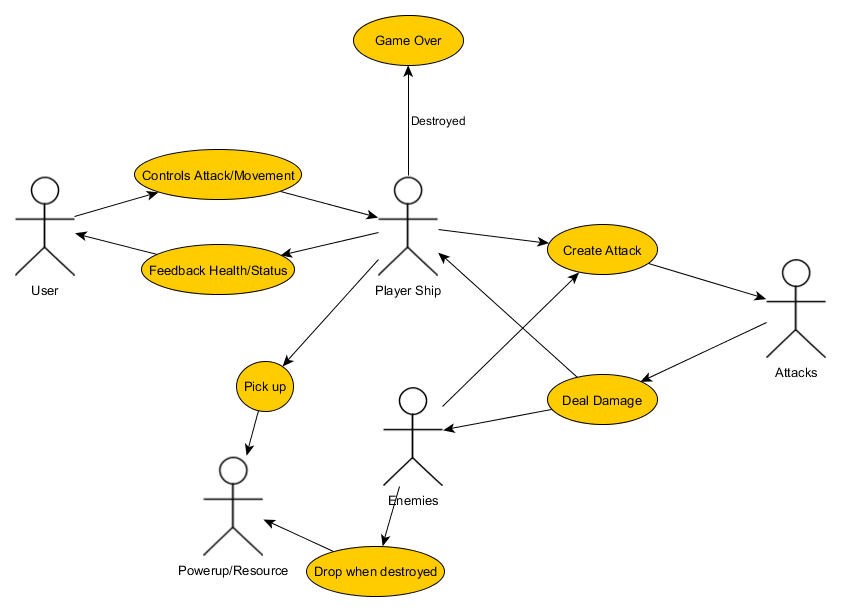
Proposal Documentation

Design Overview

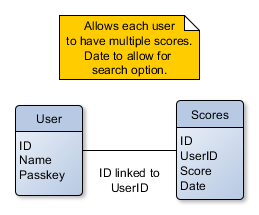
(Content by the team)

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 will 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, along with power-ups. Power-ups are activated instantly and give temporary benefits such as extra health or greater fire rate. Additionally, the ship can be upgraded between levels. The game will also feature an online high score system allowing players to view each other’s scores and will use advertisements in menus and between levels as a method of generating revenue. See Appendix A for more detailed game overview.



## Database

A database will be used for tracking high scores this is how the database will be laid out.



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 this shows 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)). We have decided on GitHub, as we will only have one repository and have been shown/found-out how to use GitHub, via the University (For additional reasons, please see the referenced article (UpGuard, 2017)).

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, one can also provide links to other material, use built-in Google Drive synchronisation, as well as features for adding labels, due-dates and members, to a particular task card (as well as other utility features of Trello). In comparison to other free project management solutions, Trello is considered the best overall, for its sleek kanban-style interface, allowing for premium features to be bought, but are not necessary to operation. Trello offers time management in a relatively simple solution (in terms of UI), making it suit our projects goals and objectives. (Gavin Graham, 2017)

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). We have decided upon using Google Drive, as it offers the greatest free storage space (15GB), syncing to the desktop application is not an issue for our team (exclusively used online, for now), along with a suite of web-based applications, exclusive to Google Drive, that can also be used offline (after changing certain settings).

(Content by Jack Evans)

Using tracking tools will help to manage the process and 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 address 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

Deadlines

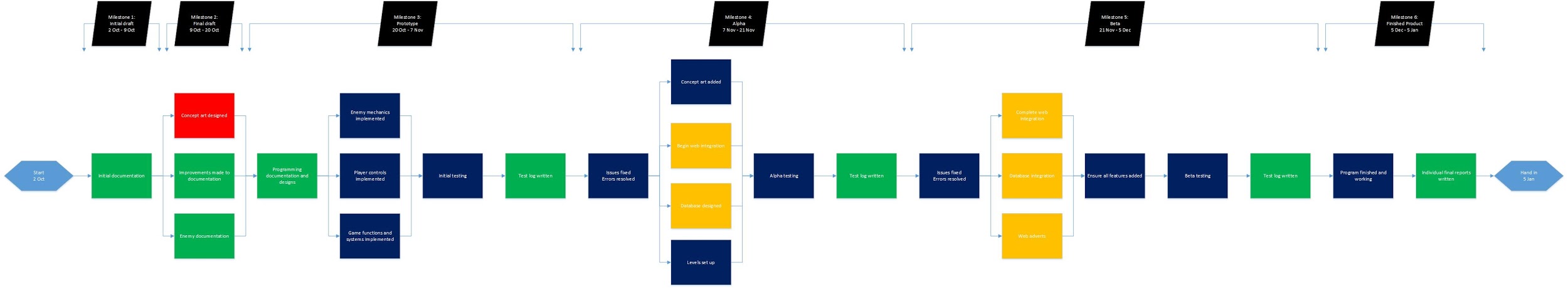
(Content by Jack Evans)

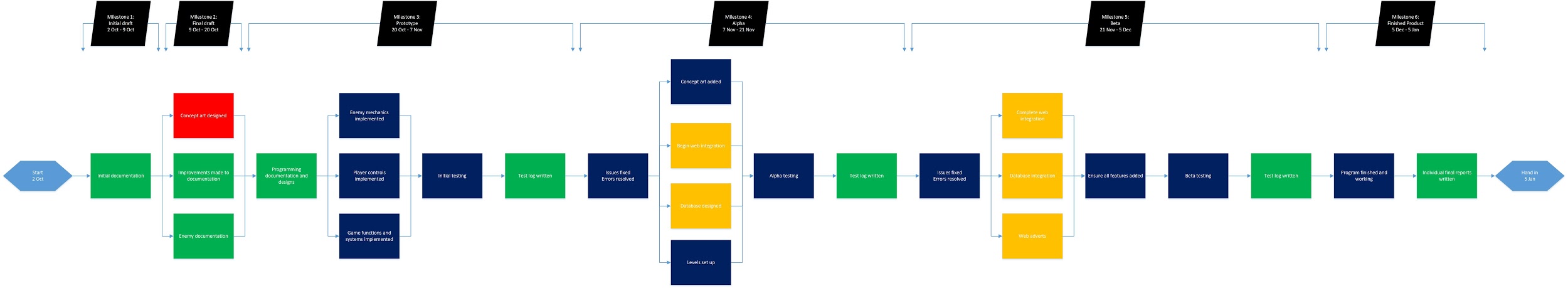
* Prototype 7th November - This should show the core gameplay mechanics only. Player movement, basic attack, simple enemy, simple enemy attack.
* Alpha 21st November - Some levels should be completed, allowing the game to be played properly. The game should have a selection of enemies and power ups that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

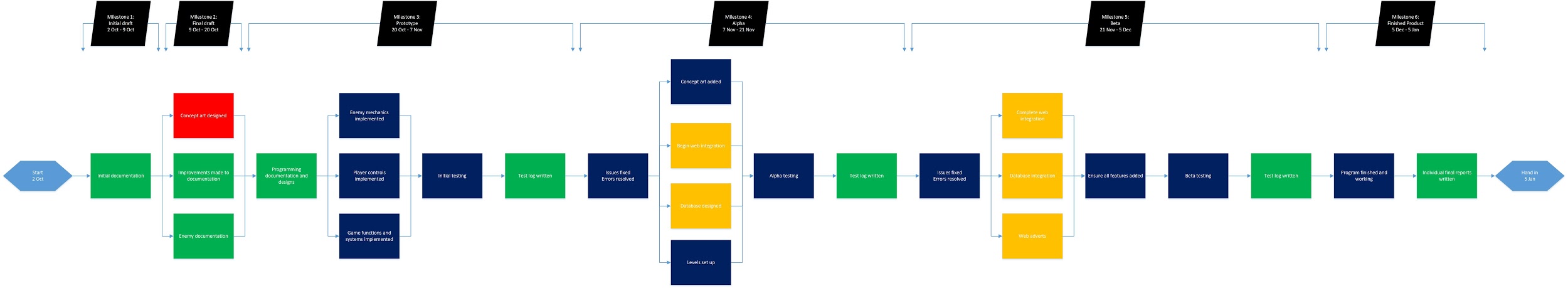
Schedule

(Content by Anthony Boys)

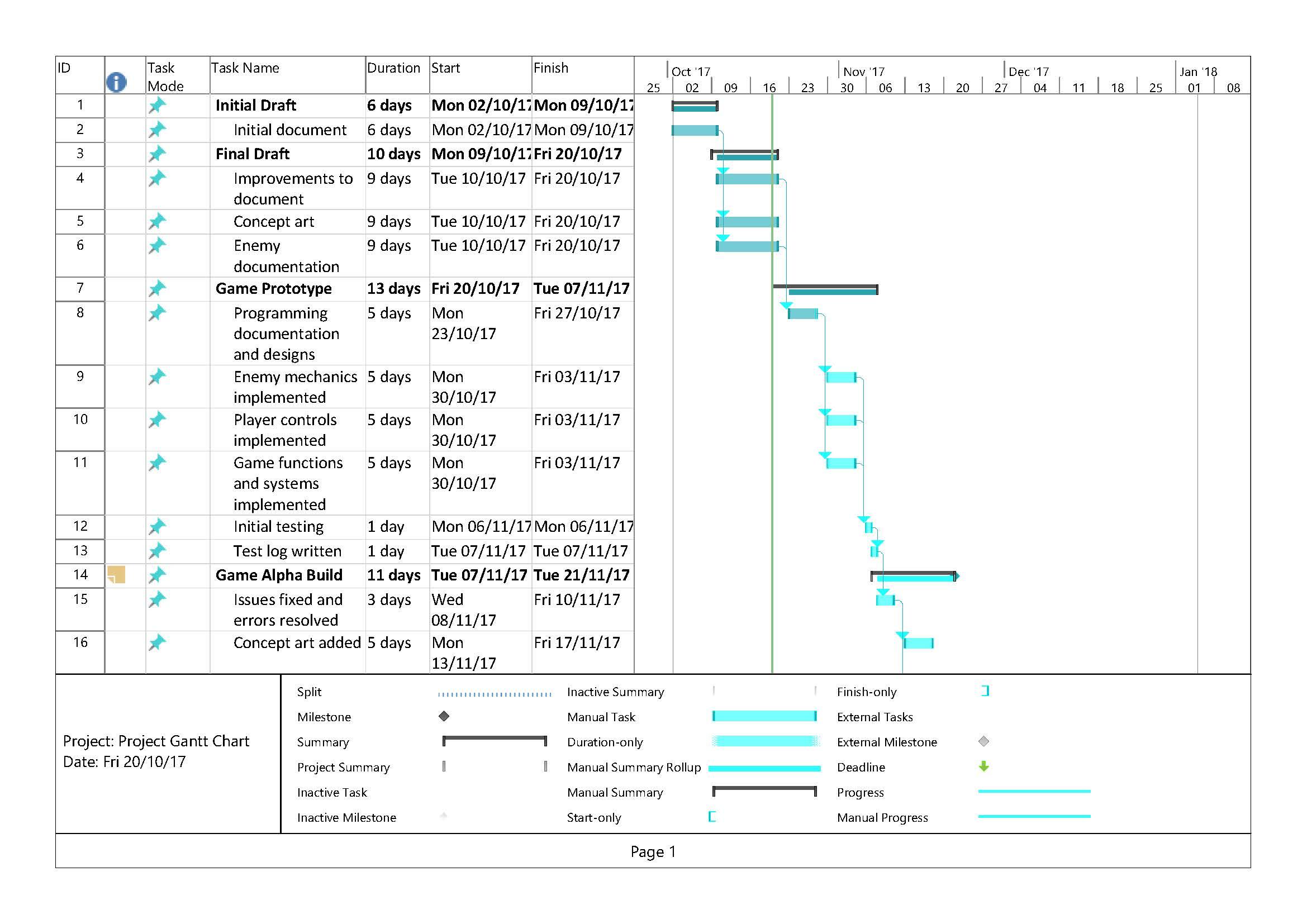
Here is the Pert chart, which documents the expected schedule of the project, and which tasks must be completed in which order, along with milestones at the top of the chart, that designate tasks in to deadlines. We have retrieved these processes from the requirements and their respective components

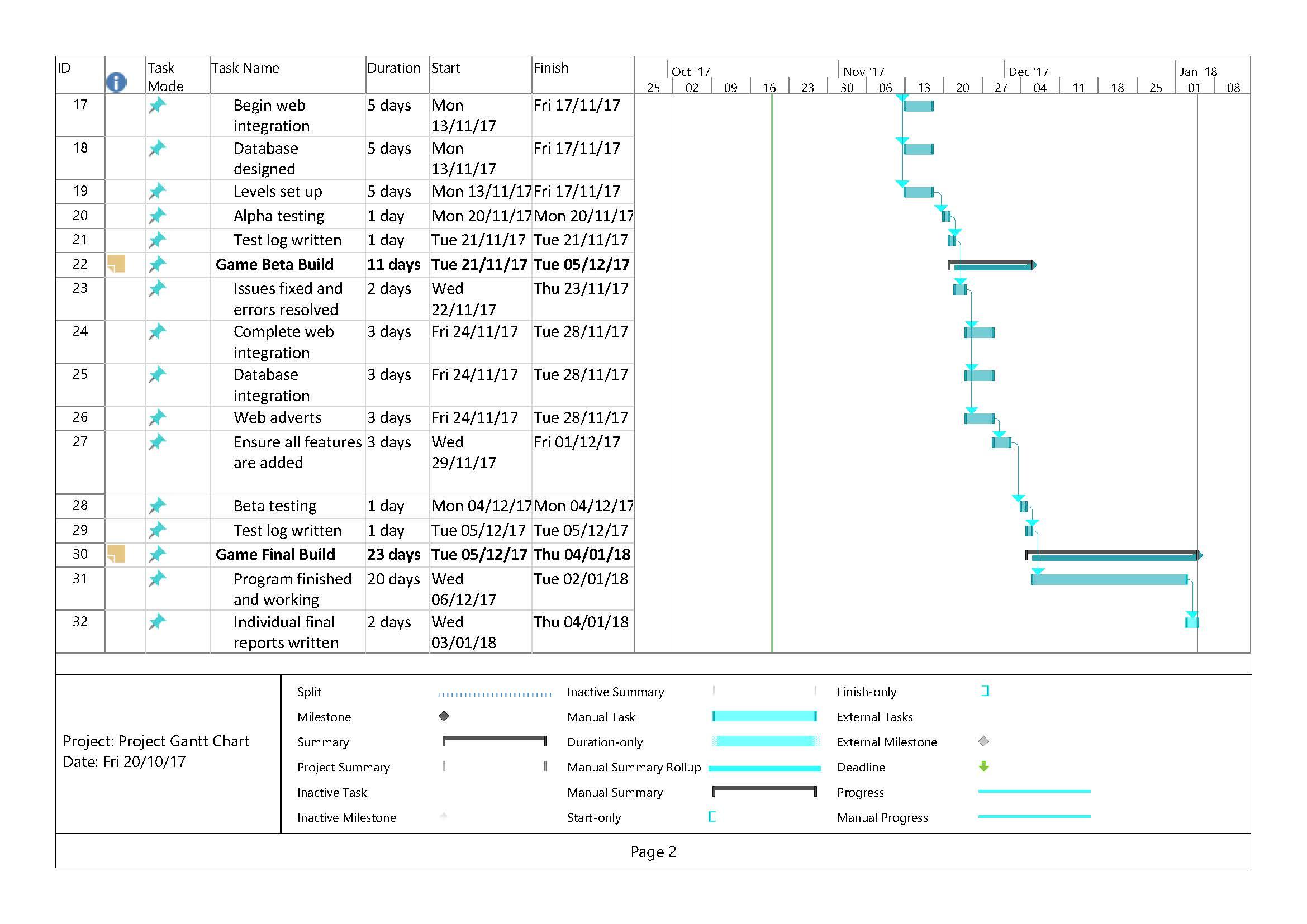






Here is the Gantt chart for the project, which contains all the processes mentioned in the Pert chart.





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)

|  |  |  |
| --- | --- | --- |
| **Phase** | **Task** | **Hours** |
| 1st Assignment | Documentation | 2 |
| Workload | 1 |
| Code Design | 3 |
| Set Up Trello | 1 |
| Attending meetings | 2 |
| Prototype | Player Ship | 3 |
| Enemy | 1 |
| Attack | 1 |
| Vertical Slice | 5 |
| Attending Meetings | 2 |
| Alpha | Main Menu | 1 |
| Pause Menu | 2 |
| Enemies | 2 |
| Attacks | 2 |
| Attending Meetings | 2 |
| Levels | 1 |
| Code Repair | 1 |
| Beta | Code Repair | 1 |
| All Enemies | 2 |
| All Attacks | 2 |
| Attending Meetings | 2 |
| High Scores | 3 |
| Finished Game | Code Repair | 4 |
| Attending Meetings | 2 |
| Total hours |  | 50 |

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** |
| 1st Assignment | Documentation | 2 |
| Workload | 1 |
| Risks | 1 |
| Set up GitHub | 1 |
| Attending meetings | 2 |
| Prototype | GitHub Management | 3 |
| Health Powerup | 1 |
| Level 0 | 1 |
| Vertical slice | 5 |
| Attending meetings | 2 |
| Alpha | GitHub Management | 3 |
| Adds | 3 |
| Power Ups | 2 |
| Attending meetings | 2 |
| Levels | 1 |
| Code Repair | 1 |
| Beta | GitHub Management | 3 |
| Power Ups | 2 |
| Ship Upgrades | 2 |
| Attending meetings | 2 |
| Code Repair | 1 |
| Finished Game | GitHub Management | 3 |
| Attending meetings | 2 |
| 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)

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk description:** | **Impact** | **Likelihood** | **Mitigation** |
| Personal Computer Failure - Reduces the time I have access to a suitable computer. The only other computers I have access to are the University’s computers. | 4 | 2 | Have enough leeway in project schedule to either get the computer repaired or replaced or make do with the University's computers. |
| Lose access to database hosting service. | 3 | 3 | Design the database such that it can be loaded on a personal computer if needed, for example a Raspberry Pi. |
| Illness/Injury - The impact of any illness or injury is dependent on the severity. | 1-5 | 1 | Ensure the project and project schedule are adaptable enough to allow the other team members to finish the game. |

James Moran

(Content by James Moran)

The Risk Identification table is shown below.

|  |  |  |
| --- | --- | --- |
|  | Technical | Non-Technical |
| Event Driven Risks | * Computer crashes before/whilst work is getting saved to disk * Programs used in the project crash whilst in use * Software is removed from a particular workstation (or multiple workstations) by the IT support technicians | * Team-member becomes ill |
| Evolving Risks | * The printer cartridge within the printer in the room where the team is working, is not replaced * Bugs in the project are not accounted for | * Team-member is absent from project meetings * A feature due for implementation, is not implemented when it was scheduled for implementation * A team-member remains in an ever worsening state of illness |

The Risk Analysis Matrix is shown below, also for the first phase of the project:

|  |  |  |  |
| --- | --- | --- | --- |
| **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**    **Bugs in the project are not accounted for** | **Team-member becomes ill** |
| **3** | **Software is removed from a particular workstation (or multiple workstations) by the IT support technicians**    **A team-member remains in an ever worsening state of illness** | **Team-member is absent from project meetings**    **A feature due for implementation, is not implemented when it was scheduled for implementation** | **-** |

The Risk Impact is then noted for each risk in the table below:

|  |  |
| --- | --- |
| 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 |
| Bugs in the project are not accounted for | 4 |
| A team-member remains in an ever worsening state of illness | 3 |

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 |

Appendix A: Game Design

Design

Overview

(Content by Jack Evans & Anthony Boys)

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 power-ups. 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 the Google Nexus 5.

Player

(Content by Jack Evans & Anthony Boys)

The player will be able to move around the screen whilst shooting at enemies, allowing them to aim bullets at the enemies and avoid enemy attacks. The player has their own ship, with a set amount of health and has upgradable statistics. The player will also be able to collect various items from destroyed enemies, which they will be able to use to upgrade their ship, or will grant power-ups that provide a temporary upgrade or special abilities.

Enemies & Boss

(Content by Anthony Boys)

The enemies are as follows:

There are seven types of enemies: base (spawns enemies), cannon (static gun), disc (flying saucer), gem (heavy armour), tank (high damage), jet (fast striker), and egg (“hatches” by transforming into an attack when destroyed).

There are seven classes of enemies; each class is an improvement of the class before.

The standard classes are: green, yellow, orange and red. The boss classes, which are a significant improvement of the standard classes, are: blue, purple and magenta.

Revenue

(Content by Jack Evans & Anthony Boys)

The game will make use of banner advertisements, in menus and splash screens between levels, as a means of acquiring revenue, allowing the game to be free to play. As an alternative, there will be an option to purchase the premium in-game currency, which allows the player to buy premium powerups, and removes advertisements for a set amount of time.

Online Integration

(Content by Jack Evans)

The game will make use of an online database for sharing player’s high scores. The game will display the high scores from each level, and potentially have a search system that allows players to see the high scores of their friends.

Menus

(Content by Jack Evans)

The game will have two menus; the main menu and the in-game menu. The main menu is the first thing a player will see after the game has loaded and will allow the player to start the game, quit, or view high scores. The in-game menu, or pause menu, will be accessible in game, and doing so will pause the game. From the in-game menu, the player will be able to resume the game, quit to the main menu, or quit the game outright.

# Appendix B: References

1. UpGuard, ©2017 UPGUARD, INC. *Github vs Bitbucket* [viewed on the 20/10/2017]. Available from: <https://www.upguard.com/articles/github-vs-bitbucket>
2. Gavin Graham, Copyright © 2017 · FitSmallBusiness. All Right Reserved. *Best Free Project Mangement Software 2017: Trello vs Asana and the Rest* [viewed on the 20/10/2017]. Available from: <https://fitsmallbusiness.com/best-free-project-management-software/>
3. Joseph Gildred, © 2007-2017 Cloudwards.net. *Best of The Big Three: Dropbox vs Google Drive vs Onedrive* [viewed on the 20/10/2017]. Available from: <https://www.cloudwards.net/dropbox-vs-google-drive-vs-onedrive/>