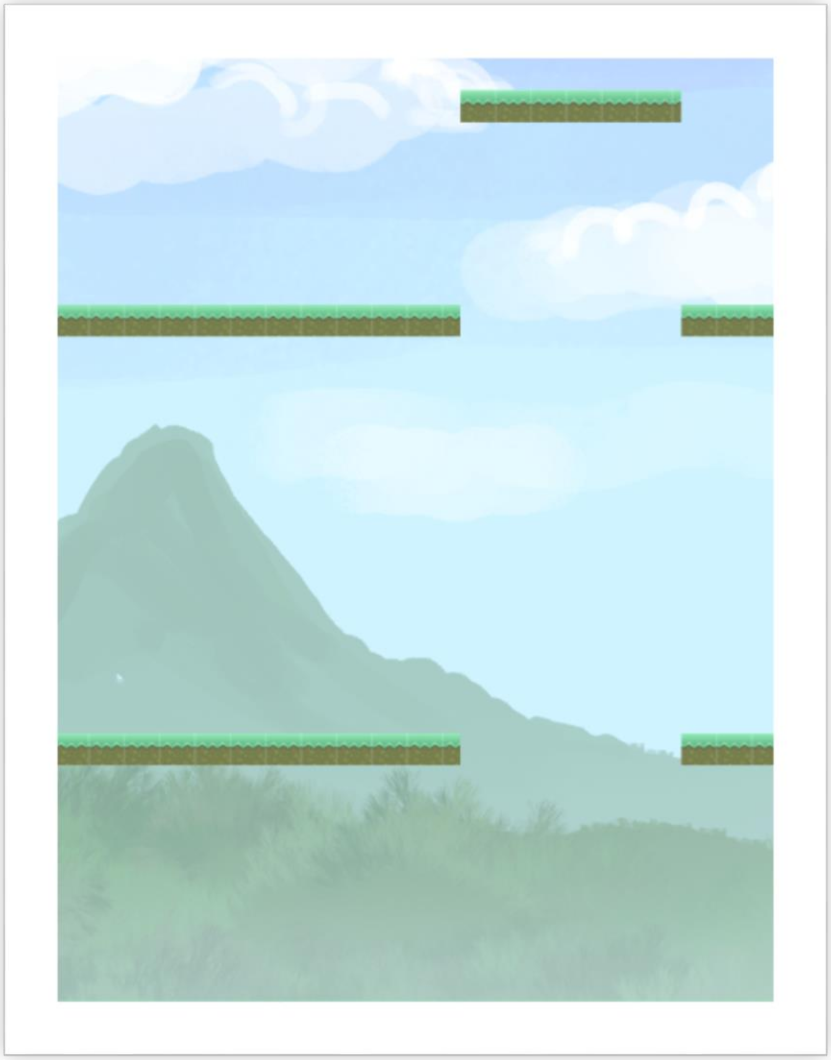
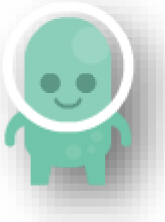
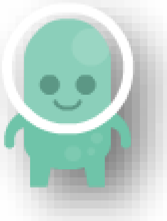


Game Specification

AN EDGE OF 5 KINGDOMS MOBILE GAME



Profile| **Mobile Game Development** | March 27, 2015

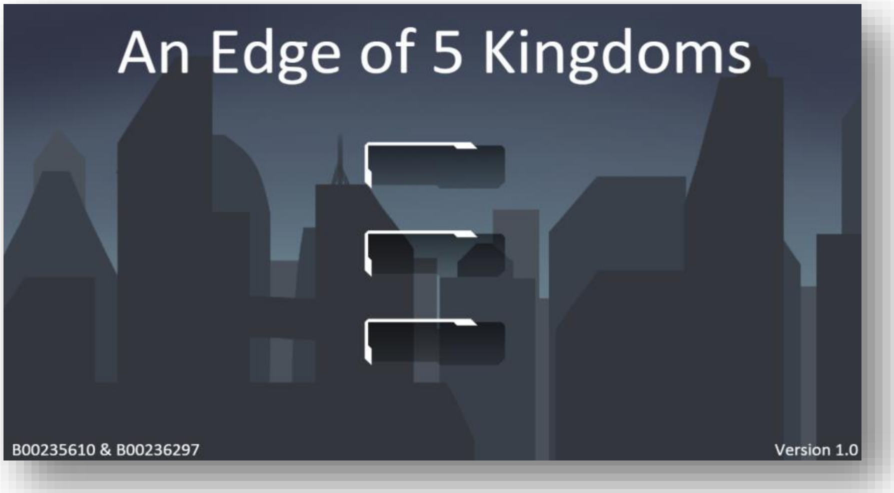


Game Design

The Idea

“An Edge of 5 kingdoms” is meant to be a simple platform shooter with 5 levels as you probably guessed from the title. The challenge not only lies in dodging pellets you fire at enemies to remove them from the level but also matching up the right pellet to kill a certain enemy. The overall goal of the game is to reach a high score which survives on a leader board downloaded from a database table. The following features from those listed in the coursework requirements will be built into the application.

- 2D Physics (Done using Phaser)
- NPC intelligence using A* path finding
- Online data store with AWS, NodeJS



Progression

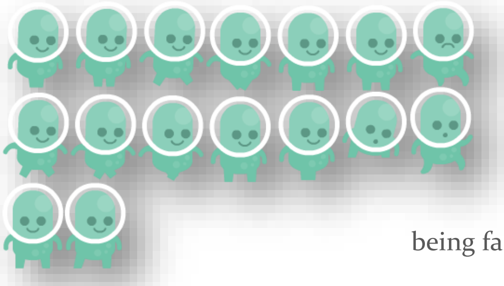
Progression is a simple step of killing all the enemies in a single scene and then being able to progress to the next one by moving right until you enter the next level. The player will need to traverse the entire scene as not all enemies will fall down to the lower level and will remain on higher platforms until they are killed.

Enemies

There are 3 types on enemies in the game and there are two behaviors for these 3 types. The first behavior is the stationary behavior where the enemy can’t leave the platform it has spawned on. The other is the limitless behavior where the enemy will follow the player down platforms. Also the pellets that the player can shoot with have to match the colour of the enemy if they don’t it will bounce of at a tangent.

Player

The player is a simple animated sprite which is controlled entirely by touch controls. A single tap gestor will set a way point for the character and a double tap will fire a projectile to land around that spot. This simple interaction system for the player should make it easy to control the player on a touch screen or a small device. The player has no life and a single contact with an enemy will cause a death and a respawn a number of seconds later. The only negative effect being a lower score at the end as deaths acts as score deduction. It’s meant to be an easy to play game that rewards careful and thoughtful players that complete the game well by being fast accurate and careful.



Scoring

Scoring is a simple system in which killing enemies with the right colour projectile will score positive points for your score and wrong colours or any death will score negative points. A full list of actions that effect the ending score are listed below. As you will see this will result in a wide variety of scores and tactics that can be applied to the game to get the best score. This is also the key element to the game as there is only a single player mode. The competitive state of play relies

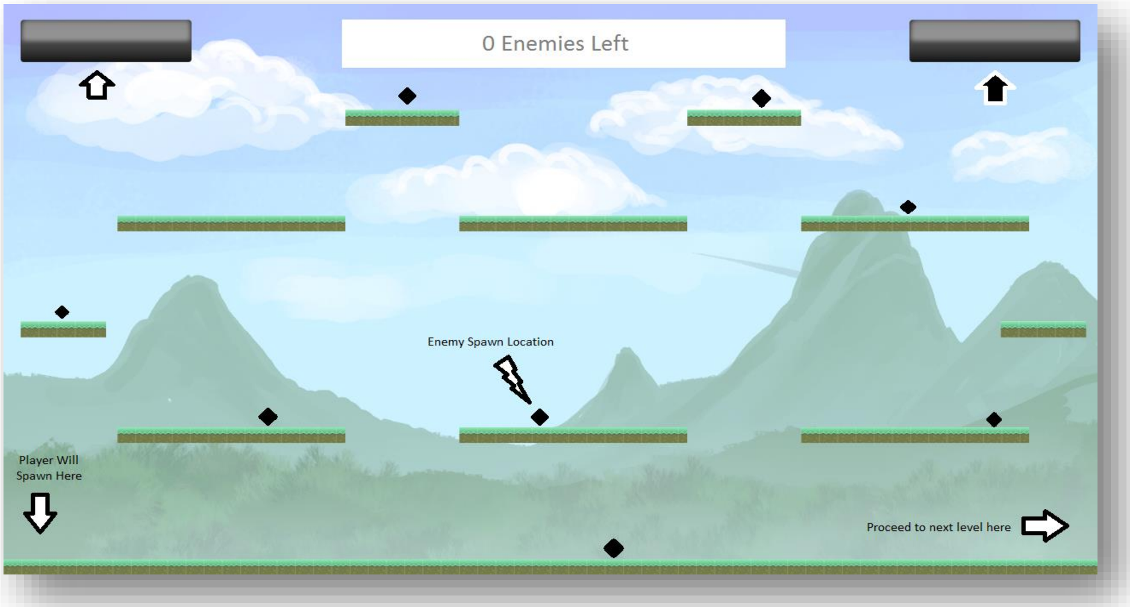
Actions	Points
Death	-10
Enemy Killed	2
Wrong Projectile	-1
Stage Cleared	100
Time Taken	250 – 1 for each second passed
No deaths in stage	75

A Sample Play Through of the First Level

This will be a short walkthrough of the level to demonstrate the game in practice. Both developers thought this would be better than an overly detailed story board in describing what the game will look like. For this description to be truly effective please look at the screenshot below which will help convey the general gameplay that the game will provide. Also note that the gameplay is the same across each level the changes for each level include platform re design, harder AI etc.

The player will spawn in the lower left corner of the screen when entering the level. Enemies which were explained earlier in the document will spawn periodically at one of the many locations indicated in the screenshot the location is picked at random. The players objective obviously is destroy all enemies in the scene at which point they can move on. The player is to move on when all enemies are destroyed at which point then can walk right, out of the scene. The next scene will be then loaded in.

The two buttons at the top represent the heads up display. The one on the left is a simple button that returns the user to the menu. It can be activated with a single tap, the other switches the pellets being fired. Note their isn't an indication what one is currently set which is a design decision to add greater difficulty in remembering which one comes next in order to get the best score.



Development

Integrated Development Environment

We will be using the Visual Studio IDE to develop our game and our server. The reason why is because both developers in the team have gotten used to a google JavaScript plugin for Visual Studio which makes it a much more capable and half decent web development tool. Also B00235610 who will be writing the server also has some experience of writing a node JS server as it's what he does as a part time job. In addition to this both developers are very familiar with the IDE which has been used by both of us since first year and we feel it's better to start with a tool we are comfortable with rather than grow to like a tool we don't yet know a lot about.

JavaScript Plugin for Visual Studio: https://code.google.com/p/js-addin/

NodeJS plugin for Visual Studio: http://nodejstools.codeplex.com/

Language & Libraries

We will be building our game with the Phaser framework with JavaScript. It's simple, elegant and has multiple render targets which means on devices that support it we can render the game in WebGL for additional performance. Additionally we will be using node js to write a simple server that stores high scores and a news list which will be visible on the main menu of the game.

Platform & Testing

We will be aiming to release the game on all platforms as HTML5 and Phaser gives us this advantage. We will be seeking to ensure that game runs well on both mobiles and tablets and have acquired suitable devices for which we will use for testing. We will also be testing them using the very helpful built in chrome emulation tools which we try to emulate what the game looks like on a wide range of devices.

Repository & Online Tools

Documentation & Repository

We have also set up Bitbucket repositories that we are using to maintain and update the game code and the server code. We will also be generating suitable documentation for the game using JSDoc both for the server and the game as well. The links down below are to the server repository and game repository respectively.

Server Repository: https://bitbucket.org/WilliamTaylor/mobilegamedev-server

Game Repository: https://bitbucket.org/WilliamTaylor/mobilegamedev

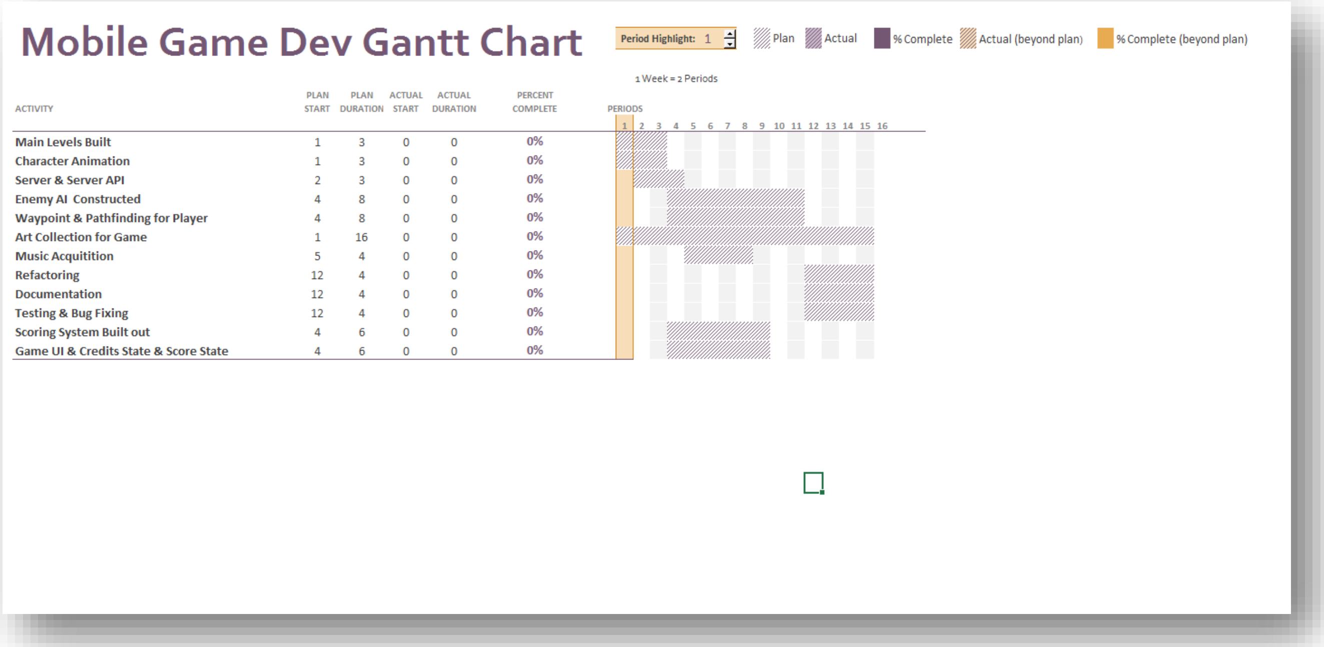
Formal Communication Tools

For development communication we will be utilizing Trello. We will be using this to generate checklist items to do as well as the general sharing of screenshots to ensure me and my project partner are on the same page.

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

Schedule / Gantt Chart

The following is our gantt chart for the development of the game. It will also be updated so we can show the official progression made when it comes to the project. Please note that we have taken a number of weeks to do some early work so we have a better idea of the task we have ahead of ourselves this includes playing around with the A* pathfinding algorithm and learning Phaser in depth so we know how to build the application.



Mobile Game Dev Gantt Chart

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Main Levels Built	1	3	0	0	0%
Character Animation	1	3	0	0	0%
Server & Server API	2	3	0	0	0%
Enemy AI Constructed	4	8	0	0	0%
Waypoint & Pathfinding for Player	4	8	0	0	0%
Art Collection for Game	1	16	0	0	0%
Music Acquitition	5	4	0	0	0%
Refactoring	12	4	0	0	0%
Documentation	12	4	0	0	0%
Testing & Bug Fixing	12	4	0	0	0%
Scoring System Built out	4	6	0	0	0%
Game UI & Credits State & Score State	4	6	0	0	0%