3110 Final Project Design Document

Caleb Mok, Talha Baig, Alan Gao, Newton Ni

November 15, 2017

1 System Description

1.1 Core Vision

1.2 Key Features

1.3 Narrative Description

2 System Design

2.1 HTTP

Our game will be designed as a client/server model, where the client communicates with the server via HTTP requests, using JSON to transfer data. We chose to design our system this way to facilitate multiple players connecting to the same game. As for division of labor, the client is responsible only for rendering the images on the user's screen; the server will handle all changes to the game state/model. The exact API can be found in the Apiary link.

To ensure that the game is synced for all players, clients will request the game state approximately 30 times a second, although the exact number is still uncertain.

3 Module Design

See interfaces.zip.

4 Data

The server will store the game model, which will contain the following fields and data:

```
player:
{
```

```
  uid: string;
  inventory: string list;     //This will be item ids
  location: int*int;
  health: int;
  direction: int OR variant    //N, E, S, W, NE, NW, SE, SW
}

item:
{
  iid: string;
  damage: int;
  projectile speed: int
  fire rate: int
}

state:
{
  players: player list;
  items: item list;
  elems: string*int*int;
  world boundaries: int*int;
}
```

When the server returns information to the client, we only return the fields/values that the client needs to render the GUI.

5 External Dependencies

6 Testing Plan