**(Game Title Here)**

Word Count: 2636

Alexander McRoberts, Kyle O’Donnell

Contents

[Game Outline: 2](#_Toc495393669)

[Initial Team Roles: 2](#_Toc495393670)

[Revised Team Roles: 2](#_Toc495393671)

[Graphical Information: 3](#_Toc495393672)

[Level Information: 5](#_Toc495393673)

[Gameplay: 5](#_Toc495393674)

[Story: 6](#_Toc495393675)

[Testing and QA: 6](#_Toc495393676)

[Target Audience: 6](#_Toc495393677)

[Scope: 6](#_Toc495393678)

[UI: 6](#_Toc495393679)

[Milestones: 7](#_Toc495393680)

[Requirements: 7](#_Toc495393681)

[Design Approach: 7](#_Toc495393682)

[Resources: 8](#_Toc495393683)

[Work Done: 9](#_Toc495393684)

[Information Sources Used: 10](#_Toc495393685)

# Game Outline:

To add: A game overview, summary, theme and to justify them

Overview: The game that we plan to make is a fairly lighthearted 3D platformer game focusing around our protagonist (….) someone with the power to absorb powers from defeated foes much like Megaman from the Capcom franchise of the same name. The game would focus on this element as the protagonist wanders about the in game world seeking to (….) via utilizing the assorted powers the player can obtain.

The game’s main selling point is the ability to gain powers based on the enemy you just defeated and using them to progress in the game.

The theme of the game as it were would be a pretty cheerful one as all we’re really aiming for is for the player to enjoy themselves with the powers and world offered rather than making things deep and philosophical. This design choice was selected both because we don’t really have the time, and skills, required to make something truly deep and interesting thematically and because we at least have some experience with trying to make fun games in the past.

# Initial Team Roles:

Initially the team established consisted of three team members and the work was allocated as appropriate. Though it’s important to note that every team member will be doing some of everything anyway.

Alexander McRoberts: Documentation Lead + Team Leader

Kyle O’Donnell: Lead Programmer

Jordan Reilly: Lead Asset Creator

Justifications:

Kyle was assigned the lead programmer role primarily because he has either the most, or the second most amount of experience with the Unity engine which is what we’ll be using to make the game. In addition to this he already had some experience making platformers in Unity as seen when he provided the base of one during the second week.

Jordan was assigned the role of asset creator largely on request as he, apparently had a fair bit of modelling and animation experience.

Alexander received his primary role, the role of documentation lead, because of his prior experience making Game design documents in other circumstances whether for college or in his free time. In addition in was usually the highest rated work handed in when he submitted work in college suggesting he was better at documentation than other tasks like coding or animation. His secondary role was given to him more because of attendance than anything else as he’d been the most consistent member in by that point meaning he’d be around to interact with all of the members whether they’re in or not.

# Revised Team Roles:

Due to complications with certain team members, namely Jordan Reilly transferring to another course his role of asset creator had to be re-distributed amongst the two remaining team members. For the most part both Alexander and Kyle kept their previous assigned tasks but were given new work to do as well leading to the roles to be as follows.

Alexander McRoberts: Documentation Lead, Team Leader, Assed Creator (concept art and basic modelling)

Kyle O’Donnell: Lead Programmer, Asset Creator (modelling and animation)

Justification: The roles were established like this as, considering he was doing documentation anyway, making Alexander design the concept art seems sensible enough. He was given only basic modeling duties because of his inexperience with animation.

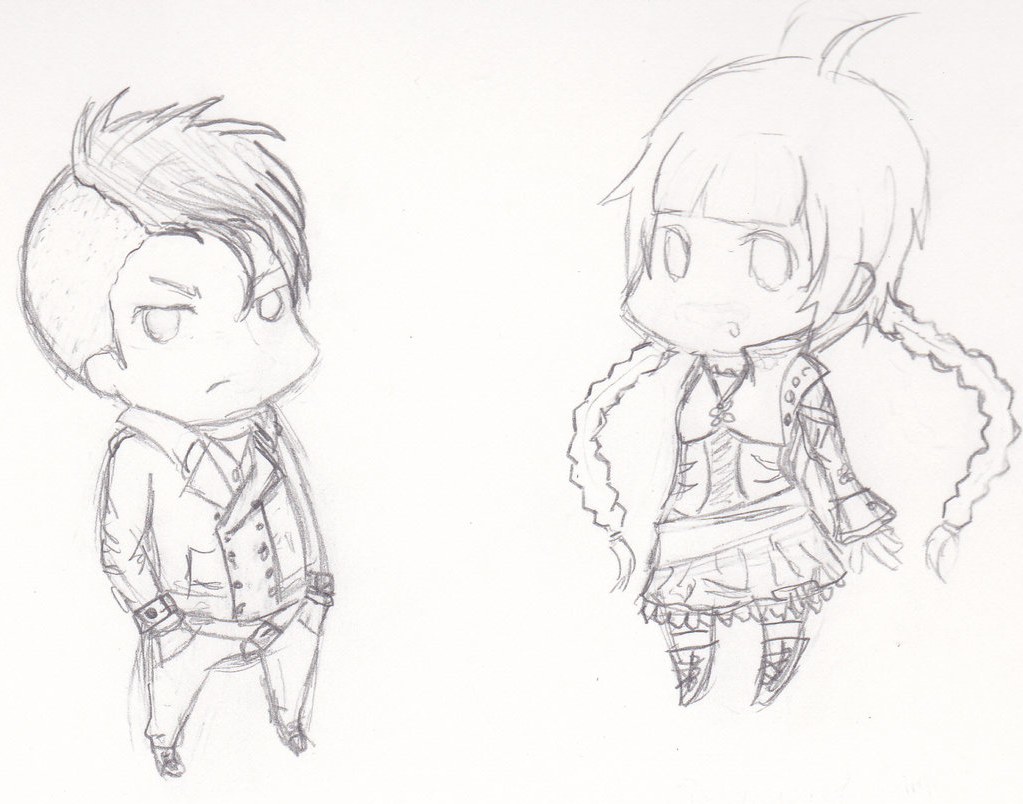
Kyle was left to also model and design some characters to tie into his programming work. This was deemed as reasonable because he seems more confident with animation and he’d have to do the rigging to match up with how he wants things to move in game anyway.

# Graphical Information:

To add: Concept art, models, information about the aesthetics, storyboards, asset list.

Graphical Style: The game’s overall art style was eventually decided to be a fairly ‘chibi’-like style. The reason this was decided upon was mostly because of it’s a fairly lighthearted non-serious style rather which suits a power-up intensive 3D platformer far more than a more realistic style such as the one seen in Naughty Dogs Uncharted franchise.

Another factor towards choosing this style was for modelling the player character and other enemies. Simply put, whilst the team might be able to model a realistic human in Maya it would take too much time which could negatively affect the game. On the other hand modelling a chibi character model takes far less time and is thus a more efficient and effective way to make models for the game.





The above images are the basic concept sketches of possible main character designs.

The models themselves will be modelled and textured in using Autodesk Maya before importing them into Unity for implementation.

# Level Information:

To add: information about the game levels along with concept art.

# Gameplay:

To add: Information about the game, how it flows and other information such as AI.

Powers: As a major part of our game we went over several different ideas for which power-ups or powers the player should get. We ultimately only included a few of our original ideas with the discarded ideas listed below.

Discarded Powers:

The powers that we did choose to include are listed below.

Used Powers:

# Story:

To add: Information about the game’s story

Initial Story Ideas: While we never intended for the story to be the major part of the game (not being very skilled writers) we threw around a couple of different ideas before settling on the final model only discarding them when they didn’t fit with some aspect of our game.

Biological Experimentation: This idea was initially put forward by Kyle as one of the first suggestions for the game’s story. It was focused around the idea of the protagonist, Nate Breckner, being involved in some human experimentation gone wrong. This was to be the source of the player’s ability to gain other powers. The player was to discover that the experiment had been intentionally sabotaged as part of some government conspiracy with the player discovering more about it as the game progressed.

While this did provide a justified reason for the protagonist’s powers it didn’t really fit with the chosen art style nor with the gameplay type either as ‘light hearted fun 3D platformer’ and ‘Government conspiracy’ don’t really match. On top of this the plot sounded like something you’d see in a spy movie rather than a 3D platformer. Ultimately, not matching up with the game we were trying to make was the reason we didn’t go with this plot.

Final Story Idea: In the end what we went with was something a bit simpler and child friendly. The game is based around the protagonist (…) waking up in a strange land where enemies, upon being defeated, grant you powers. The ‘story’ of the game as it were would basically be trying to use the powers obtained to advance in and escape the world.

It’s pretty much an Excuse Plot in every sense of the word and outside of the manual would likely not be mentioned. The reason this was chosen was because of the key fact that neither me nor Kyle is a writer and thus, rather than try to write something deep and botch it up spectacularly we chose to go with something that was simple, effective and could be ignored instead figuring that this would allow greater emphasis to be placed on the gameplay.

# Control Scheme:

To add: Information about the games control scheme.

# Testing and QA:

To add: Information about how the game will be tested along with technical test plans and filled out tests.

While the game will be tested constantly throughout its production with a variety offsets relating to its implementation the final block testing of everything that the game should be capable of can be carried out in one or more methods. Specifically there are two separate testing methods we considered for this Black box testing and white box testing.

White Box Testing

A list of intended tests for the game is listed below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Test | How the test is carried out | Expected Result | Actual Result | Comments |
| 001 | Player Movement: Left |  | The player moves to the left by a specified amount. |  |  |
| 002 | Player Movement: Right |  | The player moves to the right by a specified amount. |  |  |
| 003 | Player Movement: Forward |  | The player moves forward by a specified amount. |  |  |
| 004 | Player Movement: Back |  | The player moves backwards by a specified amount. |  |  |
| 005 | Player Jump |  | The player jumps up by a specific value. |  |  |
| 006 | Gravity |  | The player falls for a bit before hitting the ground. |  |  |
| 007 | Camera Tracking |  | The camera follows the player |  |  |
| 008 | Player Spawn |  |  |  |  |
| 009 |  |  |  |  |  |
| 010 |  |  |  |  |  |
| 011 |  |  |  |  |  |
| 012 |  |  |  |  |  |

# Target Audience:

Considering that the game’s art style, genre and mechanics seem like something you’d find in a lower rated cheerful game we decided it’d be best to have our target demographic be children aged 7 and upwards. This is because we feel that the specific style, lack of plot and more whimsical nature of the game would appeal to them most.

As a result of the factors listed above we’d be aiming for a PEGI 7 rating.

Outside of the primary demographic listed above though there would be a periphery demographic as well. To be specific we’d be aiming at those who enjoyed the other 3D platformers such as Spyro the Dragon, Crash Bandicoot and any of the 3D Mario games. Based on the age of those franchises (Spyro: started in 1988, Crash Bandicoot: started in 1996, Mario: first 3D platformer released in 1996) this audience will either consist of adults or teenagers who learned about them through later games in the franchise. Hopefully the platforming and similar game style will be enough to attract such players.

# Scope:

To add: Information about the games scope

# UI:

To Add: Information about the UI and any maps in the game.

# Milestones:

To add: A list of dates that tasks are to be completed by.

* First Project hand in: 14/11/2017
* Final project hand in 1/12/2017

# Requirements:

To add: A list of functional and non-functional requirements for the game.

Functional Requirements:

Non-Functional Requirements:

# Design Approach:

To add: Mention of the Design approach we’re using, why we picked it and what the alternatives were.

Before selecting the design approach we settled upon we examined several others though we discarded them for various reasons. Listed below are some of them.

Waterfall: the waterfall approach is a linear design approach that has the team go from one step from another in sequence without going back to any previous steps. The essential principles behind this were:

* Dividing the project into individual phases approached in sequence.
* Placing a major emphasis on points such as schedules, dates, and budgets and implementing the entire game at one time.
* Managing and extremely tight control over the project throughout its life using a large amount of documentation and formal reviews performed at the end of each stage.

The approaches strengths were as follows:

* It’s considered useful for less experience teams as it follows such a rigid straightforward schedule it’s hard to get lost.
* The extremely orderly manner in which the individual steps are performed, when combined with the frequent reviews can result in the developed software being of high quality.
* It’s easy to measure how the systems development is coming along.
* It also saves resources as everybody is following a strict schedule and knows what they’re doing.

As for the weaknesses it has:

* It is utterly inflexible and not particularly fast because of the slow step by step process and strict limitations on what could be done.
* The project rarely, if ever, moves back meaning that mistakes or changes can’t be easily fixed.
* It essentially needs everyone to know what they’re doing right at the start despite the fact that the client might not know what that is.
* It’s not uncommon to find mistakes, inconsistencies and unexpected problems when following this approach.
* The project can’t really be tested until its entirely complete meaning errors and problems (such as it not working as desired) may show up and to be pretty difficult to fix. It’s also difficult to adjust to life-cycle changes and thus advantageous changes may not be applied as a result.
* Crating the documents and keeping them up to date is taxing and time consuming and such in depth documents may be difficult for clients or even for certain team members to read or appreciate.

In the end while this makes the approach useful for tasks that have clear end goals and unchanging requirements for our task, a game with requirements that’ll likely change and that has no definite end goal it’s pretty useless. On top of this we’ll often need to go back to previous steps after completing some work such as updating the documentation with concept art or the UI section once those elements are designed and implemented so a linear design approach isn’t really applicable to our task.

In the end the design choice that we *did* choose was the one listed below.

# Resources:

**Software:** In order to properly make the game a collection of various programs or software will be required either to implement the game or to produce some of the assets for the game. The programs we suspect we’ll need will be listed below.

|  |  |
| --- | --- |
| Software | Price |
| Unity | Free |
| Microsoft Word | Already owned |
| Microsoft Powerpoint | Already owned |
| Autodesk Maya 2018 | Available for free for students |
| Paint.net | Free |
|  |  |
|  |  |

As can be seen above all of the software listed is either free for everyone, free for students or are already possessed by one or both team members. This means that none of the team members need to waste money purchasing the products to work on the task.

**Hardware:** Hardware is also required in order to produce the game as several pieces are either mandatory for getting the computer to function (such as a keyboard or monitor) or useful for producing part of the game or documentation.

|  |  |
| --- | --- |
| Hardware | Price |
| Monitor | Owned already |
| Laptop | Owned already |
| Mouse | Owned already |
| Keyboard | Owned already |
| Graphics Tablet | Owned by 1 member already (no longer needed) |
| USB Memory Stick | Already Owned |
| Scanner | Already Owned |
|  |  |
|  |  |

As can be seen above all of the necessary hardware is already possessed by at least one of the team members thus reducing any required costs to 0 barring accidents.

Extra Notes: its worth mentioning that most of the hardware and software listed above *is* available inside the university campus thus allowing for work to be performed there. It is somewhat more limited though as some classrooms or software may not be available if a class is in session at the time.

# Work Done:

Week 1:

* As the induction week nothing was done

Week 2:

* The task and teams were established.
* Contact information was shared within the group.

Week 3:

* Kyle completed a very basic base to use for the game.
* Alexander created a github location for the project.
* Alexander created a base for the GDD.
* The group as a whole decided we’d be working on a 3D platformer.

Week 4:

* Jordan left the group to transfer to another course. This resulted in team roles being reallocated.
* We had a talk about how to re-allocate team roles along with what kind of style we wanted the game to follow.
* Kyle improved the basic movement of the main character in the base made for the game.
* Alexander filled out the team roles section, some of the resources section, part of the milestones section and part of the design approach section within the Game Design Document.
* Alexander uploaded an unfinished GDD to github.

Week 5:

* Alexander worked on some concept art for some of the game’s enemies.
* Kyle designed some concept art for the protagonist.
* Alexander worked on the design document some more filling part of the story, art, and game outline.
* Kyle worked on the games implementation some more.
* Kyle completed the main character’s model.
* Alexander uploaded an updated design document to github.

Week 6:

* Alexander added information to the GDD namely to the Target audience, testing and QA, story and Resources sections.

# Information Sources Used:

**Websites Accessed:**

New York Film Academy (2014) [Website] <https://www.nyfa.edu/student-resources/how-to-write-a-game-design-document/> [Accessed 26/09/2017]

Honeysanime.com (2017) [Website] <https://honeysanime.com/what-is-chibi-definition-meaning/> [Accessed 03/10/2017]

Wikipedia (2017) [Website] <https://en.wikipedia.org/wiki/Uncharted> [Accessed 03/10/2017]

Kotaku (2017) [Website] <https://kotaku.com/the-art-of-uncharted-4-1777213424> [Accessed 03/10/2017]

Wikipedia (2017) [Website] <https://en.wikipedia.org/wiki/Mega_Man_(video_game)> [Accessed 03/10/2017]

Pegi (2017) [Website] <http://www.pegi.info/en/index/> [Accessed 10/10/2017]

Wikipedia (2017) [Website] <https://en.wikipedia.org/wiki/Crash_Bandicoot> [Accessed 10/10/2017}

Wikipedia (2017) [Website] <https://en.wikipedia.org/wiki/Super_Mario_64> [Accessed 10/10/2017}

Wikipedia (2017) [Website] <https://en.wikipedia.org/wiki/Spyro_(series)> [Accessed 10/10/2017}

Tv Tropes (2017) [Website] <http://tvtropes.org/pmwiki/pmwiki.php/Main/ExcusePlot> [Accessed 10/10/2017]

**Documents Used:**

**Books Accessed:**

**Appendixes:**