James Moran

This document is the Post-Mortem Report of the game put together for CGP605: Mobile Applications.

James Moran, Q12494305, CGP605, AE2

Post-Mortem

Contents

**No table of contents entries found.**

# Project Tracking

For our project, we put together a group-workload sheet, with the delegated tasks for each member of the group and the hours that they were expected to take on each of these tasks (see Fig. 1 of Appendix A: Figures).

I kept my progress on track for the project, by logging my progress in a table, that is editable online. This table is similar in layout to the table that I put together for group-work in Engineering Software Systems, last year (Fig. 2 of Appendix A: Figures).

So, for each phase of the project (putting together the Project Proposal, Prototype, Alpha, Beta and the Finished Game), each group member is expected to complete certain tasks as per the phase of the project (e.g. for the Project Proposal Phase, I was expected to consider the Risks of the project, provide documentation on the tasks I had further down the line and how I would manage the Workload). I was given the project-long responsibility of setting-up a GitHub repository and managing it (resolve conflicts between branches, properly merging branches etc.) A copy of my week-by-week time-log, is available under Fig. 3 of Appendix A: Figures.

Each team member was also allowed to review the meetings of a particular week, as Anthony took the minutes of each meeting, that we could use for continuous reflection throughout the project (a sample of these minutes, is available at Fig. 4 of Appendix A: Figures).

We used Trello, for a team-level overview of which tasks team-members were currently completing, as well as for communicating certain matters on the project, putting up notes going into further detail for certain aspects of the project etc. A sample of the Trello-Boards we part together, can be found under Fig. 5 and 6 of Appendix A: Figures.

Text-messages were also used on occasion (mostly between me and Jack), as a faster means of communication, when a response within a few minutes was paramount.

# Reflection

## Negative Aspects

* Clashes between me and another team-member: These were on certain components of the project, such as how collision should be handled, issues relating to coding standards and misunderstanding between me and him, on what issues we were currently facing, or success stories for certain User-Stories (mistaking one for the other)
* Concerning Modularity: As was initially planned for the project, we had intended for there to be multiple levels, with enemies having different attack patterns (causing the Player to have to adopt new movement strategies for dodging Enemy projectiles), as well as different types of enemies (commander, boss, spawner), which would have different stat lines. Even so, we had not tried to make the project’s implementation in Unity (the game engine we used), modular, which would allow for an iterative pattern to develop new levels and introduce new enemy-types to the Player (as for the one level we had, many of its assets were static for that level, not considering different types of enemies and attack patterns for example)

## Positive Aspects

* Project Tracking: We made sure to track our progress in the project, by using our own timesheet’s (a sample of mine is shown in Fig. 3 of Appendix A: Figures), as well as meeting at least once a week, to discuss our current progress and to delegate any ‘loose’ tasks (that were not delegated), to a suitable team-member
* Task Completion: Once the tasks were distributed (either in the initial stages or from a team-meeting), we all completed the tasks we had been delegated, to the best of our ability, considering the time we had predicted for each task (although, the tasks were not delegated in the fairest manner, see the Improvements section for further detail into this)

# Improvements

* Development Standards: No standard was adopted by the team, other than the one for the tracking system we used, in development of the project. This includes no Coding-Standard for the C#-scripts we assembled for use in the Project’s Unity implementation, as well as standards for assembling other assets in Unity (such as ‘prefabs’, accounting for modularity to use them at multiple points in the project). Therefore, standards to normalise the development process across the whole team (so we would be ‘on the same page’), for at least what has been mentioned, should have be adopted from the beginning, for future team-projects
* Properly Delegating Tasks: Although we made sure to delegate all of the necessary tasks for completion, fairly evenly (in terms of the quantity of tasks), between each team-member, we would not have taken into account which team-member was best suited to a task. Therefore, when delegating tasks, a team should properly analyse whom would be best suited at completing a task, given the predicated time for that task, as well as a given team-member’s skillset. A system should be adopted for consistent analysis of task-delegation, at any stage of the project
* Utilising a system to account for ‘slack’: Although we had a very capable tracking system in place for the tasks we had delegated ourselves in the first instance, there was no system to allow for a team-member whom had completed the tasks assigned to them, to help other team-members with tasks they were having issues with, or to initiate tasks that had not been delegated to the team. Therefore, it would make sense adopt a system to allow team-members to pick-up ‘slack’, for task-progress, to make sure a team-member was not left idle at a certain stage in the project, when they could be getting on with other tasks that are suitable for them to complete, or helping other team-members with any issues they are having with their tasks

# Appendix A: Figures

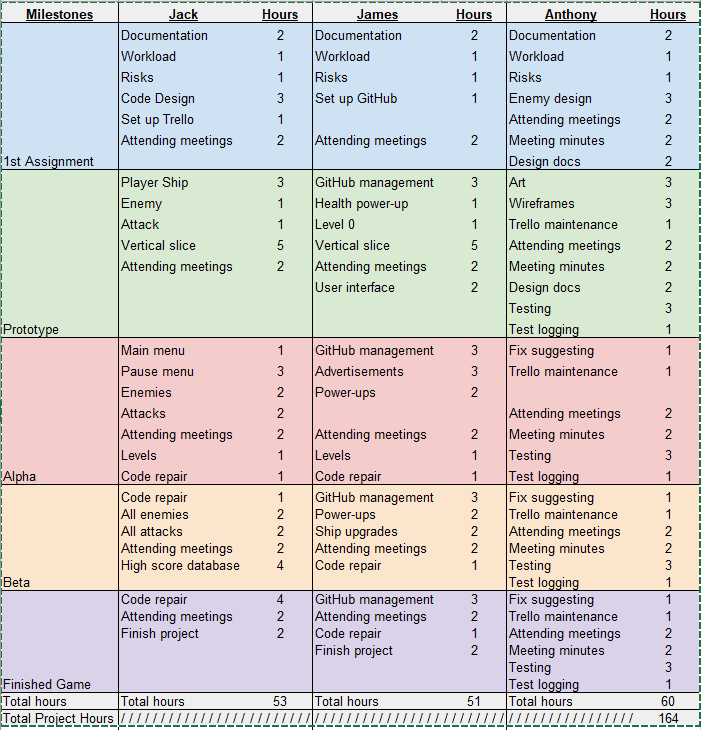
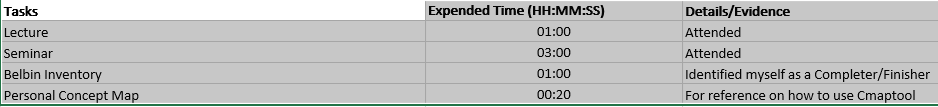
Figure 1: The group-workload sheet of each team-member’s delegated tasks, and the predicted hours for each task:

Figure 2: An example of the table that was used for Engineering Software Systems, as a basis for the table I have put together for use in this project:



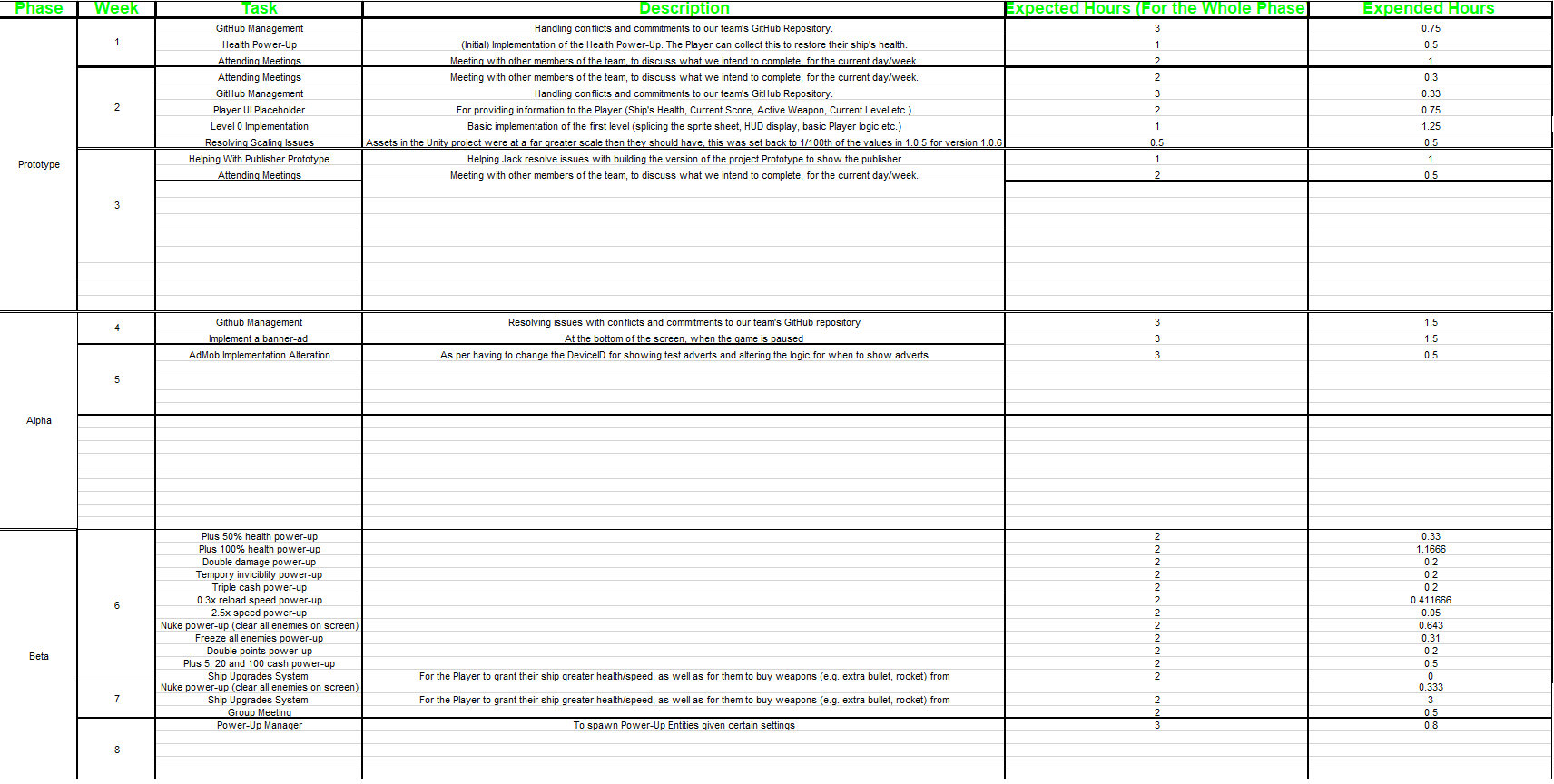
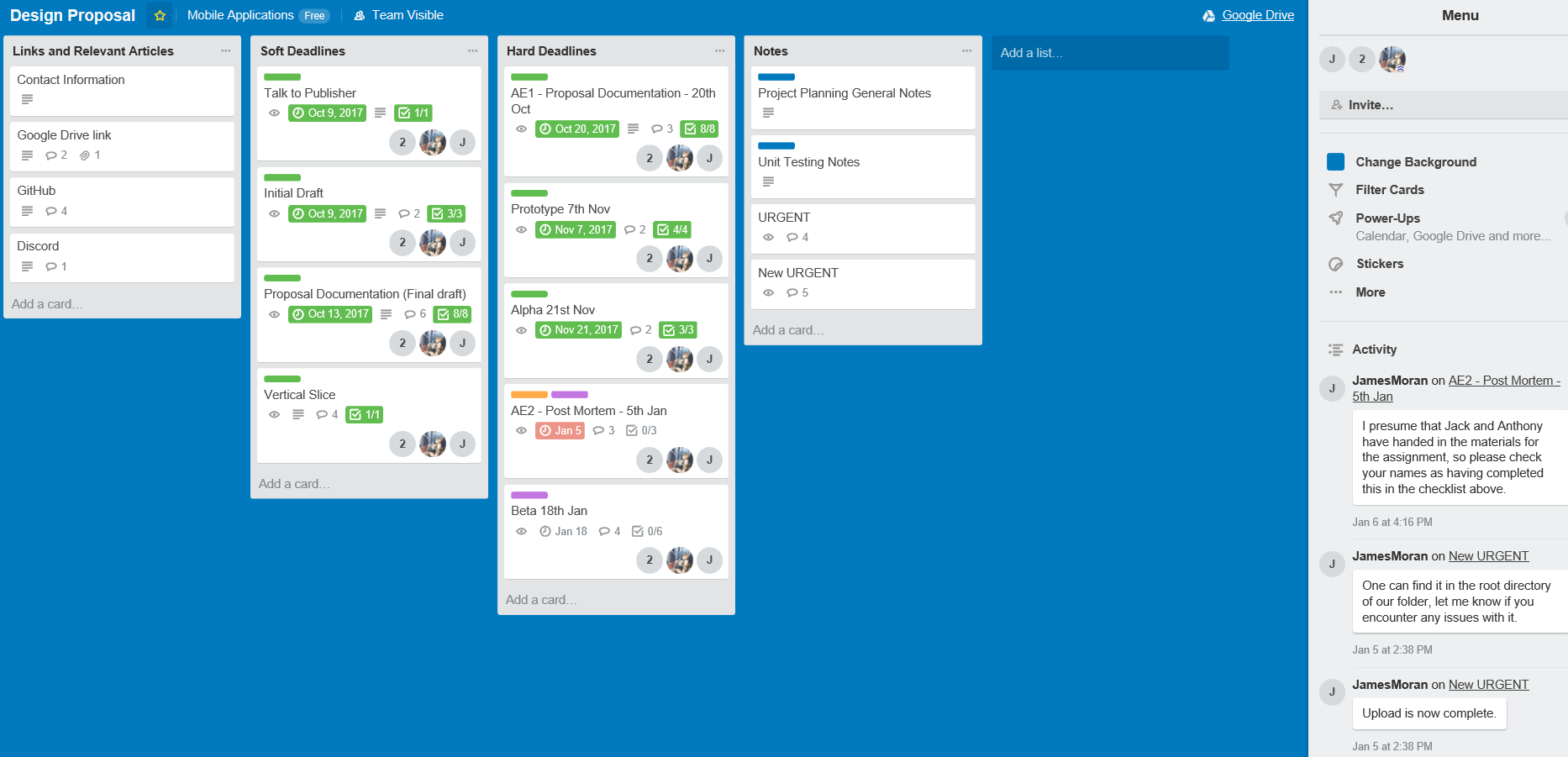
Figure 3: A sample of my week-by-week time-log used during the project:

Figure 4: A sample of the minutes for a meeting, put together by Anthony Boys:

This group meeting took place on the 2nd of October 2017, at 5:00 PM.  
People present: Anthony Boys, Jack Evans, James Moran.

People absent: none.

In this meeting, we first of all set up a GitHub and Trello board for the project.  
Finally, we discussed an idea for the project. We all agreed that the design will be a simple game, with the genre of either “bullet hell” or “space invaders”; that the game may have controls that are alternative to using an accelerometer or gyroscope; that there may be a system of player progression; that there will be differing enemies, including boss enemies; that difficulty will increase over levels; and that there may be power-ups, health pickups, drops and end-of-level bonuses. (Anthony Boys, 2017)

Figure 5: Our ‘Design Proposal’ Trello-Board (the primary board that we used throughout the duration of the project):

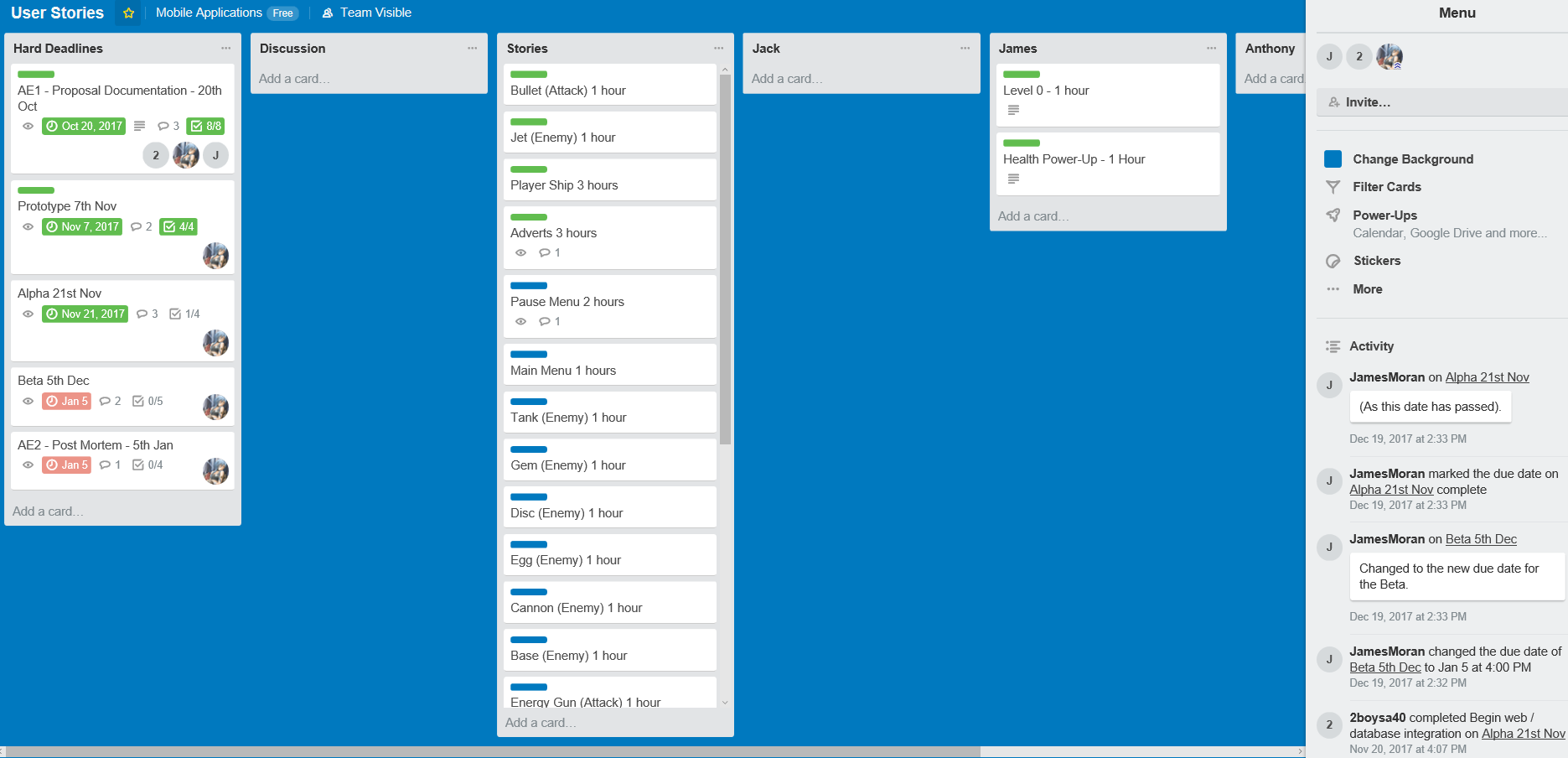
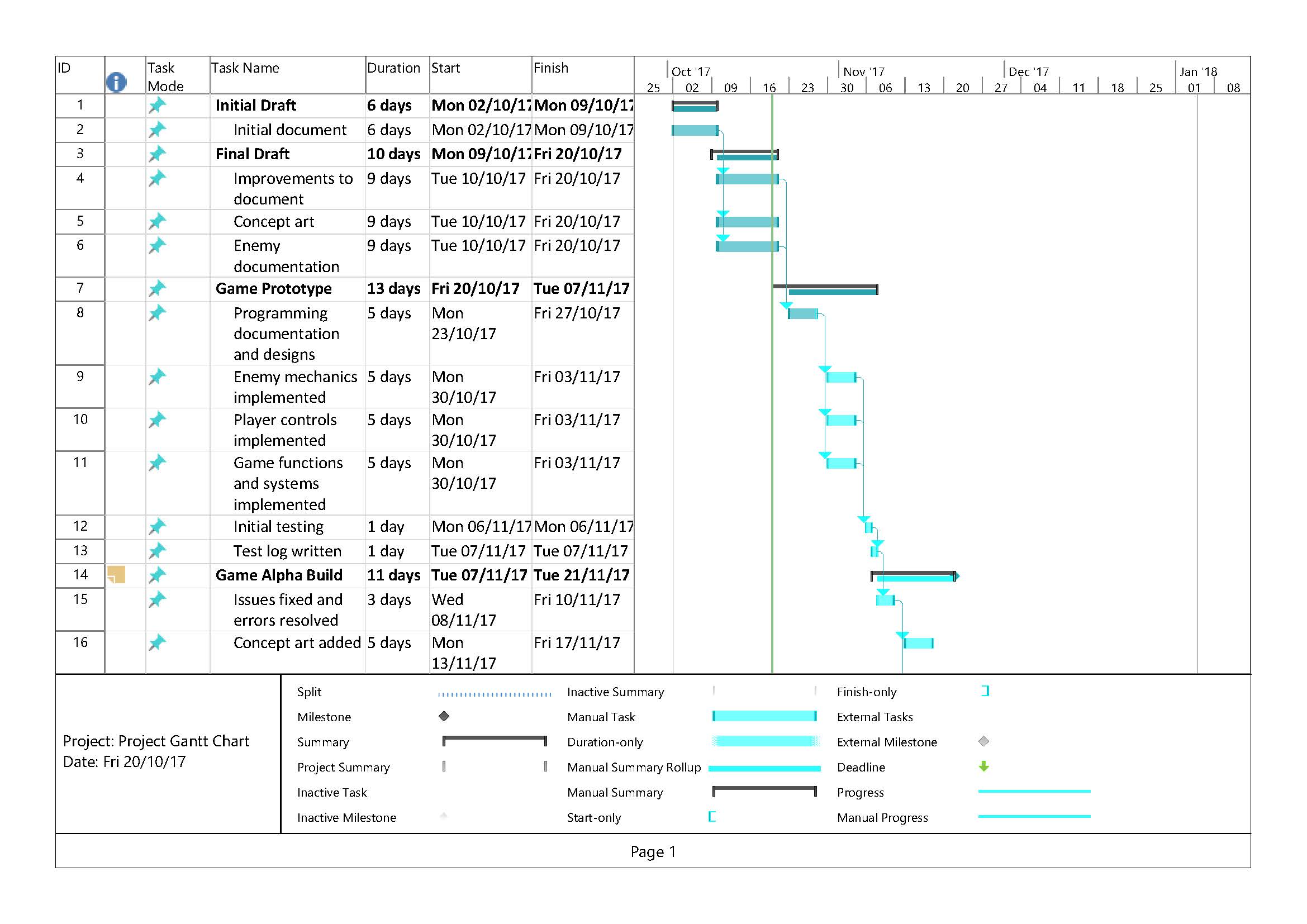
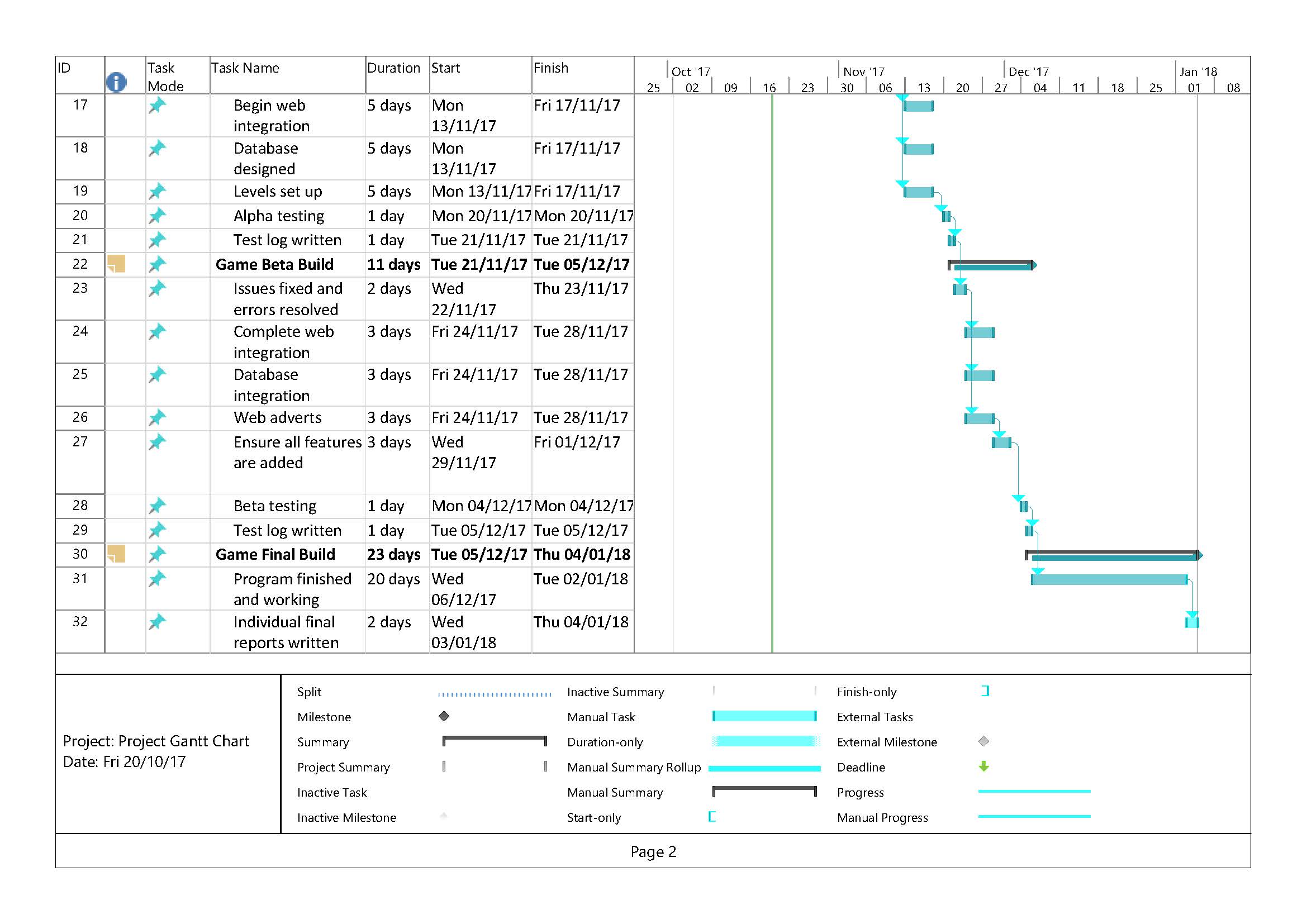
Figure 6: Our ‘User Stories’ Trello-Board (a secondary board used for tracking progress on the implementation of specific user-stories for the project):

Figure 7: The pert chart put together for our project:

Figure 8: Page 1 of the Gantt Chart assembled for our project:

Figure 9: Page 2 of the Gantt Chart assembled for our project:

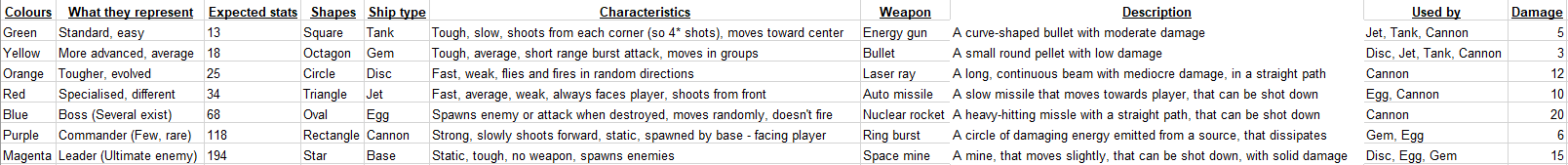
Figure 10: The table of ship types, descriptions and statistics, for each type of ship intended to be implemented in the game: (Antony Boys, 2017)

Figure 11: A copy of the tests to be conducted on our project (a full system test):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Id** | **Function tested** | **Expected outcome of the function** | **Test result** | **Pass / fail** |
| 1 | Game program loads | The program loads correctly upon launch |  |  |
| 2,a | Play button in main menu functions sufficiently | The user is taken to their current level |  |  |
| 2,b | Shop button in main menu functions sufficiently | The user is taken to the in-game shop |  |  |
| 2,c | High scores button in main menu functions sufficiently | The user is shown the high scores database |  |  |
| 2,d | Quit button in main menu functions sufficiently | The application exits and closes |  |  |
| 3,a,I | Movement control in game functions sufficiently | The player moves around the screen |  |  |
| 3,a,II | Camera moves with player | The camera follows the player, to wherever the player moves to, whenever the player moves |  |  |
| 3,b | Shooting control in game functions sufficiently | The particles relevant to the user's weapon is fired |  |  |
| 3,c | Pause button in game functions sufficiently | The pause menu displays over a paused game |  |  |
| 4,a | Resume button in pause menu functions sufficiently | The user returns to the game |  |  |
| 4,b | Main menu button in pause menu functions sufficiently | The user's level is stored, and the user is taken to the main menu |  |  |
| 4,c | Quit button in pause menu functions sufficiently | The user's level is stored, and the application exits and closes |  |  |
| 5,a,I | Balance displays in shop | The quantity of currency that the user owns is displayed |  |  |
| 5,a,II | Score displays in shop | The total score is the user is displayed |  |  |
| 5,a,III | Username displays in shop | The user's chosen name is displayed |  |  |
| 5,a,IV | Item names appear in shop | The names of the items for sale in the shop are displayed |  |  |
| 5,b,I | Correct amount of lives displays in shop | The quantity of lives that the user owns, plus one, is displayed |  |  |
| 5,b,II | Correct amount of health points displays in shop | The maximum amount of health points that the user can own after the upgrade is purchased is displayed |  |  |
| 5,b,III | Correct speed displays in shop | The user's speed after the upgrade is purchased is displayed |  |  |
| 5,b,IV | Correct fire rate displays in shop | The user's fire rate after the upgrade is purchased is displayed |  |  |
| 5,b,V | Correct weapon statistics display in shop | The statistics of the next weapon is displayed |  |  |
| 5,b,VI | Correct prices display in shop | The prices of all the articles that are sold within the shop are displayed correctly |  |  |
| 5,c,I | Buying an extra life functions sufficiently | The amount of lives that the user owns increases by one, and both the price and the amount displayed in the shop updates, and the user's balance decreases |  |  |
| 5,c,II | Buying a health upgrade functions sufficiently | The maximum amount of health points that the user can own increases, and both the price and the amount displayed in the shop updates, and the user's balance decreases | |  |
| 5,c,III | Buying a speed upgrade functions sufficiently | The user's maximum speed increases, and both the price and the amount displayed in the shop updates, and the user's balance decreases |  |  |
| 5,c,IV | Buying a fire rate upgrade functions sufficiently | The user's fire rate increases, and both the price and the amount displayed in the shop updates, and the user's balance decreases |  |  |
| 5,c,V | Buying a new weapon functions sufficiently | The user's weapon changes, and the new weapon and its statistics display in the shop, and the user's balance decreases |  |  |
| 5,c,VI | Donating functions sufficiently | The user's balance decreases |  |  |
| 6,a,I | Level number displays in game scene | The number of the level that the player is currently on is displayed |  |  |
| 6,a,II | Balance displays in game scene | The amount of currency that the user owns is displayed |  |  |
| 6,a,III | Score displays in game scene | The total score is the user is displayed |  |  |
| 6,a,IV | Amount of health displays in game scene | The quantity of health points that the user currently owns is displayed |  |  |
| 6,a,V | Amount of lives displays in game scene | The quantity of lives that the user owns is displayed |  |  |
| 6,a,VI | Player weapon icon displays in game scene | The icon of the weapon that is currently equipped is displayed |  |  |
| 6,a,VII | Player weapon name displays in game scene | The name of the weapon that is currently equipped is displayed |  |  |
| 6,b,I | Bullet functions sufficiently | The bullet moves in a straight path from its spawn position, and deals 3 damage upon collision |  |  |
| 6,b,II | Energy particle functions sufficiently | The energy particle moves in a straight path from its spawn position, and deals 5 damage upon collision |  |  |
| 6,b,III | Laser ray functions sufficiently | The laser ray moves in a straight path from its spawn position, and deals 12 damage upon collision |  |  |
| 6,b,IV | Space mine functions sufficiently | The space mine moves slightly in random directions from its spawn position, and deals 15 damage upon collision |  |  |
| 6,b,V | Ring burst functions sufficiently | The ring burst emits and expands from its spawn position, deals 6 damage upon collision, and is destroyed when it takes 1 or more damage |  |  |
| 6,b,VI | Auto missile functions sufficiently | The auto missile moves slowly towards the player, deals 10 damage upon collision, and is destroyed when it takes 1 or more damage |  |  |
| 6,b,VII | Nuclear rocket functions sufficiently | The nuclear rocket moves slowly in a straight path from its spawn position, deals 20 damage upon collision, and is destroyed when it takes 1 or more damage |  |  |
| 6,c | Shop displays in between levels | The user is taken to the shop upon the completion of any level |  |  |
| 7,a,I | Colliding with a +5¤ power-up functions sufficiently | The user's balance increases by 5 |  |  |
| 7,a,II | Colliding with a +20¤ power-up functions sufficiently | The user's balance increases by 20 |  |  |
| 7,a,III | Colliding with a +100¤ power-up functions sufficiently | The user's balance increases by 100 |  |  |
| 7,a,IV | Colliding with a +50% h.p. power-up functions sufficiently | The user's health increases by a maximum of 50%, providing that the total amount of health doesn't exceed 100% |  |  |
| 7,a,V | Colliding with a +100% h.p. power-up functions sufficiently | The user's health increases by a maximum of 100%, providing that the total amount of health doesn't exceed 100% |  |  |
| 7,b,I | Colliding with a double damage power-up functions sufficiently | The damage dealt by the user's weapons temporarily increases by 100% |  |  |
| 7,b,II | Colliding with a double points power-up functions sufficiently | The amount of points scored by the user temporarily increases by 100% |  |  |
| 7,b,III | Colliding with a 0.3\* reload time power-up functions sufficiently | The user's fire rate temporarily increases by 233% |  |  |
| 7,b,IV | Colliding with a triple cash power-up functions sufficiently | The amount of currency power-ups obtained by the user temporarily have their currency value increased by 200% |  |  |
| 7,b,V | Colliding with a 2.5\* speed power-up functions sufficiently | The user's speed temporarily increases |  |  |
| 7,b,VI | Colliding with an invincibility power-up functions sufficiently | The user temporarily becomes immune to receiving damage |  |  |
| 7,b,VII | Colliding with an enemies freeze power-up functions sufficiently | The enemies that are displayed on the screen are temporarily unable to move or fire |  |  |
| 7,b,VIII | Colliding with a nuke power-up functions sufficiently | The non-boss enemies that are displayed on the screen are destroyed |  |  |
| 8,a,I | Green base functions sufficiently | The green base spawns green enemies |  |  |
| 8,a,II | Green cannon functions sufficiently | The green cannon shoots a cluster of 3 bullets forward every 3.33 seconds, this unit has 12 health and doesn't move |  |  |
| 8,a,III | Green disc functions sufficiently | The green disc shoots a bullet in a random direction every 1.67 seconds, this unit has 6 health and moves at a rate of 15 speed in a random direction |  |  |
| 8,a,IV | Green egg functions sufficiently | The green egg has a 90% chance of hatching into 4 ring bursts, and a 10% chance of spawing a green enemy |  |  |
| 8,a,V | Green gem functions sufficiently | The green unit shoots a ring burst every 2 seconds, this unit has 11 health and moves at a rate of 3 speed forward |  |  |
| 8,a,VI | Green jet functions sufficiently | The green jet shoots an energy particle forward every 2.5 seconds whilst facing the player, this unit has 7 health and moves at a rate of 12 speed forward |  |  |
| 8,a,VII | Green tank functions sufficiently | The green tank shoots a bullet from each corner every 5 seconds, this unit has 13 health and moves at a rate of 2 speed forward |  |  |
| 8,b,I | Yellow base functions sufficiently | The yellow base spawns yellow enemies |  |  |
| 8,b,II | Yellow cannon functions sufficiently | The yellow cannon shoots a nuclear rocket forward every 5 seconds, this unit has 14 health and doesn't move |  |  |
| 8,b,III | Yellow disc functions sufficiently | The yellow disc shoots a bullet in a random direction every 1 second, this unit has 8 health and moves at a rate of 16 speed in a random direction |  |  |
| 8,b,IV | Yellow egg functions sufficiently | The yellow egg has a 65% chance of hatching into 6 ring bursts, and a 35% chance of spawning a yellow enemy |  |  |
| 8,b,V | Yellow gem functions sufficiently | The yellow gem shoots a space mine every 4.29 seconds, this unit has 14 health and moves at a rate of 5 speed forward |  |  |
| 8,b,VI | Yellow jet functions sufficiently | The yellow jet shoots a cluster of 3 bullets forward every 3.33 seconds whilst facing the player, this unit has 7 health and moves at a rate of 12 speed forward |  |  |
| 8,b,VII | Yellow tank functions sufficiently | The yellow tank shoots an energy particle from each corner every 5.71 seconds, this unit has 15 health and moves at a rate of 4 speed forward |  |  |
| 8,c,I | Orange base functions sufficiently | The orange base spawns orange enemies |  |  |
| 8,c,II | Orange cannon functions sufficiently | The orange cannon shoots a cluster of 4 enery particles forward every 3.64 seconds, this unit has 20 health and doesn't move |  |  |
| 8,c,III | Orange disc functions sufficiently | The orange disc shoots a space mine in a random direction every 3.33 seconds, this unit has 10 health and moves at a rate of 20 speed in a random direction |  |  |
| 8,c,IV | Orange egg functions sufficiently | The orange egg has a 45% chance of hatching into 3 space mines, and a 55% chance of spawning an orange enemy |  |  |
| 8,c,V | Orange gem functions sufficiently | The orange gem shoots a ring burst every 1.11 seconds, this unit has 15 health and moves at a rate of 6 speed forward |  |  |
| 8,c,VI | Orange jet functions sufficiently | The orange jet shoots a bullet and an energy particle forward every 1.82 seconds whilst facing the player, this unit has 11 health and moves at a rate of 20 speed forward |  |  |
| 8,c,VII | Orange tank functions sufficiently | The orange tank shoots a cluster of 2 bullets from each corner every 4.44 seconds, this unit has 16 health and moves at a rate of 5 speed forward |  |  |
| 8,d,I | Red base functions sufficiently | The red base spawns red enemies |  |  |
| 8,d,II | Red cannon functions sufficiently | The red cannon shoots 2 laser rays forward every 3.08 seconds, this unit has 24 health and doesn't move |  |  |
| 8,d,III | Red disc functions sufficiently | The red disc shoots a bullet and a space mine in a random direction every 2.72 seconds, this unit has 14 health and moves at a rate of 22 speed in a random direction |  |  |
| 8,d,IV | Red egg functions sufficiently | The red egg has a 60% chance of hatching into 6 auto missiles, and a 40% chance of spawning a red enemy |  |  |
| 8,d,V | Red gem functions sufficiently | The red gem shoots a ring burst and a space mine every 2.72 seconds, this unit has 19 health and moves at a rate of 13 speed forward |  |  |
| 8,d,VI | Red jet functions sufficiently | The red jet shoots a cluster of 2 bullets forward every 0.9 seconds whilst facing the player, this unit has 12 health and moves at a rate of 23 speed forward |  |  |
| 8,d,VII | Red tank functions sufficiently | The red tank shoots a bulletand an energy particle from each corner every 4 seconds, this unit has 17 health and moves at a rate of 5 speed forward |  |  |
| 8,e,I | Blue base functions sufficiently | The blue base spawns blue enemies |  |  |
| 8,e,II | Blue cannon functions sufficiently | The blue cannon shoots a spread of 2 nuclear rockets forward every 2.5 seconds, this unit has 44 health and doesn't move |  |  |
| 8,e,III | Blue disc functions sufficiently | The blue disc shoots a ring of 8 bullets every 1.67 seconds, this unit has 30 health and moves at a rate of 30 speed in a random direction |  |  |
| 8,e,IV | Blue gem functions sufficiently | The blue gem shoots 3 ring bursts every 1.25 seconds, this unit has 40 health and moves at a rate of 20 speed forward |  |  |
| 8,e,V | Blue jet functions sufficiently | The blue jet shoots a spread of 2 energy particles forward every 0.67 seconds whilst facing the player, this unit has 22 health and moves at a rate of 32 speed |  |  |
| 8,e,VI | Blue tank functions sufficiently | The blue tank shoots a spread of 3 bullets from each corner every 2.4 seconds, this unit has 40 health and moves at a rate of 14 speed forward |  |  |
| 8,f,I | Purple base functions sufficiently | The purple base spawns purple enemies |  |  |
| 8,f,II | Purple cannon functions sufficiently | The purple cannon shoots a spread of 5 laser rays forward every 2.22 seconds, this unit has 81 health and doesn't move |  |  |
| 8,f,III | Purple disc functions sufficiently | The purple disc shoots a cluster of 3 space mines every 1.67 seconds, this unit has 51 health and moves at a rate of 30 speed in a random direction |  |  |
| 8,f,IV | Purple gem functions sufficiently | The purple gem shoots a ring of 3 space mines every 1.11 seconds, this unit has 59 health and moves at a rate of 22 speed forward |  |  |
| 8,f,V | Purple jet functions sufficiently | The purple jet shoots 2 spreads of 3 bullets forward every 0.67 seconds whilst facing the player, this unit has 49 health and moves at a rate of 32 speed forward |  |  |
| 8,f,VI | Purple tank functions sufficiently | The purple tank shoots 3 bullets and an energy particle from each corner every 2 seconds, this unit has 55 health and moves at a rate of 16 speed forward |  |  |
| 8,g,I | Magenta base functions sufficiently | The magenta base spawns magenta enemies |  |  |
| 8,g,II | Magenta cannon functions sufficiently | The magenta cannon shoots 2 auto missiles and 2 nuclear rockets forward every 1.25 seconds, this unit has 102 health and doesn't move |  |  |
| 8,g,III | Magenta disc functions sufficiently | The magenta disc shoots 4 bullets in a random direction and 3 space mines every 1.25 seconds, this unit has 86 health and moves at a rate of 40 speed in a random direction | |  |
| 8,g,IV | Magenta gem functions sufficiently | The magenta gem shoots a ring burst and a ring of 4 space mines every 1.43 seconds, this unit has 95 health and moves at a rate of 25 speed forward |  |  |
| 8,g,V | Magenta jet functions sufficiently | The magenta jet shoots a spread of 3 energy particles forward every 0.33 seconds whilst facing the player, this unit has 85 health and moves at a rate of 32 speed forward | |  |
| 8,g,VI | Magenta tank functions sufficiently | The magenta tank shoots 4 bullets and an energy particle from each corner every 1.43 seconds, this unit has 84 health and moves at a rate of 22 speed forward |  |  |
| 9,a,I | Banner advertisements display in main menu | The main menu contains a functioning banner advertisement somewhere on the screen |  |  |
| 9,a,II | Banner advertisements display in pause menu | The pause menu contains a functioning banner advertisement somewhere on the screen |  |  |
| 9,a,III | Banner advertisements display in shop menu | The shop menu contains a functioning banner advertisement somewhere on the screen |  |  |
| 9,b | Interstital advertisements display in between screens | The interstital page between levels, the shop, and pages of the menu, all contain a functioning advertisement |  |  |
| 9,c | High scores are sent to database when game ends | The user's total score is sent to the high scores database to be stored |  |  |
| 9,d | Current high scores display when button is tapped | The high scored database that is displayed is consistently updating when new high scores are enterd |  |  |
| 9,e,I | Database stores player's statistics and progress | The user's current statistics and progress, consisting of scores, currency balance, level, upgrade purchases and life quantity, are stored in the database |  |  |
| 9,e,II | Database loads player's statistics and progress | The user's stored statistics and progress, consisting of scores, currency balance, level, upgrade purchases and life quantity, are accessed and loaded into the game |  |  |

(Antony Boys, 2017)

Figure 12:

# References

1. Anthony Boys, 2017. *Copy of Meeting Minutes* [Viewed on the 10/01/2018]. Available from: <https://drive.google.com/open?id=1zxLlAtjeqHG9ca_-QfohmyLYrLHpk49oHcUAeVVD6rI>