CMPUT 404 Lab 8

October 26, 2016

Overview

- Learn how to utilize WebSockets and Phaser.
- Create a basic phaser game with WebSocket connectivity for real-time server to client communication.

Steps

1. Clone this repository.
2. In the root, run `npm install`
3. Run the application `./bin/www`
4. Go to http://phaser.io and view some examples
5. Navigate to http://phaser.io/examples/v2/tilemaps/csv-map-collide
6. Create a new folder instead `public` called `assets`.
7. Download the three necessary asset files and place them instead `public/assets`:
    - https://github.com/photonstorm/phaser-examples/raw/master/examples/assets/tilemaps/csv/catastrophi_level2.csv
    - https://github.com/photonstorm/phaser-examples/blob/master/examples/assets/tilemaps/tiles/catastrophi_tiles_16.png
    - https://github.com/photonstorm/phaser-examples/blob/master/examples/assets/sprites/spaceman.png
8. Copy the code from this example and place it into `public/javascript/game.js`:
    - https://github.com/photonstorm/phaser-examples/blob/master/examples/tilemaps/csv%20map%20collide.js
9. Rename `new Phaser.Game`'s 4th argument from `phaser-example` to `phaser`
    - To match the `<div>` ID we specified in the template.
10. Modify the paths in `preload()` to match the path we downloaded everything to—namely, `assets/` (the web framework removes the `public` part).
11. (optional) Uncomment `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 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/a/105074/6626414
16. Update the server side code to handle players instead of the one rabbit object: https://gist.github.com/awwong1/90d50ffa41cfc5ef7ea4

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 equal the following: https://gist.github.com/awwong1/2280e439b81c0fa666f7
20. Run the application. Open up a new browser window and run the application. What happens?
21. (Bonus) Get the application working with up/down/left/right animations

22. (Bonus) Get the WebSockets working with http://phaser.io/examples/v2/tilemaps/csv-map-with-p2

Questions

1. What is a WebSocket? Why were they created?
2. What is long-polling? Briefly explain what code you would need in the browser-side JavaScript and what code you would need the server-side to enable long-polling.
3. Why should WebSockets be used instead of long-polling?
4. What does the constructor of the `Client` class do?

5. WebSockets require callback methods. In the `openConnection()` method of the `Client`, these are set by using the following:

```
this.ws.onmessage = this.onMessage.bind(this).
this.ws.onerror = this.displayError.bind(this);
this.ws.onopen = this.connectionOpen.bind(this);
```

   Why is `Function#bind()` necessary? In other words, why couldn't the code just do this?

```
this.ws.onmessage = this.onMessage;
this.ws.onerror = this.displayError;
this.ws.onopen = this.connectionOpen;
```

6. What is Phaser (in the context of this lab)?