GAM6001-18 Major Project

Individual Report

Assessment 02

McDonald, Bradley

1608793

Contents

[Abstract (Do Last) 2](#_Toc7681049)

[1.Background 2](#_Toc7681050)

[1.1 Aims of the project. 2](#_Toc7681051)

[1.2 The Team Composition 2](#_Toc7681052)

[1.3 The authors role in the team 2](#_Toc7681053)

[1.4 The state of production from first milestone 3](#_Toc7681054)

[2.Introduction 3](#_Toc7681055)

[3.Technical Art 4](#_Toc7681056)

[3.1 A Designer Creating Technical Art tools 6](#_Toc7681057)

[3.1.1 Brief discussion of relevant literature 7](#_Toc7681058)

[3.1.2 How this work was approached (design etc) 7](#_Toc7681059)

[3.1.3 Problems, solutions and evaluation of this work in comparison to research undertaken 8](#_Toc7681060)

[3.2 Testing method 9](#_Toc7681061)

[3.2.1 Brief discussion of relevant literature 9](#_Toc7681062)

[3.2.2 How this work was approached (design etc) 9](#_Toc7681063)

[3.2.3 Problems, solutions and evaluation of this work in comparison to research undertaken 9](#_Toc7681064)

[4. Potential political views on the project 10](#_Toc7681065)

[4. Reflection 11](#_Toc7681066)

[5. Conclusion 11](#_Toc7681067)

[6. Bibliography 12](#_Toc7681068)

[Bibliography 12](#_Toc7681069)

Figure Table

[Figure 3- Moving Platform Tool 6](file:///E:\Uni%20Work\3rd_Year\Dissertation\MCDONALDBradley_Draft_Dissertation004.docx#_Toc6577879)

[Figure 4- Fence Tool 6](#_Toc6577880)

# Abstract

This report covers the intricate details of the development of the project Koala Rama from one of the designers. The topics covered are to be backed up with relevant literature and reliable sources. The topics mentions were a key part of development may that be in the start, middle or end. Topics such as technical art, testing and an intricate detailed look of how the project could be perceived after if it was released.

# 1.Background

The Team, for this module, created a 3D platforming game. The game was based on a bi-pedal Koala, who has to rescue his friends throughout a variety of worlds by utilising a variety of mechanics. The title of this game was Koala Rama.

## 1.1 Aims of the project.

The team wanted to create an aspiring 3D platformer with added unique gameplay mechanics whilst also including generic 3D platformer mechanics (such as Walking, Running, Jumping and double Jumping) as to still seem familiar to people who have played 3D platformers before. The team decided to add the unique elements as to set it aside from other previously released games in the same genre. Games of the same genre consist of Mario Odyssey (Nintendo,2017) and Spyro Reignited Trilogy (Activision, 2018), though these are developed by different companies they fall within the same genre, a 3D platformer. A 3D platformer as described by (Minkkinen, 2016) “’Platformers’ or ‘platform games’ are games that mainly revolve around a character controlled by the player, which runs and jumps to avoid obstacles and/or to defeat enemies”. This was the main gameplay element of the project, Koala Rama to explore multiple world whilst jumping to multiple platforms floating or not, avoiding hazards along the way in the form of obstacles or enemies.

## 1.2 The Team Composition

The team was made up of five third year undergraduate games students from various disciplines, an artist, a programmer and three designers. Each person applied skills that adhered to their role. Whilst branching into artist or programmer depending on workload. The artist created 3D models, unwrap, textured and imported into engine for the designers to place within the game. The Programmer focused primarily on Enemy AI and fixed bugs when attempting to build the game. The Designers would create tools and polish to the game in the form of particle effects or sound whilst creating and designing multiple levels for the player to navigate in Unreal Engine 4 (UE4), (Epic Games,1998).

## 1.3 The authors role in the team

The Author carried out his role in the team as a designer in a small independent game development (indie) team would. To quote Mike Bithell, an indie game designer, writer(Freeman, 2015) of the guardian newspaper interviewed Mike Bithell who went onto say when asked the question: How varied is your role? Mike Bithell replied ‘*But then I still design game mechanics, I still do lots of coding, I still write the scripts, and do lots of things. It really doesn’t come under a single banner, so I use designer, because nobody really knows what that is anyway’*.

Which implies to use and learn a wide range of skills within the project, this is different than being a designer on a large AAA team, where the role would have been more specific, interviews with industry developers such as (Ismail, 2018) says how they have specific team members for specific jobs where it be seeing if ancient hieroglyphics are correct or researching locations. The role of being a generic designer meant going into different aspects of game development, such as; Gameplay Programming, Animating, UI Design, Technical Art and VFX. This allows for constant learning and developing new skills as the project progressed.

## 1.4 The state of production from first milestone

The prototype was at as stage where the core mechanics were working, such as; walking, jumping, rolling and throwing the boomerang. With these mechanics there were a small number of visual glitches to do with animations, of which needed polishing. The game itself was playable with one main level. However, with the overall design of the level done and with the better tools developed for level design, it became easier and more apparent to make more levels. The team then began planning on creating more levels and refining the ones already made.



## 2.Introduction

As one of the Designers of the team, many roles had to be undertaken to provide a quality product. The document will go into detail of these roles, what they were, how they were done and what effect these implementations had on development. These effects can range from implementing feedback from testing results, to setbacks in development which led to rework of the original design. This document will also go into detail of relevant literature and research of which informed the approach in this project, and if these findings actually were reflected in the project or not. The document will go into detail of the work the author has done on the project, the testing of the project and the potential political implications of the project. This will be followed by a reflection and conclusion of the authors work through the project, Koala Rama.

The author will relate back to previous submitted work at relevant parts to show how the project developed based upon certain aspects of the submitted work and project. This will primarily be focused upon when discussing testing and implementation of design into the developed project, including the potential political aspects of the project.

# 3.Technical Art

This section will cover the role of a technical artist and how the author took up this role through development to provide a greater service to the team and project.

The job of a technical artist as defined by (Full Sail University, 2018)

“*Something of a hybrid between an artist and a programmer is the video game technical artist. This person works under the direction of the art director and technical art director, and is responsible for the systems and tools associated with creating and porting art assets*”.

Though that is a definition of a technical artist, it is not the clearest. As an article by (Sokanu, 2019) states that

“*The role of technical artist is a relatively new one, but it is becoming increasingly important as consoles and PC hardware become more complicated. A technical artist works closely with the lead artist and the creative director, as well as the lead programmers. Their responsibilities include setting up and maintaining the workflow of art production, deciding which art packages and tools a studio should use, investigating new techniques, and then going ahead and implementing them*”.

Both quotes interlink with saying that the technical artist works alongside programmers and artists however verge when discussing tools, (Full Sail University, 2018) makes it seem that the technical artist simply decides on the tools that are being implemented whereas (Sokanu, 2019) goes into that the technical artist actually creates the tools needed for production. A technical artist at Ubisoft Bucharest (Ubisoft, 2016) goes onto say that “*In the beginning we’re focused on creating the pipeline, workflow and the tools the artists can work with*”. This keeps with how the author approached the role, creating various tools for the designers and artists whilst maintaining a stable version of the game through optimisation.

To take up the role of a technical artist, as based upon by the above, is to utilise both art and programming to create a multitude of tools so that Designers can speedily design, create and test level layouts and adjust them easily based on feedback. Be that feedback internal or external. That was one aspect of the role. As the author’s role of being a designer, with experience in both art and programming. The job was possible. With existing knowledge of game design, allowed the author to tailor the tools so that a fellow designer would be able to fully utilise the tool with little or no tutorial. As they knew themselves of what a tool needed and how it needed to function in the editor. As designers have an advantage of knowledge in other disciplines, as described by (Schell, 2008) in the Chapter ‘What Skills Does a Game Designer Need?’ she foes onto describe “ *In Short, all of them. Almost anything that can be good at can become a useful skill for a game designer*”. She then goes to list a number of skills from Anthropology to Visual arts. As the team was mostly designers and were all disciplined in these wide variety of skills, communication was simple through development of tools and of the project overall.

Another task that comes with the role of a technical artist it to optimise the game, by viewing the in-engine profiler, to see if any imported assets, created by the artist, effect the game. Either by performance (Frames per second) or if they create visual glitches (missing polys or wrong facing normals). This can lead to unanticipated stops in production of which then need to be solved before any new alteration or iterations are added to the game.

Figure 2- Unreal Engine Profiler- ‘Koala Rama After Optimisation’. Final Build

Figure 1- Unreal Engine Profiler- ‘Koala Rama Before Optimisation’. Early Build

The green bar seen in figure 1 above, shows the memory usage for the Third Person character. This visualisation aids in debugging which assets need correcting. In this instance it was the Third Person Character Blueprint that was taking the majority of the memory. So, to optimise the game the author researched a variety of methods to optimise the game. Originally in the design of the game the player was to gradually gain their abilities such as double jump and roll, with this in mind the author researched ‘Composition over inheritance’ a resource is (Johansson, 2015) whom details to ‘define objects of what they can do rather than what they are’. So that later I could define the playable character as an object that could ‘fight’ or ‘roll’ rather than constantly checking if they could due do those things by what it is. This method was ideal however later in production the team decided it was better to ensure the player has all the abilities from the start of the game. This decision was made due to internal testing and also development time constraints, the team could not get the ‘unlock ability’ system working whilst also implementing other assets into the game. As the system was never in place, the level designer never designed a level based upon gradual unlock and made them more open as they were used to having everything unlocked.

## 3.1 A Designer Creating Technical Art tools

The tools that were created were to primary aid designers in prototyping levels and to quickly alter levels for further development. The tool as shown in figure 3 is the moving platform tool. The designer can place a platform of any mesh they want and designate the location they want it to go, the designer can also decide how long they want the platform to get there over a designated amount of seconds, they can also chose if they want the platform to rotate and in any direction they want, the platform will always go back to its original location and rotation. This made implementing gloating platforms and jumping puzzles easy as the designer can simply place them in and with the seconds over time feature can easily match up two or more platforms in a row so that they do not become out of synch.



Figure 1- Moving Platform Tool

The Fence tool as shown below in figure 4 allows the designer to quickly place fences around a given area whilst maintaining the same direction as the previously placed fence as to give the fence aligning some continuity. This feature can be overridden easily by manually adjusting the fence to create more sharp shapes such as for creating corner or to square of a section. Using this method also decreased draw calls within the level which means the level loads faster, as the single spline mesh is loaded rather than individual meshes. By utilising spline meshes we are reducing draw calls so that artists and level designers were able to place more assets into the level there by making them more decorated and pleasing for the player to look at and attempt to make the feel like part of the world.

Figure 2- Fence Tool

## 3.1.1 Technical Art In Industry

An article by (Haynes, 2008) writes about the multitude of roles that a technical artist has (page 2), for instance the tools programmer, of which was briefly just discussed above, goes onto agree with how the tools technical artist develops tools to assist in pipeline such as the artist. It used to be the job of the programmer to make these tools, but he goes onto say how the programmers had trouble communicating with artists, but as the technical artist role has become more apparent in the industry, a person with experience in both art and programming the communication problem seems to dissipate.

## 3.1.2 How this work was approached (design etc)

The work of technical art was not originally planned but rather was needed by the team as the project progressed. The game needed to become more stable as the game become larger, with more dense levels and more levels altogether. Optimisation techniques were researched and implemented as discussed. The tools idea came out with designers complaining of how it was taking several hours to do a single job which when using the tool could be done in minutes, tools like the path generator and the fence generator allowed the designer to quickly implement their ideas. This made the designers more adamant to level design, which were tested, with the results better levels for the players. This method is similar to the games company Volitions technical art development, as their workflow of implementation of technical art is shown in Figure 3. The tool would be requested by a fellow designer and if the author felt that the tools was achievable to make whilst working on their own work it would then be created. Once created it would be sent to the designer who requested it for feedback if the tool was not satisfactory or not what the designer had in mind a clearer detail of the tool was made as well as

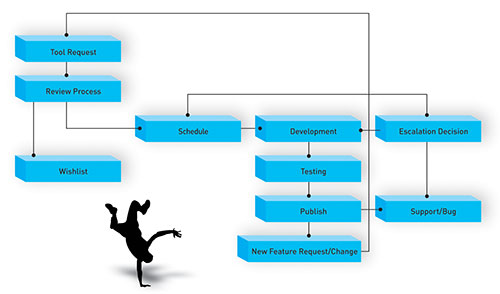


Figure 3-Volition's flowchart for feature requests as well as a change management plan.

## 3.1.3 Problems, solutions and evaluation of this work in comparison to research undertaken

The problem with technical art is that there a numerous amount of aspects to research before anything can be implemented. This can range from optimisation by using the in engine profiler as described in Figure 1 and Figure 2 to developing tools for the designers and researching how to create the tools within the engine whilst also updating and maintaining the tools to keep the demand of the fellow team members of whom were using them for primarily level design unless the tool was designed for another use. The tools themselves needed further testing before being fully integrated, and if the tool was updated it would then need further testing by designers to make sure the new version worked.

## 3.2 Testing method

When testing Koala Rama throughout development the team did internal testing. Internal testing is by our own definition is that as we discovered bugs through implementation, we would then report them to the appropriate team member. For instance, if an asset bug was discovered, the artist would fix it, if a compile error was spotted the programmer would fix it. This contradicts how triple-A studios internally test. As described by (Barlog, 2018), Creative Director of Santa Monica Studios in the interview he tells that internal tests were done every two weeks by their dedicated Quality Assurance team. As the team is small, we tested the game ourselves, after time branched out to external testing. External testing was done over multiple sessions and multiple builds. Once the team had a build that was contempt with testing, we then tested that build. The testers were a mix of students and Lecturers. All recorded data from the testers was consensual and all the testers were the appropriate age for testing. If anyone was under the required age their testing data was not recorded. The students testing was either from Bolton College or The University of Bolton. The University students were fellow games course student studying in games design, games art or games programming.

Each tester was to play the tutorial level in order to see if they understood how to play the game and its objective and then the testers play one other level of the game. The level they played depended on which testing station they were at. The team agreed to set a maximum of ten minutes playtime per level and recorded if the tester completed the level or not, if they did their time was recorded. The tester would be asked to report bugs if they encountered any, which we would fix before the next build and repeat the process. If the next builds test reports didn’t contain any of the prior bugs those bugs were then classed as fixed.

The following is the change log to show the fixes and improvements from testing different builds.

After Testing Build 01

* Changed enemy locations
* Changed Rolling to work when moving rather than stationary
* Boomerang now flies to point in the world whenever attempting to throw rather than twirl
* Locked FPS at 60
* Changed Tutorial Dialogue key words to Capitals
* Fixed Quit button not working in game over screen
* Fixed spelling mistakes and grammar.
* Can now Roll to attack enemies
* Fixed bug when trying to throw boomerang when; Running, Rolling and in air

After Testing Build 02

* Fixed bug where in the main level and yellow key quokka will get randomly freed
* Made Keys more visible
* Visual animation for picking up a key
* Animations are now independent from each other
* Displays how many keys you need to enter the portal above the respected portal
* Can no longer talk to the trapped quokka after they are freed
* Adjusted boomerang cross hair to be more accurate
* Controller now works with all UI
* Better overall mixed sound
* Added second path in MichaelLevel02
* Added more coins in MichaelLevel02
* Fixed boomerang sound not playing when coming back to the player
* Added more targets in MichaelLevel02

When designing the game, the team were inclined to create a game with many mechanics all of which needed to be tested as they were implemented into the game. As to make sure that the various mechanics do not interfere with one another and cause bugs, which they did. Attempting to do multiple actions at once, for instance, trying to aim the boomerang whilst rolling would result in the player not being about to throw the boomerang at all. Meaning that at some points where the player needs to hit a target the game was incompletable. That bug was discovered by various external testers, as developers of the game, the testing is internal. As the team developed the game the team played it properly, so this bug never occurred until external testing was done.

## 3.2.1 Brief discussion of relevant literature

## 3.2.2 How this work was approached (design etc)

## 3.2.3 Problems, solutions and evaluation of this work in comparison to research undertaken

# 4. Potential political views on the project

## Australian Culture

With the increase in political movement and censorship in recent years as shown by (Mochizuki, 2018), (Grayson, 2019), it is wise to monitor what is being put into the game and to properly research before anything is implemented (Ismail, 2018) of Assassin’s creed goes into how much depth their studio goes into research in order to properly respect the culture and history of the real places they base their games. For our project we based our project in a fictious setting however some may feel that it is basing upon the stereotypical Australian culture. As the game Koala Rama contains indigenous animals to Australia such as Koalas, Quokka and Kangaroos. The main character, of whom you control, their main weapon is a boomerang made famous by Australian culture. Residents of Australia may not take kindly to using a form of sterotype in the game and may demand for either a censored version such as by changing the weapon or changing the character models, or simply not permitting it for sale in Australia. As Australia is not new to banning certain games (Walker, 2018) though most of these games are mostly for a mature audience their reasons can stem for violence, sex and drugs. Since in the game you play as a koala collecting golden eucalyptus leaves, the main food source for koala, most people think they get a drug high from these leaves so people could assume that we are prompting drug use, just as they claim the same with mario, collecting mushrooms (Lee, 2018).



## Fantasy Animal Violence

Since the game contain fantasy violence of animals harming animals it can relate to PETA’s movement against Pokémon where by PETA suggested that “Pokémon are not ours to use or abuse. They exist for their own reason.” With this statement created a parody game titled Peta’s Pokemon Black and Blue – Gotta free them all. Where you play as a Pokémon exploring the world and killing humans whilst freeing other Pokémon. This was to represent a movement of real-world caged animals of which are captured and made to perform tricks for amusement. Which is very similar to the project where the main objective is to find correctly corresponding keys to free your captured friends.





## Capitalistic propaganda

Throughout playing the game the player in encouraged to collect coins in order to progress to newer levels, however some could view this as a capitalism propaganda or a ‘Far Right’ movement. By enforcing to the player that the game does not progress unless you have a certain amount of credit.

## Censorship

The main character in the project as do all other characters in the game do not wear any pants, however some wear shirts which can lead to the idea that they in fact exposed. Though the game does not show anything explicit it. Sony told that they are enforcing new censorship guidelines as depicted by (Mochizuki, 2018) of the Wall Street Journal, are “so that creators can offer well-balanced content on the platform” and gaming “does not inhibit the sound growth and development” of young people.

“Sony is concerned the company could become a target of legal and social action,” a US Sony official said to the wall street journal.

An example of a released game of which had to follow Sony’s new guidelines was Devil May Cry 5 in which they use additional lens flares to obscure a female bottom. This censorship is only on the PS4 (Grayson, 2019) as the censorship cannot be found on PC or Xbox One.

With these new censorship rules in place for Sony, if Koala Rama were to be ported to PS4 they may or may not demand a censored copy of the game whereby the characters are forced to wearing clothing, differing from the developer’s original design.

## 5. Reflection

## Individual Reflection

When reflecting on the project it is important to remember how it started, developed and ended. When starting the project, the planning could have been done better, such as the design document and Whitebox. As for when developing the game, the design document was almost forgot about, which lead too making content for the game what was never discussed or talked about. This used up valuable time and resources. Though with this additional content the team did hit all their milestones, meaning we were then able to implement this additional content.

When in the middle of development, the work load was not shared equally I find, as some people in the group either did not do the work or did it too such poor quality that somebody else had to do, needless to say the work got done in the end or at least till the next milestone.

Towards the end once we fixed all the problems with how to make a successful build it was all polish, adding sounds, particles and testing the game. Fixing any problems found, I do wish we tested more externally sooner before deadline so that all the bugs found would have had longer to fix and polish. The majority of them were fixed and it was better to know about them and learn how to fix them than to not know of them at all.

## 5.1 Team Reflection

The team worked very well together for the most part, though there were some communication problems which leads back to what was previously stated about creating additional content without consulting anyone. Though this additional content was created after the requirements was done for the milestone, this time could have ben used towards the following milestone as the next few weeks’ worth of work was displayed on the HacknPlan, assigned to the appropriate people. The team was very well formed, if the designers needed something off the artist they would ask if it done and if not, when would it be. To which the artist would give an honest answer, if the requested work was a week away that was fine, as the designers can do other things whilst that it being taken care of or would work with what the artist had just done and implemented. Giving the models in game checks as stated in the technical art section of this document. Working with the programmer was frustrating to start with as there were several delays of the breaks which needed fixing, else production of the game would have to stop for the majority of the team till the issues were fixed, the main problems were with builds or source control. The build problems came with the designers using the unreal engine in visual coding system called blueprint whilst the programmer was doing C++ programming language. Though the blueprint system is based on C++ is was difficult at first to merge the two into a single buildable project.

## Team Contribution

I do feel that when working on this project some people worked harder than others to deliver a better product, and this came with multiple reasonings. Now though the designers were using blueprints when implementing into the game, the skill level between the designers was massive. So, when a lower blueprint skilled designer made an object I, a higher programming skilled designer would look over it. Most of the time I found myself having to redo their work, simply because it created too many bugs or could have been done a more efficient way. I would try to explain to designers how to do it properly and where they can look to learn but instead never improved upon their skills. Giving me more jobs to do.

With the programmer struggling with the unreal engine, I had to help them too. So, I found myself for a time trying to explain unreal and blueprints and answering questions. When the programmer ran into a problem, they would immediately get me to help, thereby preventing me from doing my work. I felt like I was doing most of their work too at times since it only worked when I was there to fix it. The programmer more so became a QA member at that point constantly playing the game and calling out issues or problems of which I was made to fix since I was the only one with the skill to do it, however this took time out of what I was supposed to be doing and increased my work load. I did however learn a lot and increased my knowledge in programming and team management. This was referred too at the start of the document where I stated the authors role, my role, was more of that of an indie developer working in a lot of aspects of game development because I had to.

## 6. Conclusion

At being at university provided the author with many skills both as personal character development and as to the desired job within the games industry. Through the university course gave an exposure to the insider knowledge and work ethic that is to be expected come when coming to the industry. Something one cannot find in many places. This knowledge and ethic came into great practise for when developing the group project.

Though the developed skills taught in university are very good, a lot of time needs to be spent on research and improving on the skills you already have and acquiring new ones. Doing this in the break between terms has offered me great knowledge and insight on how to communicate with other people in many branches of the industry, being taught as a designer with other designer. With being closed off from the other groups for the majority of the time, it was great to communicate with a wider range of people outside of the acquired skill set to then learn more from.

At the start planning was a problem, as the author comes from a more scientific academic history where you are simply graded on a exam at the end of term and that is your grade. You could revise an hour before, keep the knowledge within your short-term memory and do the exam and do well. Game development is a completely different thing you can tell straight away if someone has spent and hour or 100 hours on a project and this time investment needs planning. This effect came mostly in the second year when first introduced to the artists in a group exercise and now you have to organise around other people. Failing at that made me realise how important it was to prepare for in our major project.

The major project taught the author how to fully utilise the skills gained and use them effectively within a small team environment, coordinating when certain tasks were done and if they were not done what to then do, certain things do in that situation is to talk to the team and discuss when it would be done or if it was actually possible, this would be determined by the person doing it to see if it was within their skill set. Someone else could take up the role or the task could be dismissed or put aside for later implementation if there is extra time.

Audio Notes

Ashraf Ismail (Game Director) of Assassins creed- <http://interactive.libsyn.com/ashraf-ismail-of-assassins-creed>

-Talks about respecting culture and history (32:25

-Prototyping mechanics (44:10-44:40)

Tim Shafer -<http://interactive.libsyn.com/tim-schafer-of-double-fine-productions>

-Brainstorming

-Humour in games

-Crowdfunding

Jenova Chen (thatgamecompany) <http://interactive.libsyn.com/thatgamecompanys-jenova-chen>

-interpreting test data like a doctor (36:50-

(37:30-42:50)

42:50-) “think like an experience doctor, evaluate the data from experience rather than from knee jerk reaction”

Cory Barlog (Creative director) <http://interactive.libsyn.com/cory-barlog>

-Implementing features (47:25-48:15

Testing (46:30-46:50) Playtesting every two weeks

-level design (50:00- 51:30

Ian Dallas -<http://interactive.libsyn.com/ian-dallas>

Mechanics -13:50

<https://www.theseus.fi/bitstream/handle/10024/94503/Ruuska_Essi.pdf?sequence=1>

QA Thesis

# Bibliography

HacknPlan

Unique New Academic Degree in Visual & Game Programming for Southern California; Curriculum to Train 'Technical Artists' for Animation and Video Game Industries 2006, , PR Newswire Association LLC, New York

# Bibliography

Barlog, C. (2018, May 25). God of War's Cory Barlog. (T. Price, Interviewer)

Full Sail University. (2018). *Technical artist*. Retrieved from GetInMedia.com: http://getinmedia.com/careers/technical-artist

Grayson, N. (2019). Devil May Cry 5 Slightly Censored, But Only On PS4. *Kotaku*, 1.

Haynes, J. (2008). The Code/Art Divide: How Technical Artists Bridge The Gap. *Gamasutra*, 1-3.

Ismail, A. (2018, August 14). Ashraf Ismail of Assassin's Creed. (T. Price, Interviewer)

Johansson, M. P. (2015, 9 20). *Composition Over Inheritence*. Retrieved from Medium.com: https://medium.com/humans-create-software/composition-over-inheritance-cb6f88070205

Lee, D. (2018). Top video games depict unrealistic drugs use says study. *BBC NEWS*, 1.

Minkkinen, T. (2016). *Basics of Platform Games.* kajaanin: kajaani university of applied sciences.

Mochizuki, T. (2018). Sony Cracks Down on Sexually Explicit Content in Games. *The Wall Street Journal*, 1.

Schell, J. (2008). *The Art of Game Design: A book of lenses.* Boston: CRC Press.

Sokanu. (2019, 04 15). *What does a technical artist do?* Retrieved from CareerExplorer: https://www.careerexplorer.com/careers/technical-artist/

Ubisoft. (2016, 04 14). *Tehnical Art At Ubisoft.* Retrieved from Youtube: https://www.youtube.com/watch?v=mJCkPqpn\_Zk

Walker, A. (2018). A Look Back At Some Of The Games Australia Has Banned. *Kotaku*, 1.

Freeman, W. (2015, Oct 13). Mike bithell: 'the highs and lows of creative freedom are very addictive'. The Observer Retrieved from <https://search-proquest-com.ezproxy.bolton.ac.uk/docview/1721796971?accountid=9653>

EPIC GAMES. (1998) *Unreal Engine.* [Software] Unreal Engine 4.20.3 Cary, NC: EPIC GAMES.

Hawkins, BM 2005, Real-Time Cinematography for Games, Charles River Media, Hingham. Available from: ProQuest Ebook Central. [1 April 2019].

Nintendo Entertainment Planning & Development.(2017) *Super Mario Odyssey*.[Download] Nintendo Switch. Kyoto, Japan: Nintendo

Toys For Bob. (2018) *Spyro Reignited Trilogy*. [Disc] PlayStation 4. Santa Monica, CA: Activision

<https://www.theseus.fi/bitstream/handle/10024/119612/Thesis%20-%20Toni%20Minkkinen.pdf?sequence=1>

<https://www.gamesindustry.biz/articles/2013-10-28-going-indie-im-more-scared-of-staying-in-aaa-right-now>

