CMPUT 404 Lab 8

1-2nd March 2016

OVERVIEW

Learn how to utilize websockets and phaser. Create a basic phaser game with websocket connectivity for real-time server to client communicate.

STEPS

1. Check out the repository at https://github.com/awwong1/CMPUT404LAB8_W2016
2. Ensure that **npm** and **node** are installed and added to your classpath
3. Run the application from the checked out repo with **./bin/www**
4. Go to http://phaser.io and view some of the examples.
5. Navigate to http://phaser.io/examples/v2/tilemaps/csv-map-collide
6. Create a new folder inside **public** called **assets**
7. Download the three necessary assets files and place it inside the **public/assets** folder:
    a. https://github.com/photonstorm/phaser-examples/blob/master/examples/assets/tilemaps/csv/catastrophi_level2.csv
    b. https://github.com/photonstorm/phaser-examples/blob/master/examples/assets/tilemaps/tiles/catastrophi_tiles_16.png
    c. https://github.com/photonstorm/phaser-examples/blob/master/examples/assets/sprites/spaceman.png
8. Copy the code from this example and place it into the **public/javascripts/game.js** file https://github.com/photonstorm/phaser-examples/blob/master/examples/tilemaps/csv%20map%20collide.js

9. Rename the game constructor 4th argument from ‘phaser-example’ to ‘phaser’
    a. This matches the div id we specified in the jade template
10. Rename the preload sprite paths to match the specified path we download everything to.

```
var game = new Phaser.Game(800, 600, Phaser.CANVAS, 'phaser',
 { preload: preload, create: create, update: update, render: render }
);

function preload() {
 game.load.tilemap('map', 'assets/catastrophi_level2.csv', null,
Phaser.Tilemap.CSV);
 game.load.image('tiles', 'assets/catastrophi_tiles_16.png');
 game.load.spritesheet('player', 'assets/spaceman.png', 16, 16);
}
```

11. (Optional) Set the layer debug to be true to see collision
12. Run the application again using **./bin/www**
13. Stop the application. Add the websocket client code back to **public/javascripts/game.js** https://gist.github.com/awwong1/20b3acea02019f43a88f
14. Add the client websocket instantiation at the bottom of the create() method:

```
this.client = new Client();
this.client.openConnection();
```

15. Add in a uuid generator in the client. Create a new function with the following:. http://stackoverflow.com/questions/105034/create-guid-uuid-in-javascript

```
function guid() { function s4() { return Math.floor((1 +
Math.random()) * 0x10000) .toString(16) .substring(1); } return s4()
+ s4() + '-' + s4() + '-' + s4() + '-' + s4() + '-' + s4() + s4() +
s4(); }
```

16. Update the server side code to handle players instead of the one rabbit object https://gist.github.com/awwong1/90d50ffa41cfc5ef7ea4

```
var players = {};

wss.on('connection', function(ws) {
  ws.on('message', function(message) {
    var incommingMsg = JSON.parse(message);
    players[incommingMsg.uuid] = {x: incommingMsg.x, y:
```

```
incommingMsg.y};
    for(var i in wss.clients) {
      wss.clients[i].send(JSON.stringify(players));
    }
  });
  ws.send(JSON.stringify(players));
});
```

17. Set the variables of the class within **game.js** to equal the following:

```
var map;
var layer;
var cursors;
var players = {};
var id = guid();
players[id] = {};
var player = players[id];
```

18. Add the following code to the bottom of the update method:

```
  if (this.client.connected) {
    this.client.ws.send(JSON.stringify({uuid: id, x: player.x, y:
player.y}));
  }
```

19. Modify the client onMessage function to be equal to the following: https://gist.github.com/awwong1/2280e439b81c0fa666f7

```
Client.prototype.onMessage = function(message) {
  var msg = JSON.parse(message.data);
  for (var key in msg) {
    if (key in players) {
      players[key].x = msg[key].x;
      players[key].y = msg[key].y;
    } else {
      players[key] = game.add.sprite(48, 48, 'player', 1);
      players[key].x = msg[key].x;
      players[key].y = msg[key].y;
    }
  }
};
```

20. Run the application. Open up a new browser window and run the application. What happens?
21. (Bonus round) Get the websockets working with http://phaser.io/examples/v2/tilemaps/csv-map-with-p2

QUESTIONS

Question 1:

What is a websocket?

Question 2:

Why should websockets be used instead of client sided long-polling?

Question 3:

What is phaser (in the context of this lab)?

Question 4:

What is Jade (in the context of this lab)?

Question 5:

What is Express (in the context of this lab)?