500 000 apps available in Apple App Store and to be the number one market for mobile apps. They also reached 18 billion app downloads from Apple App Store. (Apple Special Event 2011) In December 2011, Apple listed the best apps of the 2011 year under the “App store Rewind 2011” name. (O’Grady 2011)

The award of the iPhone App of the Year went to Instagram, a photo-sharing social network with the variety of stunning photo filtering effects. The iPhone Game of the Year award went to Tiny Towers, a game where the main goal was to grow an economic empire by building towers and populating it with businesses and citizens. The iPad App of the Year was Snaapseed, an app that makes photography extraordinary and adds gorgeous effects to the photos. The iPad Game of the Year was Dead Space, mobile horror third-person shooter remake of earlier video game which succeeded in 2008 on another platform such as PC, Xbox, and PlayStation. Image 2 presents the best apps of 2011 year.



Mentioned Apps

Instagram by Burbn, Inc.	Snapseed by Nik Software, Inc.	Tiny Tower by NimbleBit LLC	Dead Space™ for iPad by Electronic Arts	VidRhythm by Harmonix
Band Of The Day - Music (Free) by 955 Dreams	Tiny Wings by Andreas Illiger	Touchgrind BMX by Illusion Labs	djay by algoriddim	Jamie's Recipes by Zolmo
Contre Jour HD by Chillingo Ltd	Superbrothers: Sword & Sworcery EP by Capybara Games Inc.			

IMAGE. 2 Best Apps of 2011 according to Apple. Picture by Bryan M. Wolfe, 2011.

On March 5 in 2012, Apple announced that they reached over 25 billion app downloads and more than 550 000 were apps available in 123 countries around the world. (Miller 2012) With such a growth rate and amount of apps, Apple App Store became the most competitive market for mobile software. However, oversaturated market had its issues.

According to Apptrace which is a free app analysis service to monitor apps performance in the mobile market. In August of 2012, over 650 000 apps were available in Apple App Store but it listed only 300 top ranked apps for 43 categories in the App Store. On August 2, 2012, only 307 396 apps were ranked, meaning that more than 50 percent of the apps were simply unavailable for the users. Apptrace decided to narrow the numbers of ranked apps taking off from the list the apps that could not hold a steady position for at least one week. This brought the results of only 256 959 apps holding the position from July 18th until July 25th. The remaining apps were off the ranks and was labelled as “dead or zombie apps”, existing outside of the user’s eye. (AppTrace Blog 2012)

At the end of 2012, Apple revealed the best of and most popular apps both free and paid (Image 3). The top paid app of 2012 was Angry Birds, as well as the top-selling app of the year. Apple chose Action Movie FX as the best iPhone app of the year, the app that allows users to add Hollywood visual effects to video recordings. Paper by FifryThree was named the best iPad App of the Year, the app allows users to create notes and sketches simulating paper. The best iPhone Game of the Year award was given to Rayman Jungle Run. Rayman Jungle Run is a mobile platformer-runner game based on Rayman Origins which is platform video game originally released in 2011 for PC and consoles. The best iPad Game of the Gear award was won by The Room – a puzzle game, where the main goal of the game is to escape the room.

More than that, according to Apple’s press release from January 2013, Temple Run - endless survival-runner game was downloaded more than 75 million times. Apple also mentioned such companies as Backflip Studios and Supercell who earned over $100 million combined for their successful titles: DragonVale and Clash of Clans. (Miller 2013)

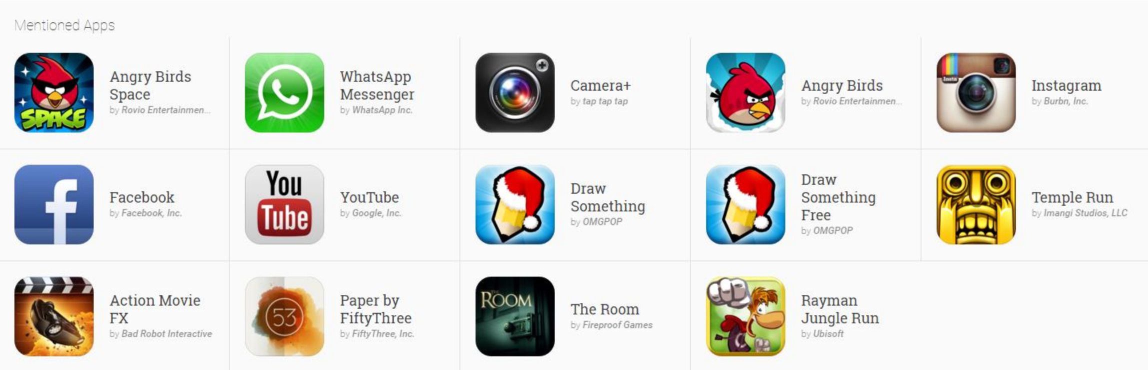
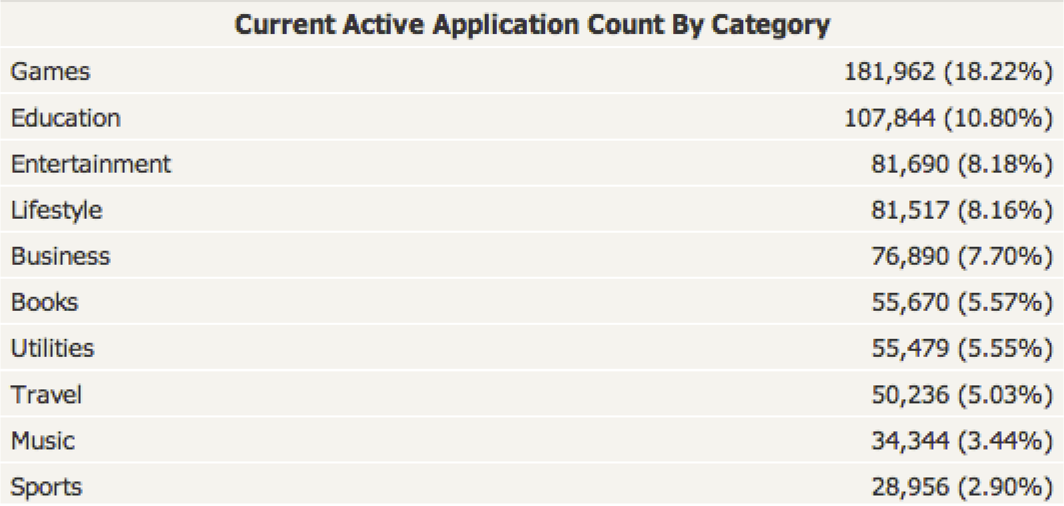


IMAGE 3. The best apps of the App Store in 2012. Picture by Brent Dirks, 2012.

On January 17 in 2013, Apple announced that they had reached 40 billion app downloads and 775 000 apps were available in Apple App Store. As well as, the fact Apple App Store covered 155 countries around the world. (Miller 2013)

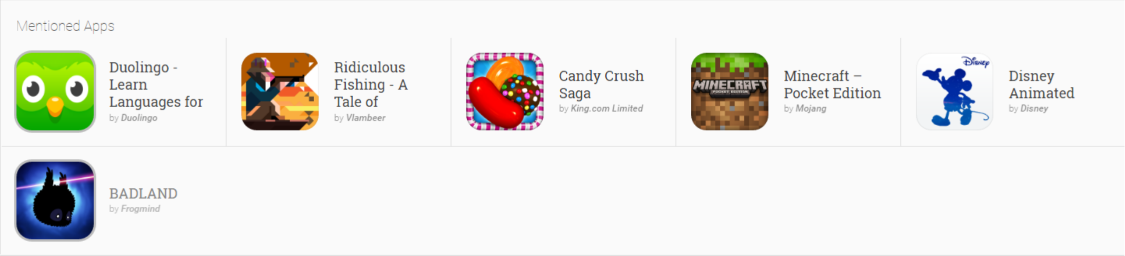
By the end of 2013, Apple customers had spent over $10 billion on Apple App Store and developers earned the total of $15 billion. (Miller 2014) More than that, Apple also reached over 1 million apps available in Apple App Store. According to 148App.biz data which were collected at the end of November, there were 997 673 active apps. However, over 370 000 apps were no longer available for download anymore. (Jones 2013) The largest app categories were Games, followed by Education and then Entertainment (Image 4). The picture below represents categories and their share of the market.



Current Active Application Count By Category	
Games	181,962 (18.22%)
Education	107,844 (10.80%)
Entertainment	81,690 (8.18%)
Lifestyle	81,517 (8.16%)
Business	76,890 (7.70%)
Books	55,670 (5.57%)
Utilities	55,479 (5.55%)
Travel	50,236 (5.03%)
Music	34,344 (3.44%)
Sports	28,956 (2.90%)

IMAGE 4. Application count by categories presented in the Forbes article about future of the Apple App Store. Picture by Chuck Jones, 2013.

Moreover, at the end of 2013, Apple again announced the best apps of the year (Image 5). Apple’s two most popular free apps were Candy Crush Saga and Minecraft – Pocket Edition. Candy Crush Saga is a match-three puzzle and Minecraft – Pocket Edition is a mobile adaptation of PC game. However, the award of iPhone App of the Year went to Duolingo, which is online language trainer. The iPhone App of the Year was Ridiculous Fishing, a stylized humorous finishing game. As for the iPad the App of the Year, it was Disney Animated, a paid app which shows the process behind animation creation of Disney Studio. The Game of the Year award was given to BADLAND, an action game where the main objective is to travel through a level and survive with the character.



Mentioned Apps

Duolingo - Learn Languages for
by Duolingo

Ridiculous Fishing - A Tale of
by Vlambeer

Candy Crush Saga
by King.com Limited

Minecraft – Pocket Edition
by Mojang

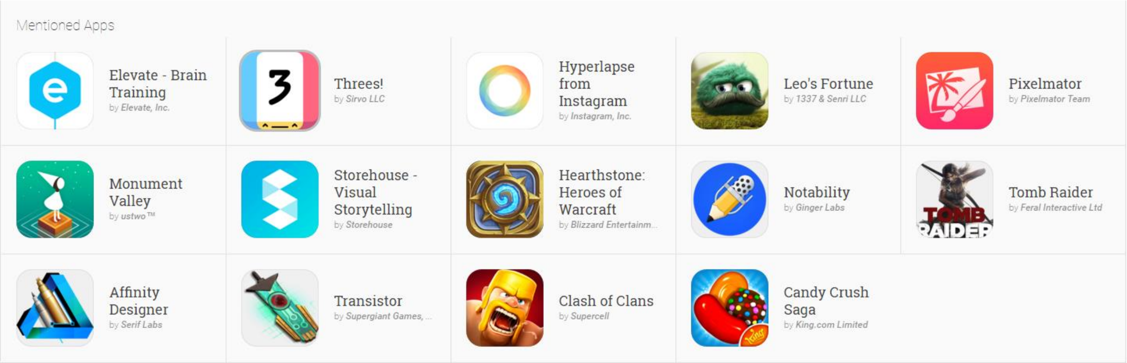
Disney Animated
by Disney

BADLAND
by Frogmind

IMAGE 5. The best Apple App Store apps of 2013. Picture by Joe White, 2013.

During Apple’s Worldwide Developer’s conference in June of 2014, the number of 75 billion app downloads was announced to the public. (Perez 2014) Despite the success of App Store, the development of apps became much harder due to the immense saturation of the app market. Adjust released terrifying numbers of “Zombie Apps”. Out of 1 197 087 apps, 957 387 apps were “Zombie Apps”, which was almost 80% of total apps available in Apple App Store. Thus, the discovery of an app became the primary issue for the developers. In June of 2014, according to Adjust, there were only 1 197 087 apps available out of 1 601 413 total of apps that had been uploaded to the App Store. (Adjust 2014)

In December 2014, Apple chose the best apps and presented them as “Best of 2014”. The best iPhone App of the Year was Elevate - Brain Training, an app that allows a user to train his/her brain with a variety of small games which require cognitive skills, memory, processing, math, precision, and comprehension. The Game of the Year was Thees!, a small puzzle game. Apple also mentioned the Runner-Up apps, which were Hyperlapse, an app for shooting time-lapses, and Leo’s Fortune, which is platformer adventure game. As for the best iPad App of the Year, image-editing app Pixelmator took that award and the best iPad Game of the Year was Monument Valley, a surreal exploration puzzle game. Moreover, iPad Best Apps also included Runner-Ups, which were Storehouse, an app that allows users to create a story by combining their photos, and Hearthstone: Heroes of Warcraft, competitive online card game. More than that Apple also presented all the top apps of the year. The image 6 shows the best apps of 2014 year.



Mentioned Apps

Elevate - Brain Training
by Elevate, Inc.

Threes!
by Sirvo LLC

Hyperlapse from Instagram
by Instagram, Inc.

Leo's Fortune
by 1337 & Senri LLC

Pixelmator
by Pixelmator Team

Monument Valley
by ustwo™

Storehouse - Visual Storytelling
by Storehouse

Hearthstone: Heroes of Warcraft
by Blizzard Entertainm...

Notability
by Ginger Labs

Tomb Raider
by Feral Interactive Ltd

Affinity Designer
by Serif Labs

Transistor
by Supergiant Games, ...

Clash of Clans
by Supercell

Candy Crush Saga
by King.com Limited

IMAGE 6. The Best Apple App Store apps of 2014 year. Aldrin Calimlim, 2014.

By June 2015 the number of apps that were available in Apple App Store reached 1.5 billion and the total number of app downloads reached the 100 billion mark. Moreover, for the first time, the number of submitted apps for release went over 54 000. However, the statistic showed that a lot of downloaded apps were used only once in half a year. The most popular app category was Games. It occupied almost a quarter of the whole application market (Image 7).

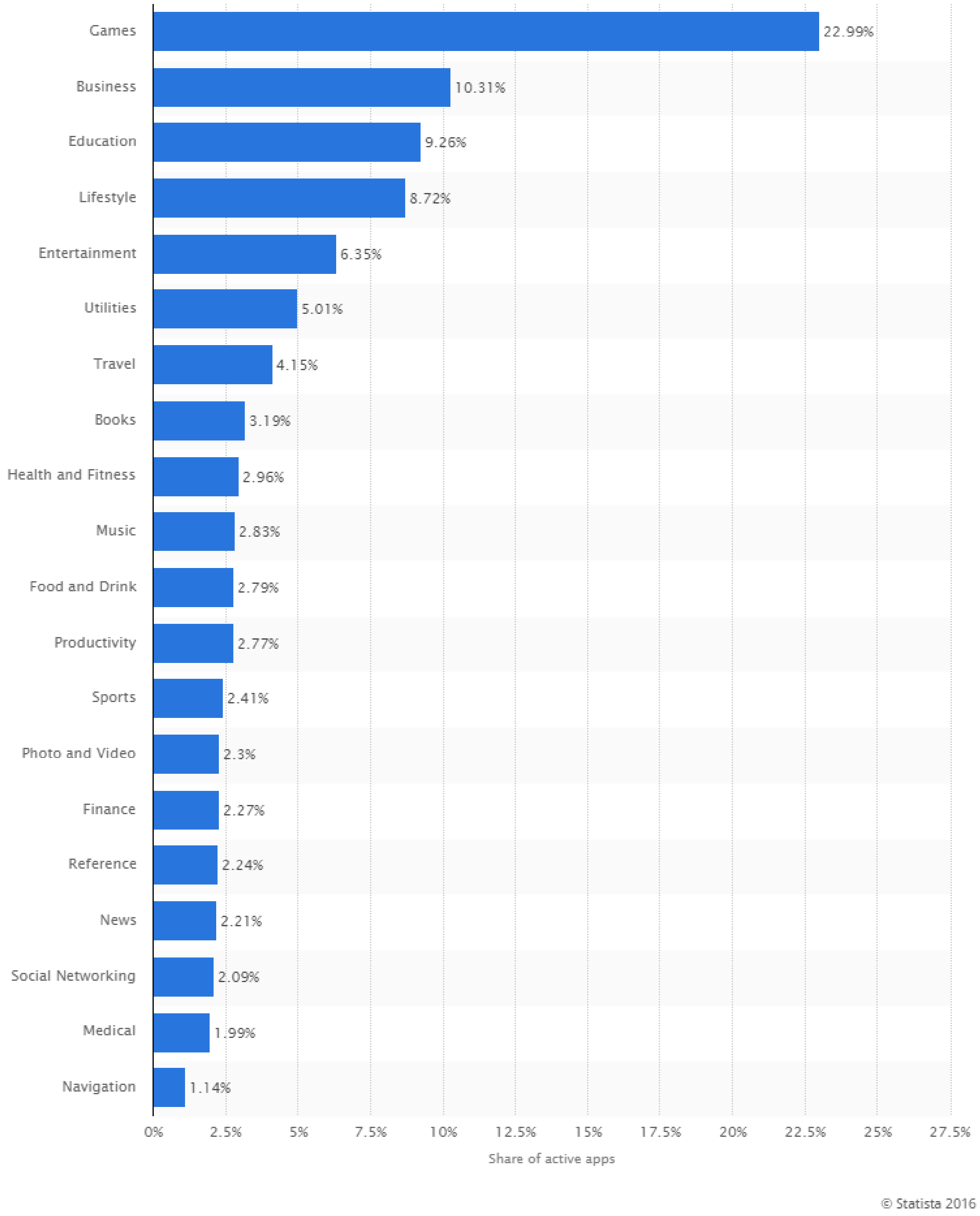
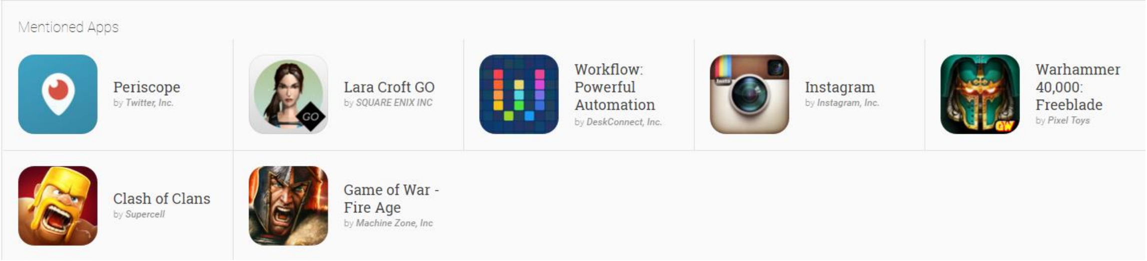


IMAGE 7. Apple App Store application categories and their total share by the active apps by Statista. (Statista 2016)

By the end of 2015 the best apps were chosen by Apple (Image 8). The App of the Year award went to Periscope. The app that was developed by Twitter, it allows users to stream video in real-time. The award Game of the Year was given to Lara Croft GO, a mobile adaptation of Lara Croft game franchise. As for the Best App award, it went to Instagram, a photo-sharing social network. While Warhammer 40,000: Freeblade received the award of the best game, an action combat game based on Warhammer universe. The top grossing apps were Clash of Clans and Game of War – Fire Age, which are both mobile strategy games. (White 2015)



Mentioned Apps

Periscope
by Twitter, Inc.

Lara Croft GO
by SQUARE ENIX INC

Workflow: Powerful Automation
by DeskConnect, Inc.

Instagram
by Instagram, Inc.

Warhammer 40,000: Freeblade
by Pixel Toys

Clash of Clans
by Supercell

Game of War - Fire Age
by Machine Zone, Inc

IMAGE 8. The best Apple App Store apps of 2015. Joe White, 2015.

As for 2016, in January Apple announced that almost $40 billion had been paid to app developers. (Israel 2016) However, there had not been any official numbers on apps or app downloads by now. It is already seen what apps that were published in 2016 and they are dominating the market. The examples of those apps are Clash Royale and Angry Birds 2.

3.2 Conclusion

From this research, it became very clear that the future of the Apple App Store will be mainly focused on mobile games. Games are the most popular and profitable category. They occupy the biggest share of the market and have the best user retention. Games category attracts most of the mobile users and they tend to spend most of the time in games. This makes the mobile game industry a very prosperous and yet competitive environment for game developers.

4 ANIMATION

4.1 Introduction

Working in the game development industry often requires a big variety of technical and artistic skills, extreme problem-solving abilities, as well as creativity. These are the most important qualities that a game designer should have.

Character creation and its animation are considered as the hardest tasks in game development. Creating a character from zero and then developing it further is an extremely heavy and prolonged task.

In order to compare animation methods, the appropriate studies should be made of the animation principles that are commonly used in the game development industry but before that it is necessary to know the methods of creating animation and the animation fundamentals. More than that, the animation methods would be implemented practically for the evaluation and comparison. It would be done through the process of character creation and animating.

4.2 The principles of animation

In order to understand how animation methods could actually affect mobile game development, it is necessary to study the basic principles of animation that were established when the first animations were made.

The oldest principles of animation were developed by Frank Thomas and Ollie Johnston during the 1930s. The main idea behind the principles they created was to produce and the illusion of basic laws of physics happening on the screen. These principles are still used by most of the animators in game, cartoon and film industries and they are considered as a “Bible of Animation”. (Allan 2002) There is total of twelve main principles of animation.

4.2.1 Squash and stretch

The squash and stretch principle is considered the most important principle, it defines the mass and hardness of an object by changing its shape during the movement. In real life, the squash and stretch are not so noticeable by the human eye but the animation allows to exaggerate squash and stretch and fake it. As a result, the human eye perceives animation as something natural. The core rule of squash and stretch is that the deformation of the object should never affect the volume of that object. (Thomas and Johnson 1981, 47 – 51) The example of squash and stretch (Image 9) would be the bouncing ball that is distorted along its movement.

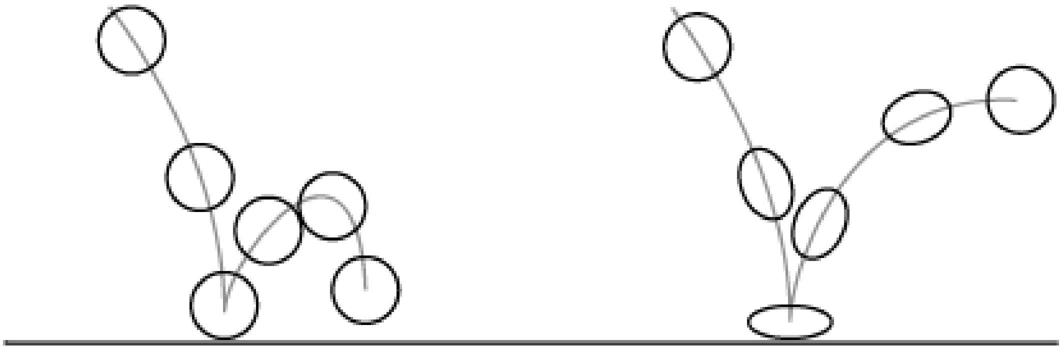


IMAGE 9. The squash and stretch principle, graphic representation. Picture by Ralph A. De Stefano, 1999.

4.2.2 Anticipation

The animation itself consists of three phases: the preparation, the action and the termination or end of the action. In animation, the anticipation is used to prepare the viewer for the action that is going to happen. The viewer may not understand the animation until there is a planned sequence of events that will lead him/her from one action to another. (Thomas and Johnson 1981, 51 – 53)

For instance, the preparation before throwing the ball is an excellent example of anticipation (Image 10). It prepares the viewer for the future action that is going to happen.

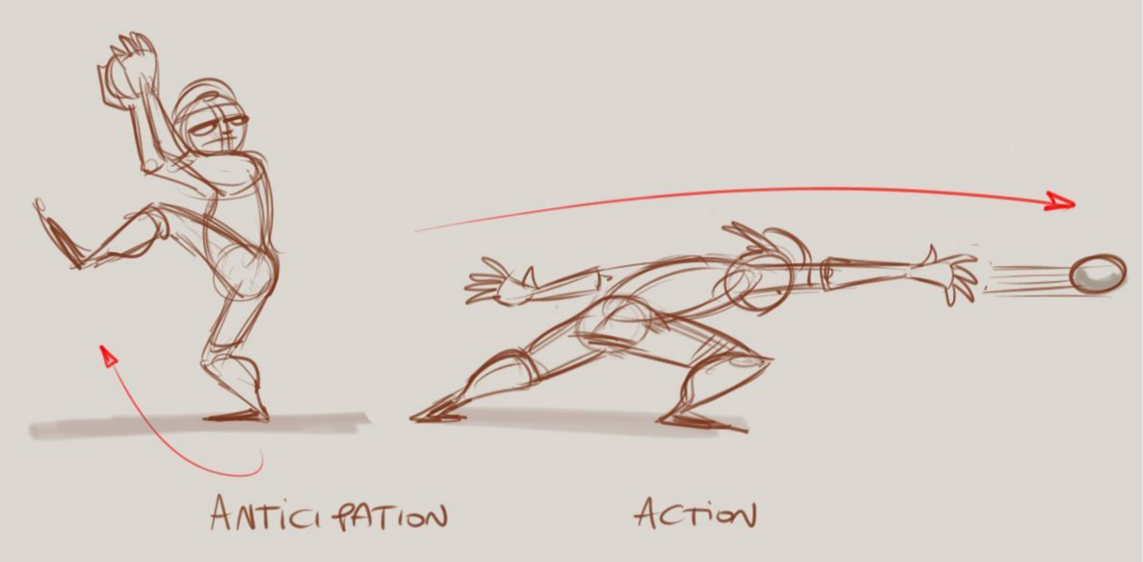


IMAGE 10. Anticipation principle example. Picture by Bash studio, 2015

Moreover, anticipation can indicate the speed of the action if the viewer is not prepared for the fast action he/she may completely miss it but the anticipation work in a way that allows a viewer to understand that the next action would be done in a fast manner. Another role of anticipation is based upon concentrating the viewer attention to a certain direction. A character looking at any direction and providing reaction shows the viewer that there is something important happening in that direction. (De Stefano 1999)

4.2.3 Staging

The staging principle is devoted to the scene or the place where the action is about to happen. It works in a way that the correct placement of the scene elements allows the viewer to see the animation in the clearest and understandable way. Usually, it is achieved by the correct placement of the characters, background and foreground elements. More than that the camera setup is the most crucial element in staging principle because it is responsible for the communication of the scene to the audience. (Masters 2014) The Image 11 represents the example of staging principle with the action scene.

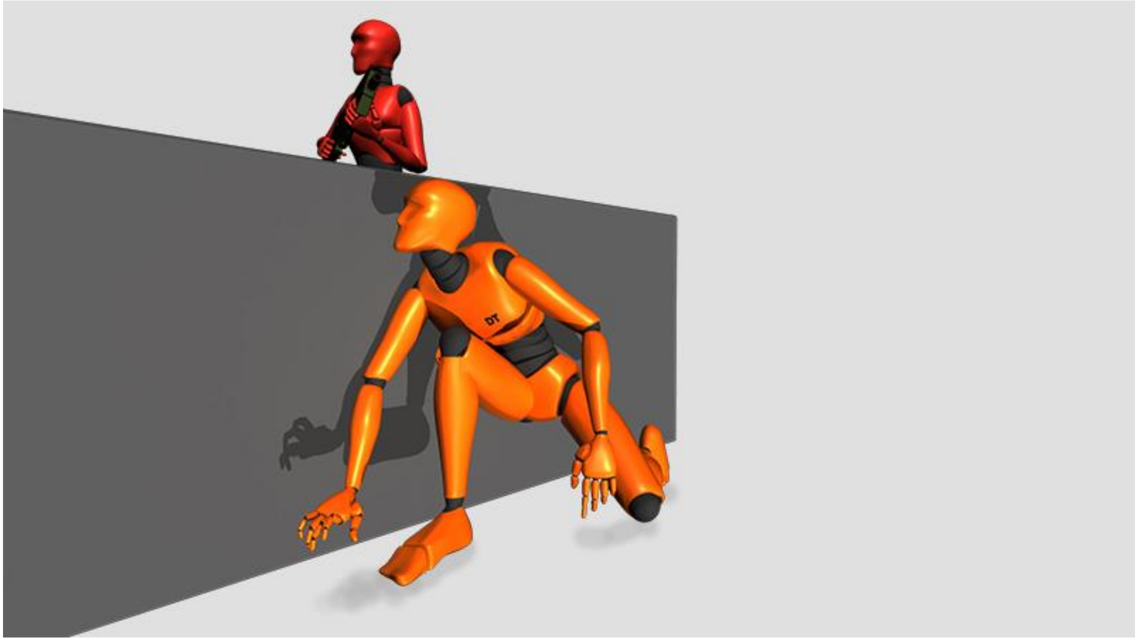


IMAGE 11. Example of staging principle. Picture by Mark Masters, 2014.

4.2.4 Straight Ahead Action and Pose to Pose

Straight ahead action and pose to pose are the two different methods how animation could be created. Straight ahead action is a method when the animation is created in a fast and spontaneous way. This is usually done by producing a lot of hand drawing sketches of animation. While pose-to-pose is a planned method where an animator creates the detailed plan of how to implement the animation by capturing the key elements of it. Pose to pose method is the method that is used in computer animation a lot because computer animation requires a lot of preparations and creation of hierarchy that is associated with the transformation of an object. (Masters 2014) The Image 12 represents an example of this principle based on character animation.

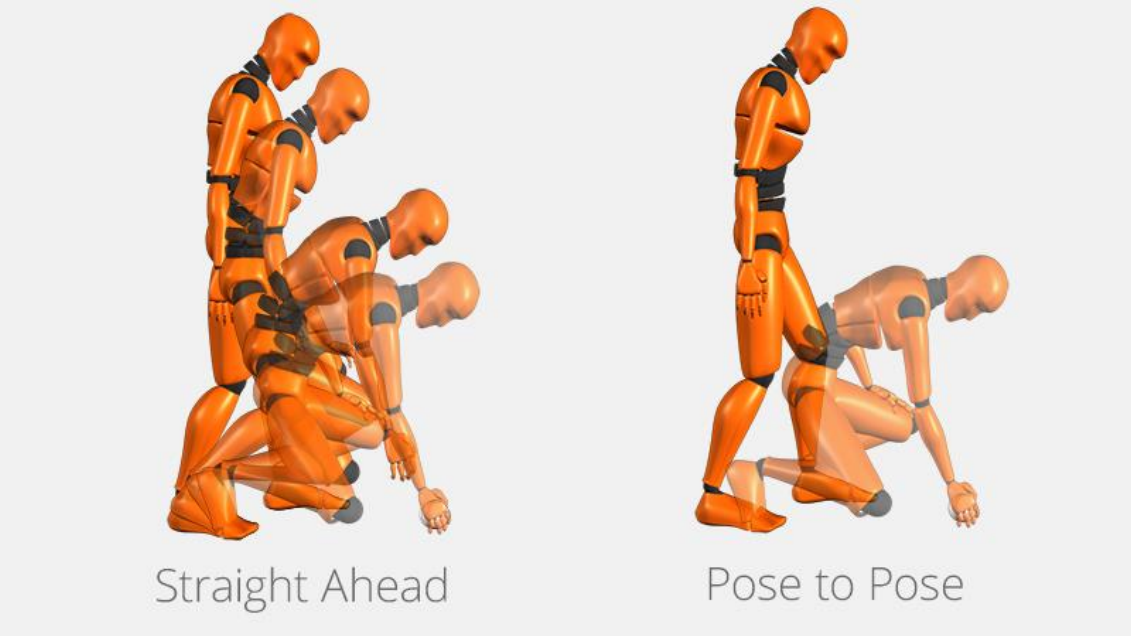


IMAGE 12. Example of Straight Ahead Action and Pose to Pose principle. Picture by Mark Masters, 2014.

4.2.5 Follow Through and Overlapping Action

Follow through and overlapping action could be divided into different principles, but they are closely related. Follow through dictates that the separate parts of an object will continue their movement even if the object has stopped. For instance, a simple walk animation represents this principle when the character stops its movement, the arms still continue to swing for a small amount of time. Overlapping action main principle is that different parts of the body move at different times. As an example, if a character raises a hand, the shoulder would be the first to move and then the rest of the arm. (Masters 2014)

4.2.6 Slow in and slow out

The slow in and slow out is a principle that defines that an object needs time for the movement to gain the speed, as well as the time to return to the rest. Any object's movement always starts from dormancy state and later the speed is increased. (Thomas and Johnson 1981, 61) For instance, the car cannot reach a 60 km/h speed instantly, as well as it cannot stop instantly having same speed.

4.2.7 Arcs

The arcs principle states that most of the actions follow an arc path except the mechanical machine actions. In nature, arcs are the most efficient and economical routes that an object uses to get from one position to another. Arcs give animation a great flow and feel of natural movement. (Thomas and Johnson 1981, 62 – 63)

4.2.8 Secondary Action

The secondary action is a principle the task of which is to enhance the main action and add more depth to the animation by complementing and/or strengthening the main action. The important aspect of the secondary action is to have control over it, otherwise, it could interfere with the main action. (Thomas and Johnson 1981, 63 – 64) As an example, two characters talking would be a primary action while the second character scratching its back would be the secondary action.

4.2.9 Timing

Timing is what creates the illusion of a character or an object moving. Timing controls the number of frames that animation contains and controls the movement speed of an object. (Thomas and Johnson 1981, 64 – 65)

4.2.10 Exaggeration

The exaggeration principle is used to develop the animation further and add more details to the action. Animations tend to have less realism and details than a real life movement, so in certain situations, it is very challenging to communicate the certain element for the viewer. (Thomas and Johnson 1981, 65 – 66) For example, a facial expression is hard to convey and they are exaggerated most of the time.

4.2.11 Solid Drawing

The solid drawing principle is about finding the correct balance, weight, volume, and pose. It is done to bring life to any animated object. With correct solid drawing, the animated object will feel realistic when it is viewed as an illusion. (Thomas and Johnson 1981, 66 – 68)

4.2.12 Appeal

The appeal principle allows creating a connection between the character and the viewer. The viewer always notices great character design, simplicity, and charm. However, the bad design could turn the viewer away from the character. (Thomas and Johnson 1981, 68 – 69)

4.3 Conclusion

These principles have proven themselves worthy many times. They are an integral part of the animation process. All animations that are being developed nowadays are a result of understanding the main principles of animation and their correct implementation.

4.4 The ways of creating animation

There are five different methods to create animations that are being used by animators nowadays. Each of them has a different method of implementation but they all require an animator to know the principles of animation. Moreover, these ways usually could be combined together to achieve the best quality, performance, and style of an animation.

4.4.1 Traditional animation

The first way of creating an animation is the traditional way. This method is referred to the hand drawing method where the animator creates an animation by drawing the object movement for every frame that the animation uses. Nowadays, all traditional animators use graphic tablets and advanced graphic software to create animation in a traditional way which greatly increases the productivity in comparison to traditional media such as pen and paper. (Meroz 2015)

4.4.2 2D vector based animation

2D vector based animation is a method of animation which allows the animator to use software which can interpolate the movement of the object and fill in the frames between the key poses of the object. More than that, some 2D animation software provides the features such as rigging and moving objects without repetitive redrawing of the object. (Meroz 2015)

4.4.3 3D animation or computer animation

The most prevalent way of creating an animation is 3D animation or computer animation. In spite of the fact that 3D animation still incorporates all animation principles that the 2D animation uses, it is a drastically different way of animation creation. The technical aspect plays a big role because the software offers an enormous variety of tools and requires an animator to have expertise in 3D software. 3D animation happens in the way that an animator interacts with a 3D model inside the 3D software using the controls that manipulate the model movement. After the 3D model is moved an animator saves the transformation to the key frames. Then the software calculates the path of the movement between key frames thus creating the animation. Another important aspect of 3D animation is that the character, for example, has to always maintain movement otherwise it will look unnatural and this mistake could be easily spotted by the viewer. (Meroz 2015)

4.4.4 Motion graphics

The motion graphics method is referred to the creation of visual effects and graphic elements that support the animation scene. More than that, mostly motion graphic is used for commercial and promotional materials. (Meroz 2015)

4.4.5 Stop motion

The last way of creating animation is stop-motion. Stop-motion is a simple process of creating an animation by taking multiple photos of an object while it is intentionally moved. When all the photos are combined together the illusion of movement is created. (Meroz 2015)

4.5 The process of developing a character, from concept to animation

4.5.1 Character concept developing

Depending on the project a character is bound to a certain visual style and a theme of the project itself. Concept art is a representation of the character's outlook by the designer. The designer creates a series of concepts, developing different ideas of how the character will look like. When the concept is finalized it should be presented and then approved for further development.

Creating concept art starts with a small series of sketches. The more sketches the designer creates the more potential ideas can be brought to the client. After finishing the series of sketches, the designer decides which of the sketches will be taken forward for creative development. The primary goal of choosing the right sketches is to design them so they become more readable, making the sense of the sketch. The main idea is not to create as much details as possible but to define the design of the character bringing the details such as hair, clothing and any items the character is possessing. After that comes the time of creating design variations. Experimentation is a great tool of achieving an outstanding design of a character. Designing the character equals to the creation of as much ideas as possible and afterwards choosing the best ones. (Bowate 2003)

4.5.2 Animation process

After the character is approved, the final piece either 2D or 3D is taken further for the animation development. The character animation itself is a long and difficult process. The animation process always depends on the type of the project but the same methods could be applied for both 2D and 3D animations.

4.5.2.1 3D animation

If the project has a 3D character, the first step would be for the modeller to create a 3D mesh which is going to be the model of the character concept. When the mesh is finalized it would be rigged first in order to create an animation. After that, the animation is created by either the artist who created the model, animator or animation team.

A character rigging refers to the creation of a digital skeletal structure which is attached to the character model and which operates as the controller for the character movement. A character rig could vary depending on the complexity of the character. Placing a digital skeleton is a simple task, but the basic knowledge of human anatomy is required. While rigging the animator places joints to create a bone hierarchy so the digital skeleton will move correctly (Image 13). Rigging can be interpreted as a bone placement for the 3D model. There are parent bones and child bones. The parent bone is always located at the beginning of bone hierarchy. The parent-child relationship in rigging works the way that if the parent bone is moved the child bone always follows the parent’s movement but if the child bone is moved the parent remains static.

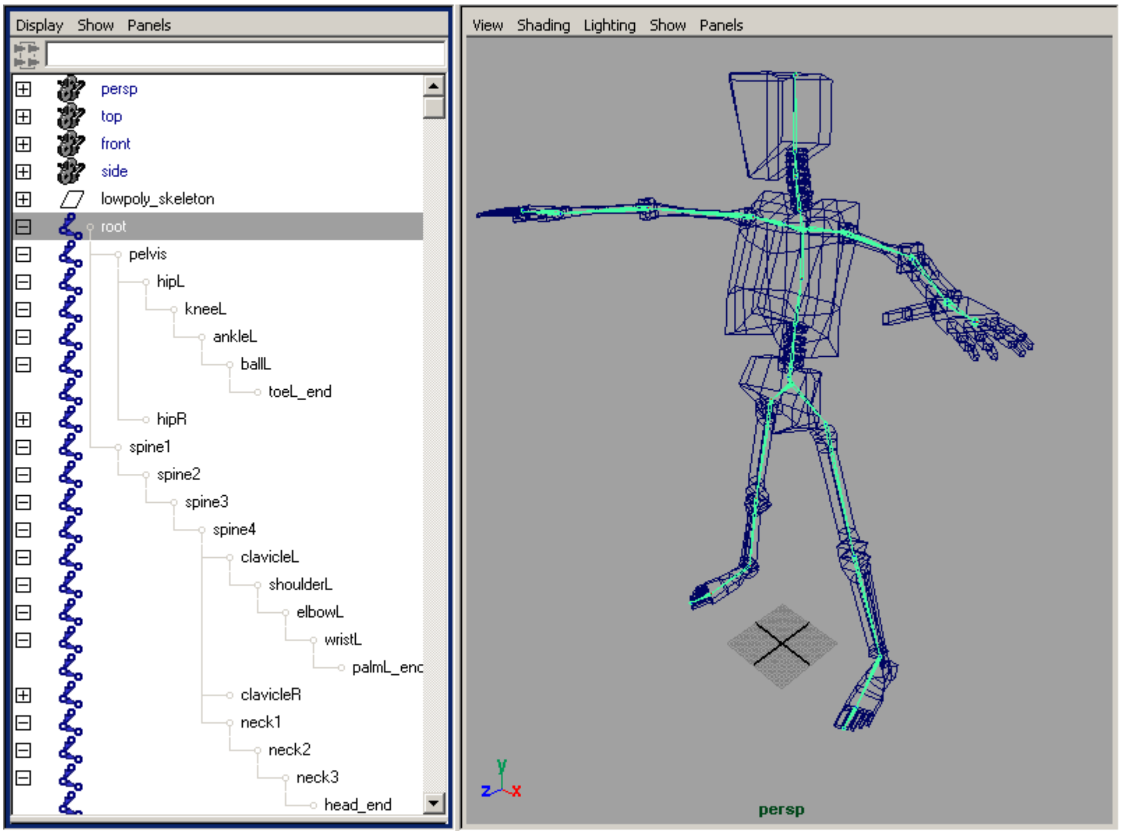


IMAGE 13 shows the completed character rig and the hierarchy of bones. (Basic Character Rigging)

However, the character rig could take a vast amount of time in the development process. Advanced rigs (Image 14, 15) usually include different controllers and constraint elements that help the rig to achieve natural movement. The following rig and rig controllers were created for real 3D project and the estimated amount of time that was spent for the rigging process was six to eight hours.

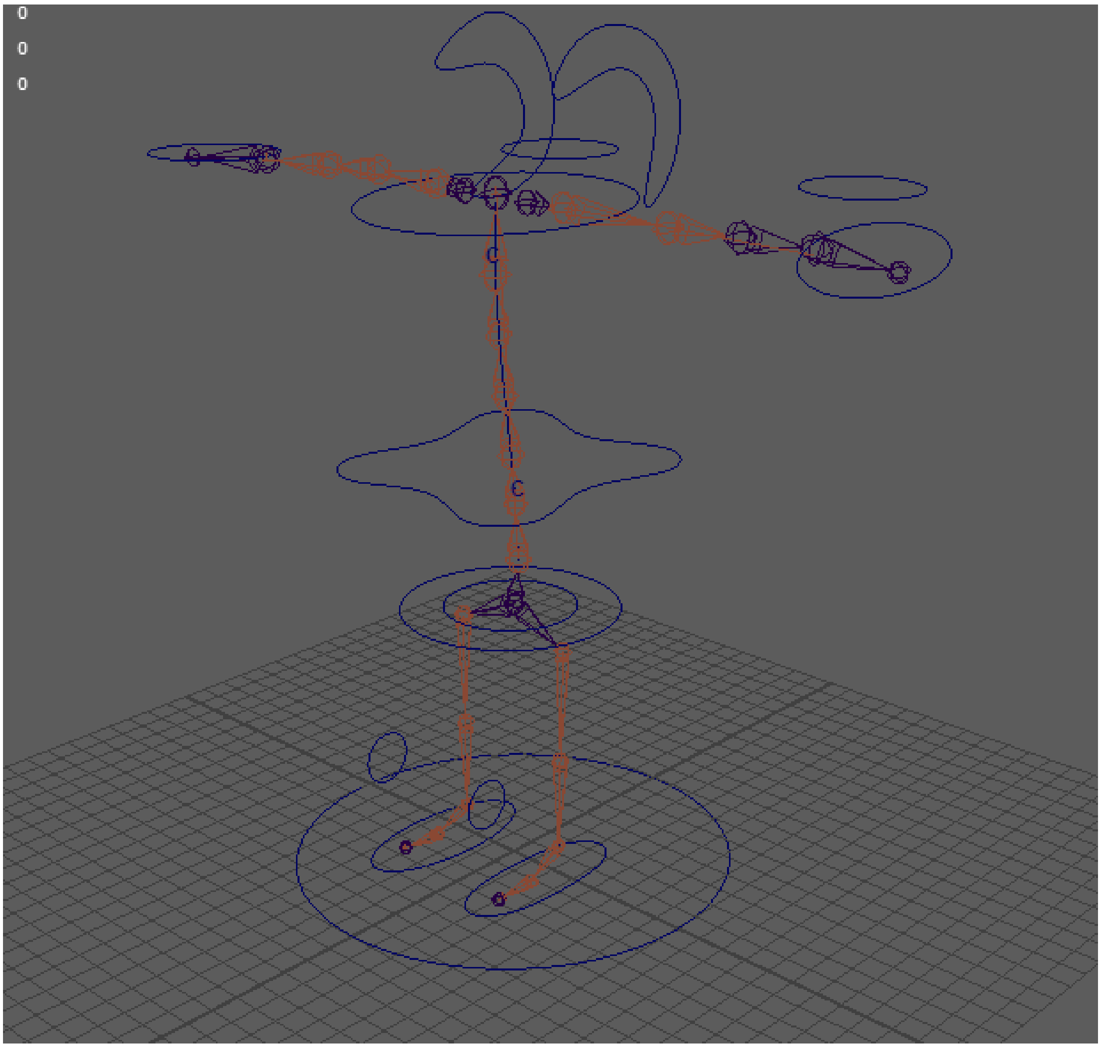
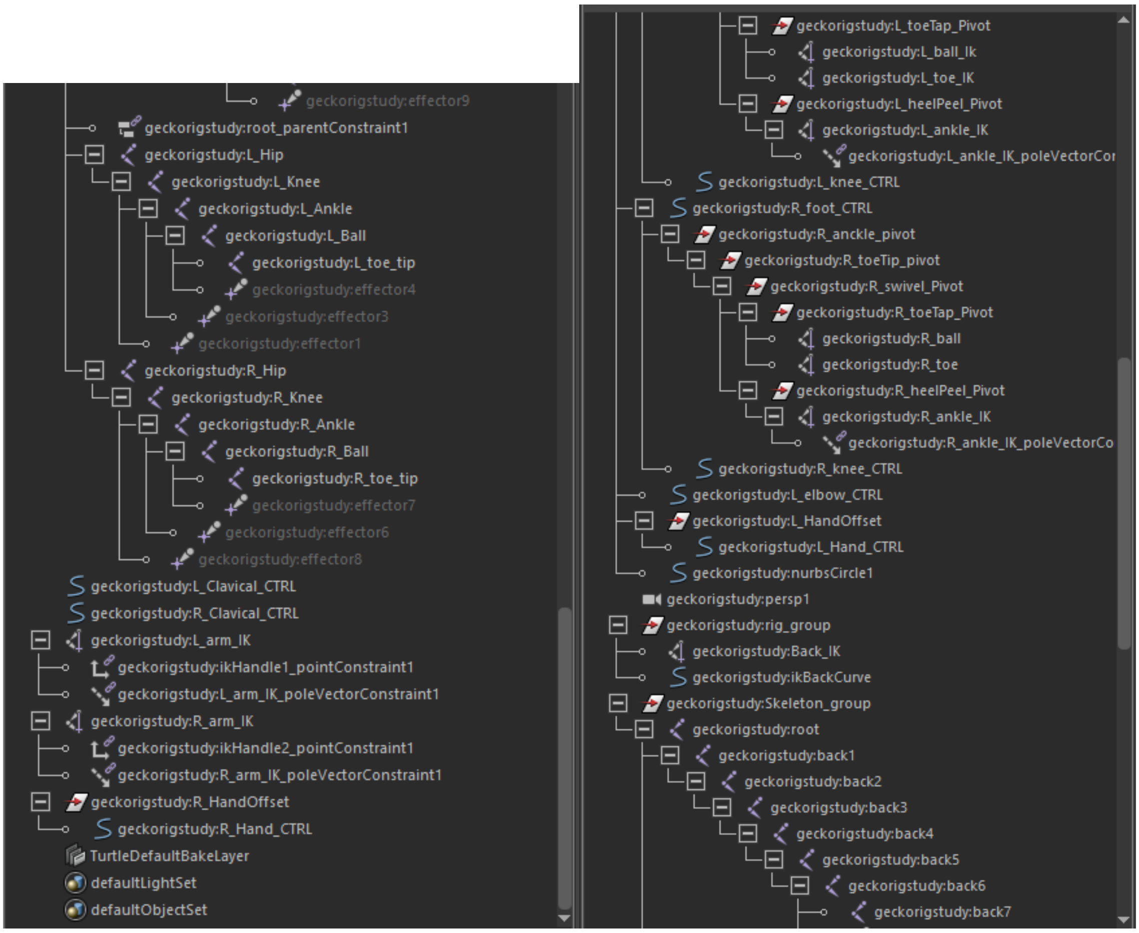


IMAGE 14. Shows the rig with controllers (blue circles and ellipses) that respond for different parts of the rig. Rig by Stanislav Belov, 2016



```
            geckorigstudy:effector9
    geckorigstudy:root_parentConstraint1
    geckorigstudy:L_Hip
      geckorigstudy:L_Knee
        geckorigstudy:L_Ankle
          geckorigstudy:L_Ball
            geckorigstudy:L_toe_tip
            geckorigstudy:effector4
          geckorigstudy:effector3
        geckorigstudy:effector1
    geckorigstudy:R_Hip
      geckorigstudy:R_Knee
        geckorigstudy:R_Ankle
          geckorigstudy:R_Ball
            geckorigstudy:R_toe_tip
            geckorigstudy:effector7
          geckorigstudy:effector6
        geckorigstudy:effector8
  geckorigstudy:L_Clavical_CTRL
  geckorigstudy:R_Clavical_CTRL
  geckorigstudy:L_arm_IK
    geckorigstudy:ikHandle1_pointConstraint1
    geckorigstudy:L_arm_IK_poleVectorConstraint1
  geckorigstudy:R_arm_IK
    geckorigstudy:ikHandle2_pointConstraint1
    geckorigstudy:R_arm_IK_poleVectorConstraint1
  geckorigstudy:R_HandOffset
    geckorigstudy:R_Hand_CTRL
  TurtleDefaultBakeLayer
  defaultLightSet
  defaultObjectSet
```

```
          geckorigstudy:L_toeTap_Pivot
            geckorigstudy:L_ball_Ik
            geckorigstudy:L_toe_IK
          geckorigstudy:L_heelPeel_Pivot
            geckorigstudy:L_ankle_IK
              geckorigstudy:L_ankle_IK_poleVectorCor
      geckorigstudy:L_knee_CTRL
    geckorigstudy:R_foot_CTRL
      geckorigstudy:R_anckle_pivot
        geckorigstudy:R_toeTip_pivot
          geckorigstudy:R_swivel_Pivot
            geckorigstudy:R_toeTap_Pivot
              geckorigstudy:R_ball
              geckorigstudy:R_toe
            geckorigstudy:R_heelPeel_Pivot
              geckorigstudy:R_ankle_IK
                geckorigstudy:R_ankle_IK_poleVectorCo
      geckorigstudy:R_knee_CTRL
    geckorigstudy:L_elbow_CTRL
    geckorigstudy:L_HandOffset
      geckorigstudy:L_Hand_CTRL
    geckorigstudy:nurbsCircle1
  geckorigstudy:persp1
  geckorigstudy:rig_group
    geckorigstudy:Back_IK
    geckorigstudy:ikBackCurve
  geckorigstudy:Skeleton_group
    geckorigstudy:root
      geckorigstudy:back1
        geckorigstudy:back2
          geckorigstudy:back3
            geckorigstudy:back4
              geckorigstudy:back5
                geckorigstudy:back6
                  geckorigstudy:back7
```

IMAGE 15. Shows the bone structure or hierarchy of the rig which is represented in IMAGE 14. Rig by Stanislav Belov, 2016

The animation methods that is used in 3D is called key-framing or key frame animation. It was a traditional method involving hand drawing where the animator drew the frames with key poses of animation at different time intervals. Later the in-between frames were filled in with more drawings to complete the whole animation. From that time, everything changed because now the software calculates the frames in-between to create a smooth transition between key frames. However, in order to achieve the best results and the plausible movement, it is required to understand the animation principles.

A small example was made showing the key-framing method using Autodesk Maya LT – 3D computer graphic software. A simple animation of bouncing ball was created representing the key-framing method. Total of 24 frames was used to create the animation capturing the key frames every 6 frames (Image 16).

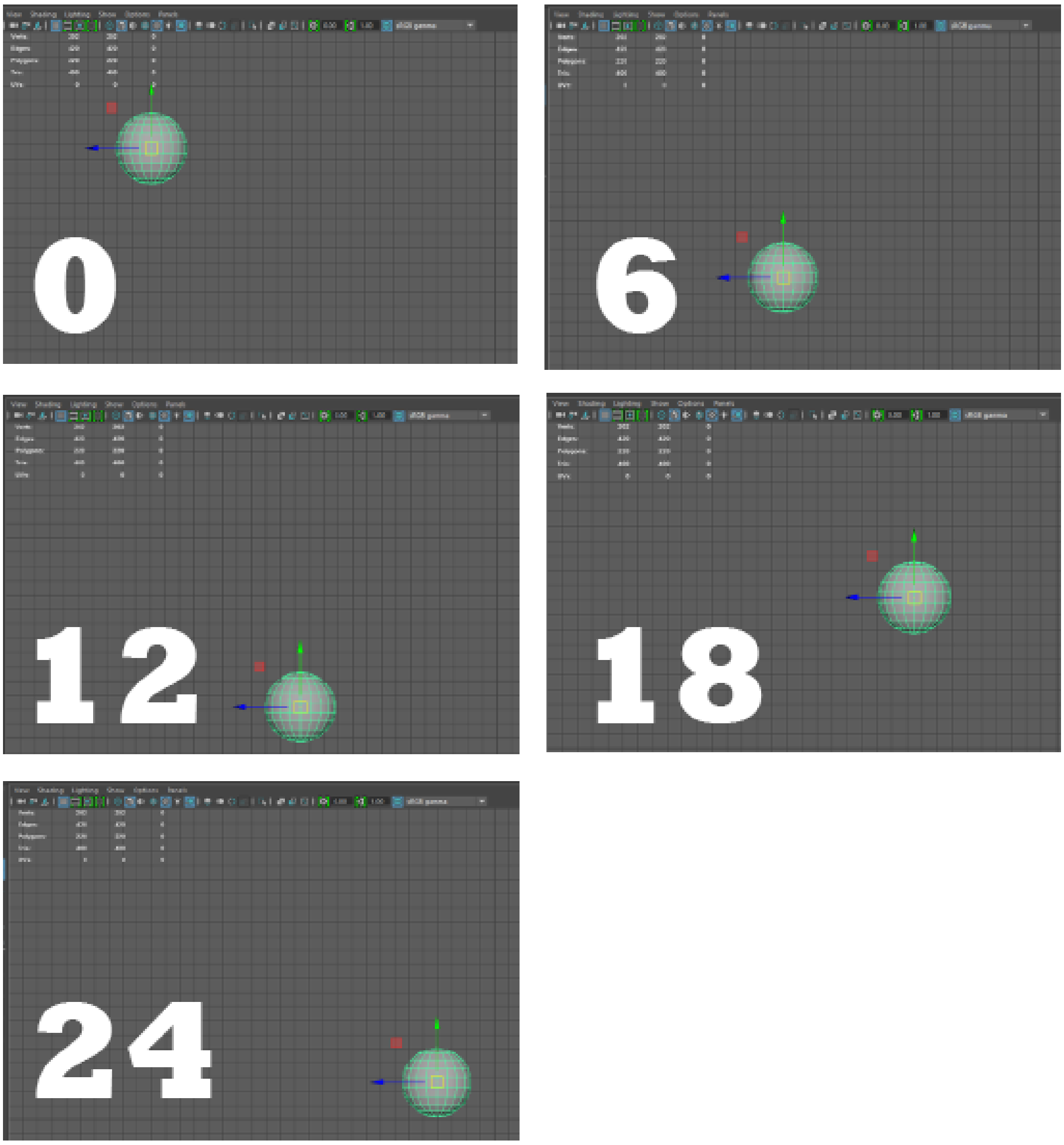


IMAGE 16. Represents the example animation implementing key framing method.

4.5.2.2 2D animation

If the project is 2D, the animator will use the same key-framing principle to create animation, either drawing frames in-between by hand (Image 17, 18) or using 2D software that allows the rigging of 2D objects and calculating in-between frames. The next example will show the key-framing method based on pixel animation. The example will present an 8 frame walk cycle. Where each frame was drawn by hand.

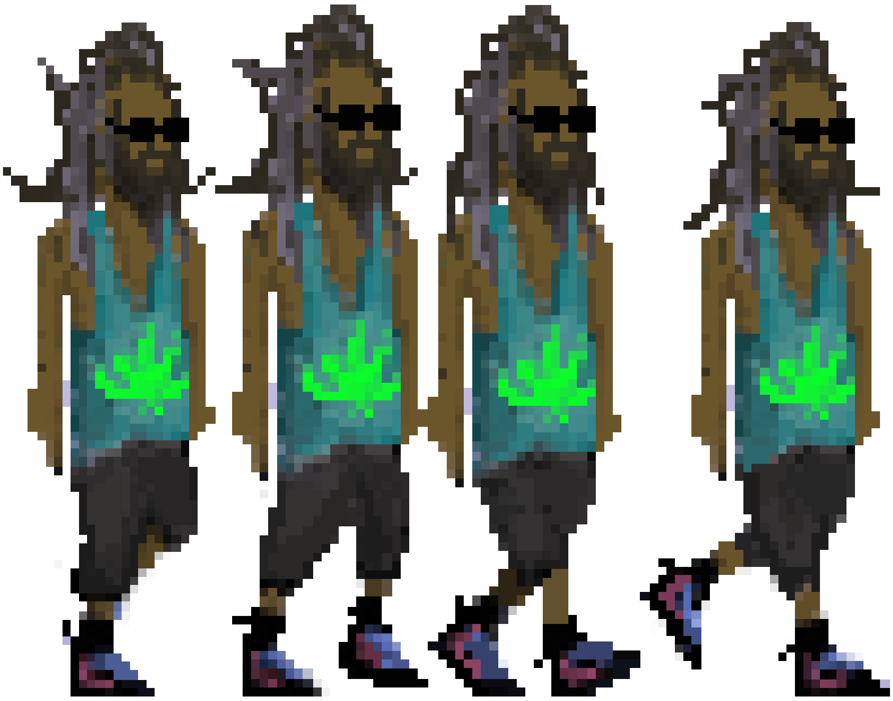


IMAGE 17. First four frames of the animation. Example by Stanislav Belov, 2016

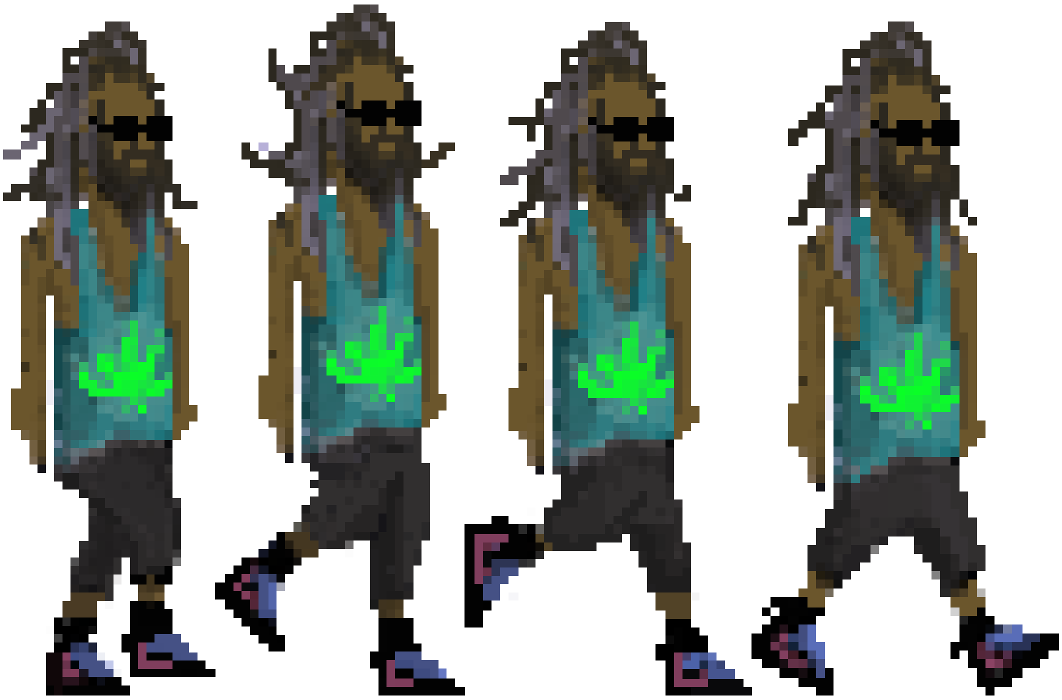


IMAGE 18. Lasts four frames of the animation. Example by Stanislav Belov, 2016

The 3D animation method showed that the process is very technical dependent and time-consuming because the rigging is an important part of animation and thus could not be skipped. Otherwise, the animation without rigging is impossible within 3D software. In addition to that, the animation process also consumes an enormous amount of time.

The 2D animation method which incorporates drawing in-between frames also proved itself to be very resource consuming. The actual drawing skills and understanding of how the graphics change through the frames is required for such type of animation creation.

4.6 2D character creation pipeline

4.6.1 Software

In this research, the main objective was to study the 2D method of animation using 2D animation software that incorporates rigging and key-framing methods. Furthermore, the research will also cover the programming of the logic behind the animation transitions and prototype the animation logic.

For this project, the character was created in Adobe Photoshop. Adobe Photoshop is a 2D graphic manipulation software. Photoshop is considered a leading photo editing software. Photoshop offers an enormous amount of graphic editing and creation tools, a variety of photo filters, color manipulations, and image transformations. However, Adobe Photoshop has a subscription based licensing and not free. The reason why it was used is the extensive knowledge about Adobe Photoshop and experience that was obtained within 7 years of usage.

As for the 2D animation software the free version of Spriter was used. Spriter is 2D vector animation software which works in the 2D environment featuring a 2D bone based animation system of the character and a modular method of animation, where the object is split into parts for simplified animating.

More than that, Unity Game Engine Kit was used for this project. Unity is a cross-platform game creation system, including a game engine and integrated development environment. The engine is mostly used for development of video games for web-sites, desktop platforms, consoles, and mobile devices.

4.6.2 Preparation of the character for animation

When the graphic asset of the character was finished, it was split into smaller parts inside Adobe Photoshop. However, the parts are split in a correct way so that they will be combined together afterward. More than that the split parts were exported as individual image files. The character asset was divided according to the rig. It is important to know in advance what kind or rig will be used for the character before splitting the graphical asset.

Spriter operates in a way that it requires all the graphical assets to be stored in one location because this location is considered as the project workspace.

When all graphical assets were split and imported into Spriter, they were combined together and based on the character pose the rig was created. Then the rig and graphical assets were merged together in order for the rig to control the movement of the body parts of the character.

4.6.3 The planning of animation

Before starting the actual animation programming in a game, animation logic was prototyped explaining the animations and their transitions. This was a necessary step that would build a logic which would be implemented behind the character animations. The events that would trigger and stop the animations had to be thought though. In study example, the character had a total of five animations. There were idle_animation1&2, walk_animation, run_animation and run_to_idle animations.

The first problem that needed to be solved was the event that started the animation. It was assumed that the player would give an input by pressing the button, which would be the start of the animation. Also, when the player would release the button the animation should stop. When there would not be any input from the player the character should remain idle, thus the idle_animation1&2 should be executed, but if the input is received the character should start moving triggering either the walk_animation or the run_animation. It was assumed that the player input would start the walk_animation meaning that as long as there was input the walk_animation would be active.

The next thing to solve was the start of the run_animation. The most reasonable solution would be for the character to pass through the walk_animation before starting the run_animation. It would give more of a natural feel of how the run_animation stars. Next, it was assumed that the same input button as for the walk_animation would trigger run_animation.

Finally, the return to the idle_animation1 from the run_animation was planned. From animation principles, it is known that fast transition would most likely result in unrealistic and harsh transition. In order to prevent that, an extra animation the run_to_idle was added to create a smooth transition from the run_animation to the idle_animations. Throughout the animation planning process, a visual representation of animations (Image 19) was created as a simplified flowchart.

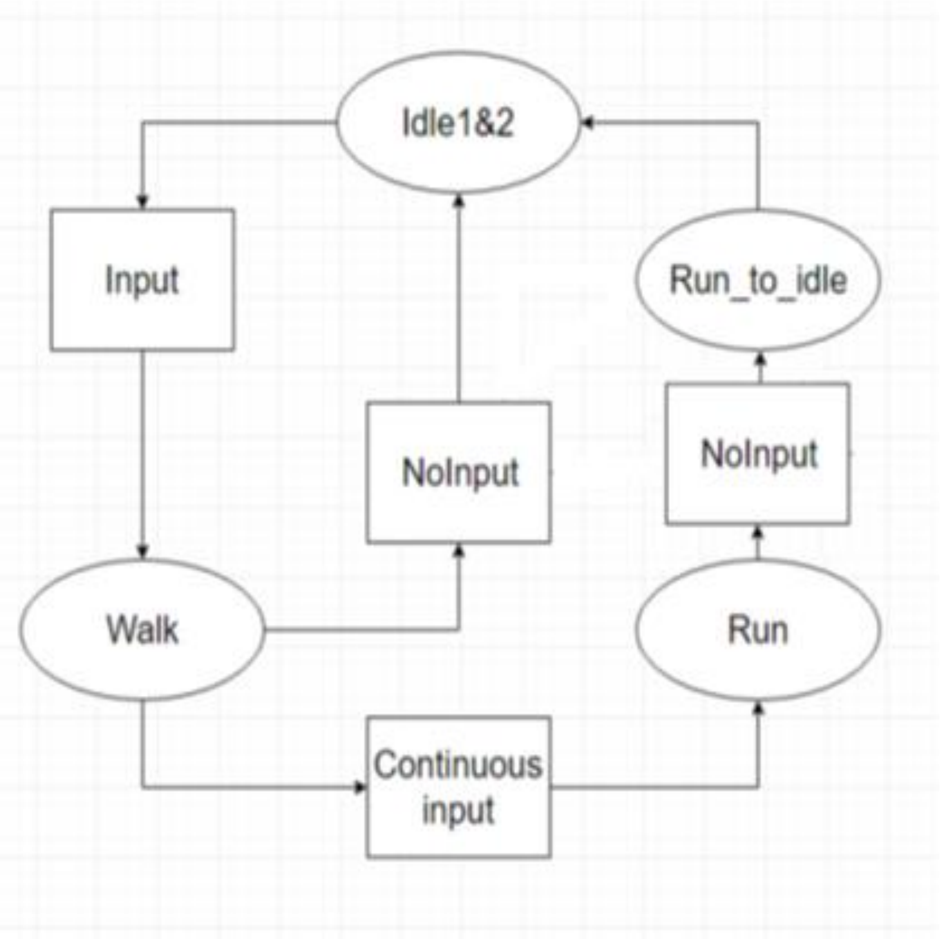


IMAGE 19. Shows the logic that was planned behind animation before programming the animations in game.

4.6.4 Animation creation

The animation creation is a time-consuming process which requires a lot of tweaking and adjustments. The method that was used to create animations was key-framing. The traditional animation method was chosen as a base for animation. All key poses of animations were captured using 100ms intervals. After that adjustments were made within every 25ms intervals giving extra frames to work with, which resulted in clean and plausible animations. After the animations were finally completed the next development phase consisted of adjusting and programming the animations in Unity Game Engine.

4.6.5 Game animation

The first task was to import the graphical assets and animations to Unity Game Engine. The reason why Spriter was used for this research is that it has a custom developed plugin called Spriter2Unity which gives developers a fast and efficient solution when it comes to importing the animations to Unity Game Engine. After the import, the plugin created animation controllers and character rig inside Unity Game Engine. More than that the plugin created a prefab file which acts as a template from which all new object instances were created.

After the import, the main project file became Unity prefab file which contained all character data. By default, the character prefab would contain an animator component which would interact with the animator controller (Image 20). The animator controller is a node based state machine that describes the animation behaviour by creating logic dependencies.

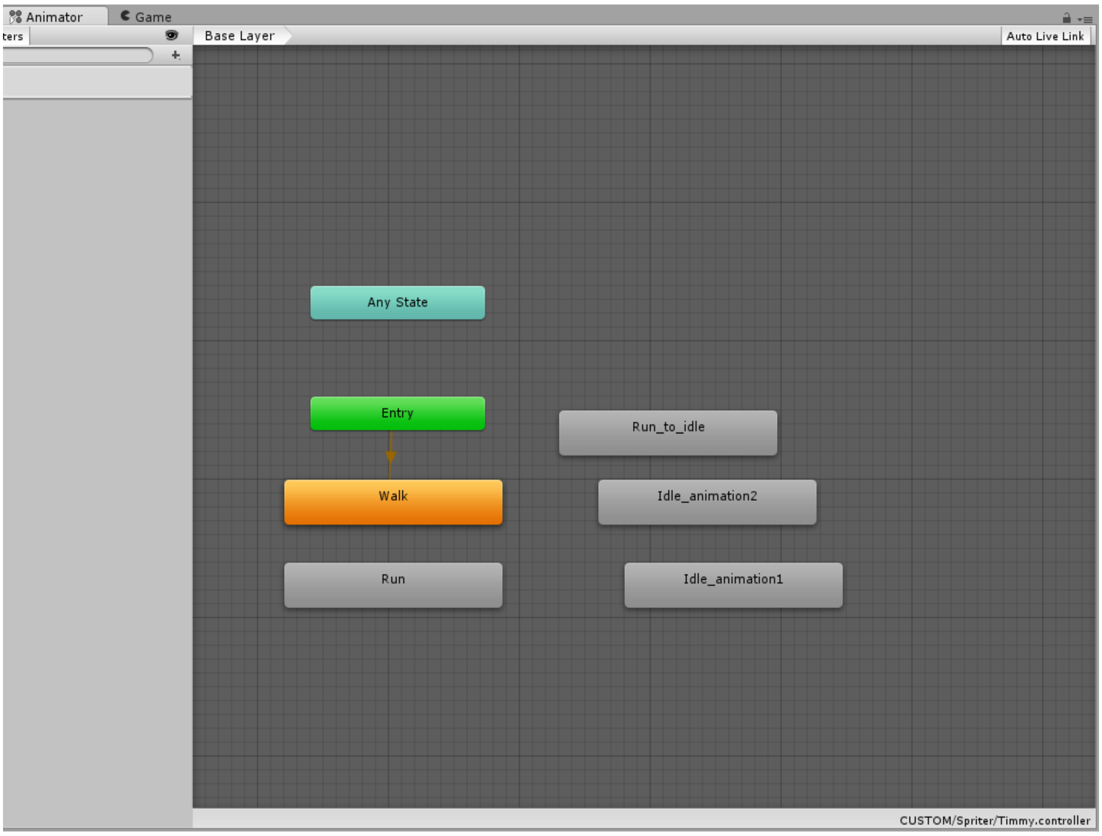


IMAGE 20. Screenshot from Unity Game Engine showing animator controller.

The animation logic that was prototyped before was implemented inside the animator controller to create basic transition structure (Image 21).

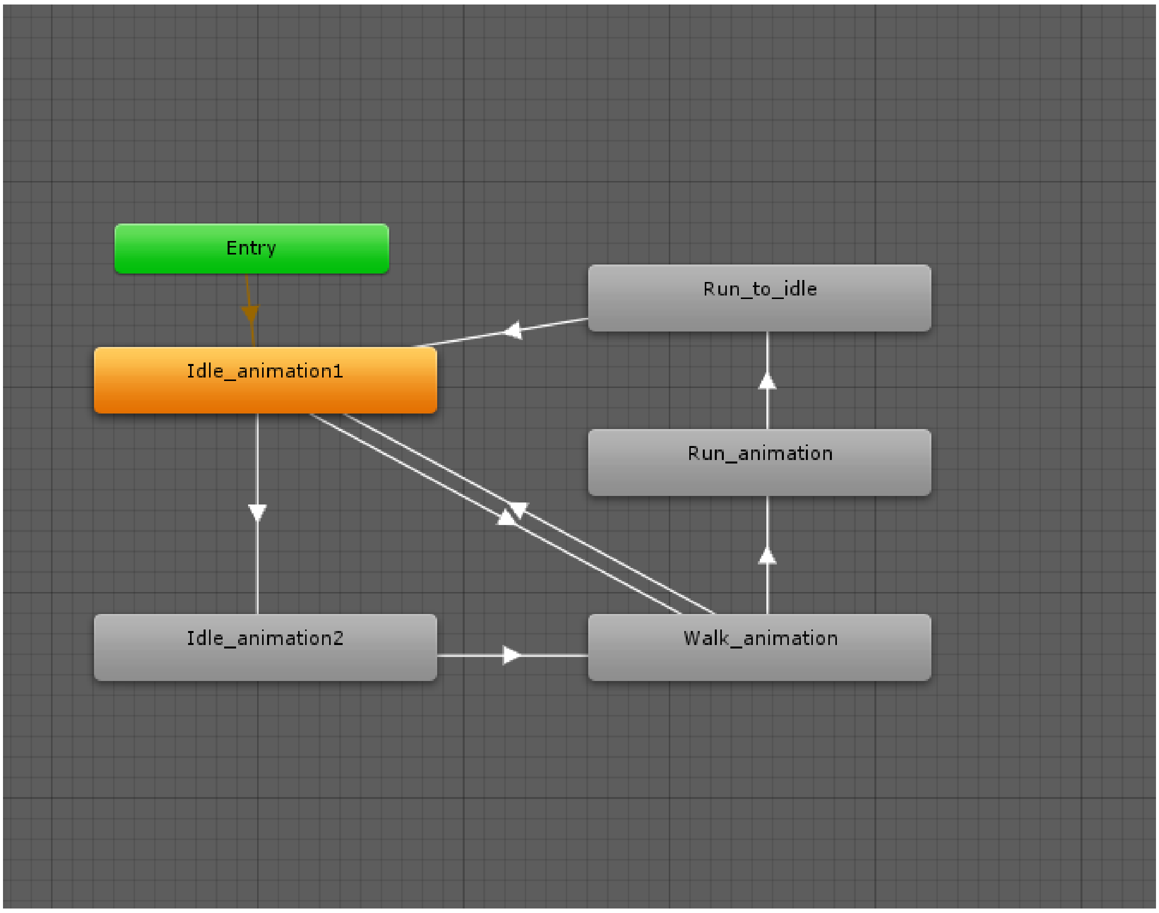


IMAGE 21. The animation transition structure according to animation logic.

The default state of the animator controlled is always marked as an orange block, according to the animation logic prototype, the idle_animation1 was the ideal choice for the default state. As a player starts the game the character should have some animation in order to clearly show the player that it is the game character.

After that, the transitions between the states were created. The animation transition allows the animator controller to change animation states. The transition is responsible not only for how long the time between the states is, but also it defines under which conditions the states should change.

First of all, the transition from the idle_animation1 to the idle_animation2 was assigned. It was done to underline the character design and bring more visual variety to the player. Meaning that the character would not be in the endless loop of one idle animation. Next, it was necessary to create the transition from the idle_animations to the walk_animation.

Moreover, the transition from the walk_animation to the idle_animations was required because at some point the player would stop and the character would be logically supposed to enter the default state which is the idle_animation1. Then the transition from the walk_animation to the run_animtaion was created, the character could not start the run_animation without passing the walk_animation. The reason behind that was that the player would see a smooth transition from the idle_animations to the run_animation. The last transition that had been created was the run_to_idle animation which allowed the character to stop the run_animation and get back to the default state. However, when the animation structure was executed it did not follow the transition order. The animation hit the endless loop. The reason that caused the loop was the lack of transition conditions.

As a result, the conditions were created for the animation structure. Their role is to explain and navigate how the transition occurs between the animations. The condition is set by a parameter that is created by the developer. Unity Game Engine allows creating parameter values such as Float and Integer, Boolean and Triggers. A transition can be set up to only occur when a certain condition is true, and these conditions are based on the values of parameters that are set up inside the Animator Controller. More than that a transition can have a single condition, multiple conditions, or no conditions at all. If the transition has no conditions, only the exit time parameter will be considered. When the exit time is reached, the transition will occur. If the transition has one or more conditions, the conditions must all be met before the transition will be triggered. (Unity Documentation)

The important part for the animator is the transition graph (Image 22) which indicates the duration of animations and transition between states.

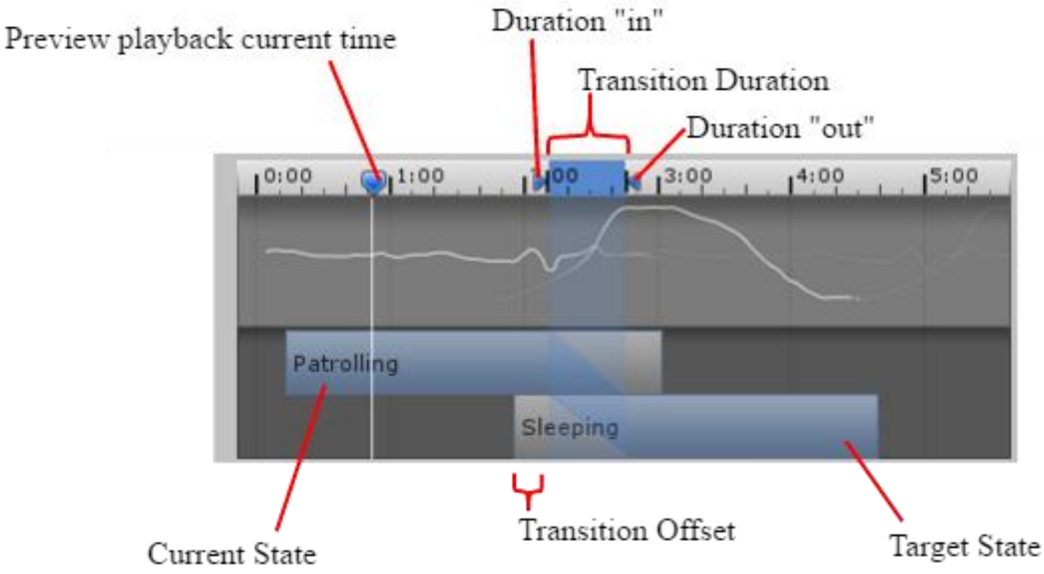


Image 22. The transition graph example. (Unity Documentation)

For example, the transition graph between idle animations was changed the most. The character had two idle animations and both of them were adjusted by animator. In order to create an appealing idle animation, the number of cycles, that the idle_animation1 did, was increased before switching to idle_animation2. This added more realistic feel to the character and took away the part when the character was constantly switching between idle states.

Once all conditions were set for the transitions, the testing part came in. The animator should be visually able to detect any defects in the transitions between the animations. Manually changing and testing the transition durations, with instant viewing of any applied changes, would help animator to exclude the unclean animation transitions.

In order to finish the character testing, it was put inside the game with movement controls. The next step was creating float parameter, called moveSpeed, that would control the character movement speed inside the game. Then the behaviour script was created from the animator controller to change the movement speed of character for any animation state.

The script is considered a custom component and often it is used to access other scripts attached to the same game object or even other game objects. The script inside itself contains different template events that could be called depending upon their need. OnStateEnter event was used. It was already contained in the behaviour file. Each time the transition switch would happen this event is called. The local variable `animator` referred to animator component where the behaviour script existed. The `animator.SetFloat` parameter allowed changing the value of the moveSpeed parameter.

The code below is the behaviour scrip of animator controller which set the movement speed of the character.

```
public class ChangeSpeed : StateMachineBehaviour {

    public float newMoveSpeed = 5; //default speed

// OnStateEnter is called before OnStateEnter is called on any
state inside this state machine
```

```
          override public void OnStateEnter(Animator animator,
AnimatorStateInfo stateInfo, int layerIndex) {
        animator.SetFloat("moveSpeed", newMoveSpeed);
          }
```

The movement speed script did not affect the actual movement speed of the character until it was connected to the input controller. Moreover, the main character project file required another component called Rigidbody 2D. Adding a Rigidbody 2D component to an object put its motion under the control of Unity's physics engine. Even without adding any code, a Rigidbody object would be pulled downward by gravity and would react to collisions with incoming objects if the right Collider component was also present. (Unity Documentation)

Another script needed to be created to access the behaviour script and to read the input in order for the player to interact with the character and move it in the game. By creating a new script, Unity Game Engine prepopulated class declaration and created start and update method. However, since the character would contain Rigidbody 2D component FixedUpdate function was preferred (Unity Documentation). The reason for this was that physics updates were carried out in measured time steps that did not coincide with the game frame update. FixedUpdate was called immediately before each physics update and so any changes made there would be processed directly.

`private Vector3 faceRight` is responsible for the side the character is facing by accessing the character file and updating the transform scale parameter. The values inside the brackets of Vector 3 represent x, y, z scale parameters so in order for the character to turn to the right or the left side the x parameter was switched from 1 to -1. In Unity Game Engine, `GetComponent` uses different style of call, it uses angle brackets <> before the normal brackets (). The angle brackets are there to get type as a parameter. Unity also had inputManager component where input axes and game action were defined. The Horizontal axis was used because it already had the input buttons assigned. `Input.GetAxis` returns virtual axis defined by Horizontal axis.

```
//Variables to describe components
private Animator myAnimator;
private Rigidbody2D myRB;
private float myAnimatorMoveSpeed;

private Vector3 faceRight = new Vector3(1, 1, 1);
private Vector3 faceLeft = new Vector3(-1, 1, 1);
// Initialization of variables
    void Start () {
        myAnimator = gameObject.GetComponent<Animator>();//
        myRB = gameObject.GetComponent<Rigidbody2D>();

        myAnimatorMoveSpeed = myAnimator.GetFloat("moveSpeed");

           }

    // FixedUpdate is called immediately before each physics update
    void FixedUpdate () {

        //If character moves to the right
        if (Input.GetAxis("Horizontal") > 0f)
        {
            transform.localScale = faceRight;
            myRB.velocity = new Vector2(myAnimatorMoveSpeed,
myRB.velocity.x);
            myAnimator.SetBool("movePressed", true);
        }
        //If character moves to the left
        if (Input.GetAxis("Horizontal") < 0f)
        {
            transform.localScale = faceLeft;
            myRB.velocity = new Vector2(-myAnimatorMoveSpeed,
myRB.velocity.x);
            myAnimator.SetBool("movePressed", true);
        }
    }
```

After that, the character was tested through the game by the animator to see if all animations were working and the transitions were plausible.

The method above proved itself as the most efficient way of creating animation, it was very responsive and efficient in comparison with previous two. Despite the fact that it still required software expertise, animation and programming skills.

5 CONCLUSION

The study was set to research the mobile game application industry, to dig inside one of the most successful application markets and see it through, to understand how it works from the inside. The study also focused on the differences between the animation creating methods for mobile games. Moreover, the study opened a number of obstacles that an individual can face, while seeking to enter the mobile game development industry.

The way that Apple App store has developed within seven years only leaves more questions to answer. What is the future for mobile applications and the industry itself? Will it collapse or continue to grow at the same speed it does now? Nevertheless, the mobile game development industry is flourishing, considering the fact that games are the applications that people want the most and the games occupy the major part or mobile application industry. Consequently, it is possible to observe how mobile game development evolves and its impact on mobile application markets.

In addition, the study showed the pipeline of different animation development methods and implemented them in practice to comprehend the most optimized method for mobile game development. As a direct consequence of this methodology, the research encountered a number of limitations, which need to be considered in order to successfully create animations using different techniques.

REFERENCES

Adjust, 2014. Birth, life and death of an app. A look at the Apple App Store in July 2014 *Adjust* [App analysis service] Available from: https://www.adjust.com/assets/downloads/AppleAppStore_Report2014.pdf

Allan, R. 2002. Review of the book: The Illusion of Life, Disney Animation. *Animation World Network.* [Online publishing group] Available from: http://www.awn.com/animationworld/walt-disneys-nine-old-men-art-animation

Apple Special Event, 2011. Video recording of Apple Special Event. *YouTube*. [Global video sharing website]. Available from: https://www.youtube.com/watch?v=dhdmZ7iNZYo

AppTrace Blog, 2012. Article about the "Zombie Apps". *AppTrace* [Free app analysis service]. Available from: http://www.apptrace.com/blog/2012-08-06/inside-zombie-land

Bash studio, 2015. Animation blog about 12 principles with examples. Bash Studio Blog [Online blog] Available from: http://br260097-sp.blogspot.fi/2015/11/12-principles-of-animation.html

Basic Character Rigging, Course. *University of Washington Computer Science & Engineering.* [Electronic presentation] Available from: https://courses.cs.washington.edu/courses/cse460/05sp/links/Rigging/basic_character.html

Bowate, C. 2003. Online course: Character Concept Art: From initial sketch to final design. *Skillshare* [Online learning platform]. Available from: https://www.skillshare.com/classes/design/Character-Concept-Art-From-Initial-Sketch-to-Final-Design/1310245862/project-guide

Bowcock, J. 2008. Apple App Store Press Release: iPhone App Store Download Top 10 Million in First Weekend. *App Store Library* [Electronic library] Available from: http://www.apple.com/pr/library/2008/07/14iPhone-App-Store-Downloads-Top-10-Million-in-First-Weekend.html

Calimlim, A. 2014. The Best Apps of 2014 by Apple. *AppAdvice* [Electronic journal] Available from: http://appadvice.com/appnn/2014/12/apple-announces-the-app-stores-best-of-2014-for-iphone-ipad-and-mac

Cook, T. D. 2015. Public speech of Chief Executive Officer of Apple Inc., at The Apple Worldwide Developers Conference (WWDC) [Video] 2015 Available from: https://developer.apple.com/videos/wwdc2015/

Cooper, J. 2012. A List of Mobile App Stores. *Mobysffiliates.* [Mobile advertising market place] Available from: http://www.mobyaffiliates.com/blog/mobile-app-stores-list/

De Stefano, R. A. 1999. The principles of animation. *The Electronic Visualization Laboratory (EVL) at the University of Illinois at Chicago (UIC).* [Electronic presentation] Available from: https://www.evl.uic.edu/ralph/508S99/contents.html

Dirks, B. 2012. Article about the best apps of the App Store in 2012. *AppAdvice* [electronic article] Available from: http://appadvice.com/appnn/2012/12/apple-unveils-the-app-stores-best-of-list-for-2012

Israel, L. 2016. Apple App Store Press Release: Record – Breaking Holiday Season for the App Store. *Apple Press Info*. [Electronic library]. Available from: http://www.apple.com/pr/library/2016/01/06Record-Breaking-Holiday-Season-for-the-App-Store.html

Jobs, S. 2010. Public speech. *Apple Press info*. [Electronic library] Available from: http://www.apple.com/pr/library/2010/01/05Apples-App-Store-Downloads-Top-Three-Billion.html

Jones, C. 2013. Forecast about Apples future. *Forbes* [electronic journal] Available from: http://www.forbes.com/sites/chuckjones/2013/12/11/apples-app-store-about-to-hit-1-million-apps/#de968c611870

Masters, M. Blog about 12 principles of animation interpreted by Mark Masters. *Digital Tutors.* [Online learning platform] Available from: http://blog.digitaltutors.com/understanding-12-principles-animation/

McCann, T. 2012. *The Art of the App Store: The Business of Apple Development.* [Electronic book]. Available from: Google Books

Meroz, M. Types of Animation. *Bloopanimation* [Blog] Available: https://www.bloopanimation.com/types-of-animation/

Miller, T. 2012. Apple App Store Press Release: Apple's App Store Downloads Top 25 Billion. *Apple Press Info*. [Electronic library]. Available from: http://www.apple.com/pr/library/2012/03/05Apples-App-Store-Downloads-Top-25-Billion.html

Miller, T. 2013. Apple App Store Press Release: App Store Tops 40 Billion Downloads with Almost Half in 2012. *Apple Press Info*. [Electronic library]. Available from: http://www.apple.com/pr/library/2013/01/07App-Store-Tops-40-Billion-Downloads-with-Almost-Half-in-2012.html

Miller, T. 2014. Apple App Store Press Release: App Store Sales Top $10 Billion in 2013. *Apple Press Info*. [Electronic library]. Available from: http://www.apple.com/pr/library/2014/01/07App-Store-Sales-Top-10-Billion-in-2013.html

Official guidelines for developers. App Store Resource Centre, 2010 *App Store Review Guidelines.* [Electronic document]. Available from: http://www.wired.com/images_blogs/gadgetlab/2010/09/app-store-guidelines.pdf

O'Grady, J. D. 2011. Best Apple apps of 2011. *Apple picks its top iPhone and iPad apps of 2011.* [Online article] Available from: http://www.zdnet.com/article/apple-picks-its-top-iphone-and-ipad-apps-of-2011/

Perez, S. 2014. Article that summarized information from Apple World Worldwide Developers conference (WWDC) 2014. *TechCrunch* [Electronic journal] Available from: http://techcrunch.com/2014/06/02/itunes-app-store-now-has-1-2-million-apps-has-seen-75-billion-downloads-to-date/

Pope, S. 2011. Apple App Store Press Release: Apple's App Store Downloads Top 15 Billion. *Apple Press Info* [Electronic library] Available from: http://www.apple.com/pr/library/2011/07/07Apples-App-Store-Downloads-Top-15-Billion.html

Porter, M. E. 1980. Competitive Strategy. Techniques for Analyzing Industries and Competitors. 3 – 44.

Presentation on Animation, 2012. Visualization, animation. *Tufts University School of Engineering* [Presentation] Available from: http://www.cs.tufts.edu/~sarasu/courses/comp150-2010sp/pdf/comp150-15-Animation.pdf

Ritchie, R. 2010. Picture representing "Hall of Fame" in 2010 from App Store. *iMore*. [Electronic newspaper and blog] Available from: http://www.imore.com/apple-launches-app-store-hall-fame

Schiller, P. 2011. Speech of Apple's senior vice president of Worldwide Product Marketing. *Apple Press info*. [Electronic library]. Available from: http://www.apple.com/pr/library/2011/01/22Apples-App-Store-Downloads-Top-10-Billion.html

Slivka, E. 2009. The publication of materials from Pinch Media presentation. *MacRumors*. [Electronic newspaper] Available from: http://www.macrumors.com/2009/02/20/appstore-secrets-price-drops-usage-and-ad-supported-models/

Statista, 2016. Most popular Apple App Store Categories. *Statista* [Statistic portal] Available from: http://www.statista.com/statistics/270291/popular-categories-in-the-app-store/

Thomas, J. and Johnson, O. 1981. The Illusion of Life, Disney Animation. Principles of animation. 47 – 71.

Unity documentation, Manual. Animation transition. *Unity Documentation* [Online documentation/API] Available from: http://docs.unity3d.com/Manual/class-Transition.html

Unity documentation, Manual. Rigibody component and FixedUpdate parameter. *Unity Documentation.* [Online documentation/API] Available from: http://docs.unity3d.com/ScriptReference/Rigidbody.html

Unity documentation, scripting API. Animator.SetFloat description. *Unity Documentation.* [Online documentation/API] Available from: http://docs.unity3d.com/460/Documentation/ScriptReference/Animator.SetFloat.html

Unity documentation, scripting API. FixedUpdate function. *Unity Documentation.* [Online documentation/API] Available from: http://docs.unity3d.com/ScriptReference/MonoBehaviour.FixedUpdate.html

White, J. 2013. The Best Apps of 2013 by Apple. *AppAdvice* [Electronic journal] Available from: http://appadvice.com/appnn/2013/12/apple-unveils-best-ios-apps-of-2013-duolingo-and-disney-animated-take-no-1-spots

Wingfield, N. 2008. Wall Street Journal interview with Steve Jobs. *Wall Street Journal* [Electronic newspaper]. Available from: http://www.wsj.com/articles/SB121842341491928977

Wolfe M. B. 2011, Summary of Apple Rewind Event. *AppAdvice* [Electronic journal] Available from: http://appadvice.com/appnn/2011/12/apple-reveals-app-store-rewind-2011

SAVONIA