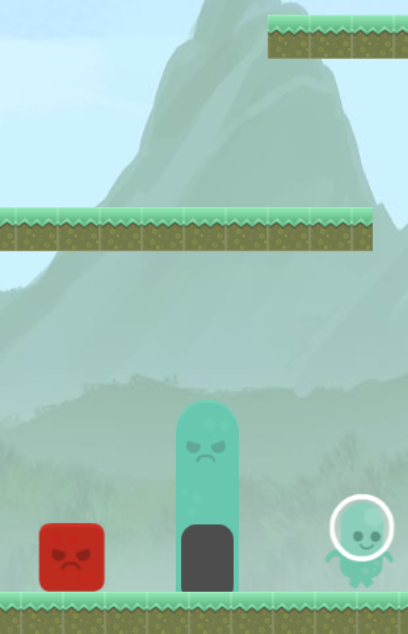
Coursework | Mobile Game Dev | April 8, 2015

Project Summary

B00235610 & B00236297



# Overall Structure

I will be quickly going over major differences from the initial specification. But I will also be writing down the additional changes added and giving reasons for these changes so they are justified.

### Major Differences

The first major difference is in how we did the AI for the enemies. We first had said that we would build a 2D array of blocks which represent the places where the AI can go on the map, so basic use of the A\* pathfinding for the enemies. However we decided to scrap that in a favor of a state based behavior system with 2 types of behaviors for spawned enemies which was simpler and also less computationally expensive. Another major difference was when it came to the server. We initially planned this just to be a simple server that provided high scores and news. For debugging reasons we then needed to add additional methods for deleting certain scores and news elements as well as look at how we test to make sure the server is still active. Input was also changed from the initial specification to better suit touch controls and with this change we ended up with a more playable version of the game. Other notable major differences include increased bullet mechanics with bullets bouncing off walls and the bounds of the screen as well as having the capability to push enemies back. Scoring was also another element that was changed. The score system we had built resulted in negative scores most of the time so we changed it to a more positive score system which awards more accurate and time efficient players with high scores.

### Additional Features

Additional features including level loading from the server allowing us to balance the game without redistributing the code. We found that the news scoring system gave the game a little replay value allowing friends to compete. We also have the game running online on a server ensuring it can be access and marked via the web and have taken a great amount of time to generate documentation for both the server and the game. Additionally we optimized the game in many ways this includes minifying the JavaScript game code and compressing image files to improve download speeds on 2G and 3G mobile connections.

# Successes & Failings

Successes

* Getting a working version of the game which runs great on both mobile and desktop.
* Having a server implementation which allows us to tweak the levels without altering the code.
* Online elements were successfully implemented allowing us to have a more expansive, community driven game.
* The game ran well and smooth on the devices we tested it on which weren’t the fastest ones around so we are extra confident that the game will behave as expected on a wider range of devices.
* We worked well as a team and came in during the Easter break and did pair programming which saw us develop the game much quick than we expected. Leaving us with a playable version of the game 3 weeks early.

Failings

* We initially had some problems with CORS. Meaning data from the server couldn’t be accessed
* We no longer required A\* which meant initial research and implementation was scrapped.
* The initial input format we set wasn’t suitable and was really annoying so we changed it.
* Some browsers won’t render the game correctly due to a WebGL issue, this was only found on B00235610’s phone though so it’s difficult to say if it’s actually a browser issue or just an issue with the phone itself.
* Other than that we didn’t have any major failings as we feel this was a very successful project.

# Contributions

B00235610

B00235610 major contribution was writing the node js server which is now running on his works server (with their permission). The server provides an API to the application to get and post news as well as handle our high score table. Uniquely he makes each score and news unique to each version of the application. As you might expect he also wrote the http post and get request objects which are just simple objects that make it easy to get data off the server. He additionally wrote the level loading code which again is stored on the server so we can easily tweak the difficulty for all applications. The level is stored as a JSON file which is then returned to the application when it is loaded. Finally he also wrote the audio code and acquired a lot of the assets (textures & audio files (MP3 & OGG)) used in the game and assisted with documentation and the other major features such as the AI and physics.

B00236297

B00236297 major contribution was writing both the physics code and the enemy AI code which were essential to the game. He also wrote it in a good object oriented manor which allows us to add additional AI behaviors in the future. He also implemented the game mechanics which meant writing a good amount of the interaction logic. Including handling input for both the mouse and touch controls. Additionally he assisted in the documentation for the game project and generated documentation using JSdoc to ensure we have good consistent and readable documentation to put in our official submission, He also responsible for the current game design itself as well as tweaking and designing the 5 levels you see in game.

# Collaboration

We would wish to point out that the amount of collaboration on this project using Trello was minimal for a simple reason. As we share every single class there was less need on collaboration using Trello as we would regularly meet to share updates and ideas 3 days a week. It’s also worth noting that when working on the project we decided to come into the labs and do pair programming as we feel it’s a great way of doing a collaborative job. This is why Trello isn’t as populated as it should be and why there are an uneven number of commits as during the Easter week and the days after it we worked on the game on B00235610’s machine hence why he has the ID for most of the commits.

# Links

Down below I have left links to the Bitbucket repository, Trello collaboration website and links to both the game on B00235610’s personal website which holds the online version of the game. Inside the submission folder you will find the documentation generated using JSdoc should you wish to take a look at the websites generated by this toolkit.

Bitbucket <https://bitbucket.org/WilliamTaylor/mobilegamedev>

<https://bitbucket.org/WilliamTaylor/mobilegamedev-server>

Trello <https://trello.com/b/tm2N9yny/mobile-game-dev-team>

Link to online game <http://www.williamsamtaylor.co.uk/programs/MGD/index.html>