Proposal Documentation

# Design

The game will be a top down 2D shooter, where the player will control a ship and have to fight off waves of enemies. The game will run on Android mobile devices and be developed using unity. The player will control the ship, using a touch interface, which allows precise control without relying on sensors that some devices might not have. There will be a variety of enemies with differing attacks and movement, as well as resources to upgrade the ship and powerups. Powerups are activated instantly and give temporary benefits such as extra health or greater fire rate, and additionally the ship can be upgraded between levels. The game will also feature an online high score system allowing players to view each others scores and will use advertisements in menus and between levels as a form of revenue.

## Overview

## (Content by the team)

Our game will be a 2D, Top-Down, Bullet-Hell Shooter, where the player is to traverse through each point of a level, dealing with various types of enemies, in order to progress to the end-of-level boss. After defeating the boss, the player can move on to the next level. In the game, there will be pick ups: for health, in-game currency, and to double damage for a limited time. The in-game currency can buy upgrades such as increased health, more damage, greater speed and so on. The game will be developed with Unity, for Android phones; most notably the Google Nexus 5.

# Progress Tracking

### (Content by James Moran)

We will use a combination of GitHub, Trello and Google Drive for our project, in order to track our progress, as well as to make sure we are meeting deadlines.

We will use GitHub as this allows us to see what changes have been made to all of the Unity related files in our project, at any time (as to who changed which particular file in the project, as well as the changes they have made to it, if it is not a binary file (e.g. a source file, in its plain text form, including which lines have been added/removed/modified)).

The task of the management of the GitHub Repository, has been delegated to James Moran.

We will also use Trello, as this will allow us to keep track of the project’s schedule, in a collective manner, keeping a record of everyone’s deadlines. In addition, once can also provide links to other material, built-in Google Drive synchronisation, as well as features for adding labels, due-dates and members, to a particular task card (as well other utility features of Trello).

Finally, we will use Google Drive, as this will allow us to store documentation files of our project. This will allow us to see who has edited which document in the folder and how they have edited it (if the file is not a binary file).

### (Jack Evans)

Using tracking tools will help to manage the progress of development. This will be done by following user stories/sprints through an iterative model, allowing us to keep track of development progress as well as adapt to any problems or changes that need to be made. To help with this we will also be making use of a work breakdown structure and activity sequencing which will give a visual overview of the project as well as helping to avoid having team members held up waiting for others.

As we will be using GitHub for source control we will also need to have a plan for how to make use of it and avoid conflicts. To this end we will keep separate branches for each team member or task and only merge into the master branch when tasks are completed.

# Schedules and Stories

## (Jack Evans)

## Deadlines

* Prototype 7th November - This should show the core gameplay mechanics only. Player movement, basic attack, simple enemy, simple enemy attack.
* Alpha 21st November - All levels should be completed allowing the game to be played from start to finish. The game should have a selection of enemies and power ups that are revealed as the player progresses. The high score database should also be set up.
* Beta 5th December - The game should be feature complete however it may have a lot of bugs.
* Final Hand In 5th January - The game should be bug free for hand over.

## Stories

### Prototype

* Player Ship
* Player Controls
* Enemy Ship
* Attacks

### Alpha +

* Main Menu
* Pause Menu
* High Scores
* More Enemies
* More Attacks
* Bosses
* Levels
* Power Ups
* Upgrade Ship
* Adds

# Asset List

### (Content by Anthony Boys)

The asset list is in correlation to the “enemy type” documentation, and .svg / .svgz files:

Player ship

Projectiles: Bullet, Energy, Rocket, Missile, Laser, Mine

Ship colours: Green, Yellow, Orange, Red, Blue, Purple, Magenta

Ship types: Base, Cannon, Disc, Egg, Gem, Jet, Tank

Pick ups: +5 cash\*, +20 cash\*, +100 cash\*, +50% hp\*\*, +100% hp\*\*, double damage, nuke, 2.5x speed\*\*\*, enemies freeze, invincibility, triple cash, 0.3x reload time, double points

\*”cash” is the in-game currency, represented by the generic currency sign, “¤”

\*\*”hp” represents “health points”

\*\*\*”speed” refers to the movement speed of the ship

# Workload

Each team member’s workload has been delegated as follows:

## Jack Evans

### (Content by Jack Evans)

## 

## James Moran

### (Content by James Moran)

This is James Moran’s Work Schedule, with each Task, the number of Hours for that task and the development phase that task is to take place within:

|  |  |  |
| --- | --- | --- |
| **Phase** | **Task** | **Hours** |
|  | Documentation | 2 |
|  | Workload | 1 |
|  | Risks | 1 |
|  | Set up Github | 1 |
| 1st Assignment | Attending meetings | 2 |
|  | Github Management | 3 |
|  | Health Powerup | 1 |
|  | Level 0 | 1 |
|  | Vertical slice | 5 |
| Prototype | Attending meetings | 2 |
|  | Github Managment | 3 |
|  | Adds | 3 |
|  | Power Ups | 2 |
|  | Attending meetings | 2 |
|  | Levels | 1 |
| Alpha | Code Repair | 1 |
|  | Github Management | 3 |
|  | Power Ups | 2 |
|  | Ship Upgrades | 2 |
|  | Attending meetings | 2 |
| Beta | Code Repair | 1 |
|  | Github Management | 3 |
|  | Attending meetings | 2 |
| Finished Game | Code Repair | 1 |
| Total hours |  | 47 |

## Anthony Boys

### (Content by Anthony Boys)

Here is my workload schedule, broken into tasks, and their hours:

|  |  |  |
| --- | --- | --- |
| **Phase:** | **Task:** | **Hours:** |
|  | Documentation | 2 |
|  | Workload | 1 |
|  | Risks | 1 |
|  | Enemy design | 3 |
|  | Attending meetings | 2 |
|  | Meeting minutes | 2 |
| 1st Assignment | Design docs | 2 |
|  | Art | 3 |
|  | Wireframes | 3 |
|  | Trello maintenance | 1 |
|  | Attending meetings | 2 |
|  | Meeting minutes | 2 |
|  | Design docs | 2 |
|  | Testing | 3 |
| Prototype | Test logging | 1 |
|  | Fix suggesting | 1 |
|  | Trello maintenance | 1 |
|  | Attending meetings | 2 |
|  | Meeting minutes | 2 |
|  | Testing | 3 |
| Alpha | Test logging | 1 |
|  | Fix suggesting | 1 |
|  | Trello maintenance | 1 |
|  | Attending meetings | 2 |
|  | Meeting minutes | 2 |
|  | Testing | 3 |
| Beta | Test logging | 1 |
|  | Fix suggesting | 1 |
|  | Trello maintenance | 1 |
|  | Attending meetings | 2 |
|  | Meeting minutes | 2 |
|  | Testing | 3 |
| Finished Game | Test logging | 1 |
| Total hours | Total hours | 60 |

# Risk

The risk for each team member's workload is as follows:

## Jack Evans

### (Content by Jack Evans)

Some of this content will be used in the full Risk Analysis of the project (below this section):

* Computer failure
* Software made unavailable (I.E Unity)
* Illness

## James Moran

## (Content by James Moran)

The Risk Identification table is shown below. This is for the first phase of the project (before the hand in of AE1, so this risk analysis section is subject to change for later milestones in the project), (Fig. 1.):

|  |  |  |
| --- | --- | --- |
|  | Technical | Non-Technical |
| Event Driven Risks | * Computer crashes before/whilst work is getting saved to disk * Programs used in the project crash whilst a team-member is using them * Software is removed from a particular workstation (or multiple workstations) by the IT support technicians | * Team-member becomes ill * (Query Nick about some ideas for risks, perhaps North Korea could be put here as one…) |
| Evolving Risks | * Printer cartridge within the printer in the room where the team is working, is not replaced | * Team-member is absent from project meetings * A feature due for implementation, is not implemented when it was scheduled for implementation |

Fig. 1. The Risk Identification table (Above)

The Risk Analysis Matrix is shown below, also for the first phase of the project (Fig. 2.):

|  |  |  |  |
| --- | --- | --- | --- |
| **Severity** | **Likelihood of Risk** | **Likelihood of Risk** | **Likelihood of Risk** |
|  | **1** | **2** | **3** |
| **1** | **-** | **-** | **Programs used in the project crash whilst a team-member is using them** |
| **2** | **Printer cartridge within the printer in the room where the team is working, is not replaced** | **Computer crashes before/whilst work is getting saved to disk** | **Team-member becomes ill** |
| **3** | **Software is removed from a particular workstation (or multiple workstations) by the IT support technicians** | **Team-member is absent from project meetings** | **-** |
| **3** | **-** | **A feature due for implementation, is not implemented when it was scheduled for implementation** | **-** |

Fig. 2. The Risk Analysis Matrix

The Risk Impact is then noted for each risk in the table below (Fig. 3.):

|  |  |
| --- | --- |
| Risk Description | Risk Impact |
| Computer crashes before/whilst work is getting saved to disk | 4 |
| Programs used in the project crash whilst a team-member is using them | 3 |
| Printer cartridge within the printer in the room where the team is working, is not replaced | 2 |
| Software is removed from a particular workstation (or multiple workstations) by the IT support technicians | 3 |
| Team-member becomes ill | 6 |
| A feature due for implementation, is not implemented when it was scheduled for implementation | 6 |

Fig. 3. The Risk Impact for each Risk (above).

## Anthony Boys

### (Content by Anthony Boys)

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk description:** | **Impact** | **Likelihood** | **Mitigation** |
| Computer crashes during work, and some work is lost | 2 | 2 | Ensure work is saved and backed up |
| Backups lost or corrupted | 3 | 2 | Use multiple and online backups |
| Software updates and therefore becomes incompatible | 2 | 2 | Use legacy software |
| Required software becomes completely unavailable | 4 | 1 | Ensure that other types of software can be used instead, and that tasks are not too dependant |
| An online service, such as Trello, GitHub or Google Drive becomes unavailable | 4 | 1 | Ensure alternatives can be found and used, and that work is saved outside these areas |
| The internet becomes unavailable | 3 | 2 | Find somewhere that has internet access |
| A team member falls ill | 2 | 3 | Delegate tasks |
| A team member does not communicate properly, or complete tasks properly, to the point where it harms progress | 2 | 2 | Delegate tasks, and try to best resolve the issue. |
| A team member, for whatever reason, is no longer a member of the team | 3 | 1 | Delegate tasks, and try to replace team member if necessary |
| Client changes their requirements | 5 | 1 | Ensure that the project is updated accordingly |
| Product is noticeably behind schedule | 4 | 2 | Ensure relevant tasks are completed on time, perhaps by working extra hours |
| Product is inadequate, or doesn’t meet requirements | 5 | 2 | Ensure all issues are resolved, maintain communication with client, make use of feedback, and that sufficient testing is carried out after each major change |
| Project gets hacked | 5 | 1 | Ensure passwords are strong |
| University becomes inaccessible | 3 | 1 | All work can be completed outside of the university, providing the relevant software is installed |