John Pascoe

15-112B

24 November 2015

Term Project Proposal

My proposal for my term project is to create a browser based multiplayer online battle arena (MOBA) game. This is a genre of game that has been popularized by the popular Warcraft 3 map “Defense of the Ancients,” commonly referred to as DOTA. Other popular games of this genre include League of Legends and Heroes of Newerth. I plan on having this game split up into a server / client pair, with the server calculating all game state based on the limited input of the client (setting move targets, performing special attacks), and sending that game state to the clients.

Many people want to be able to play games like League of Legends or DOTA at any computer, but because the game client is large, takes administrative privileges to install, and is incompatible with non-window/mac computers, they are unable to play. My term project will solve this problem by allowing anybody with a web browser to play a game similar to the popular League of Legends or DOTA. It will be just as simple as clicking a link, choosing your character, and hitting play.

For the server side of my term project, I plan on using web sockets for communication with the clients. There are many different web socket server implementations in python, but for now I’ve chosen to use Tornado. I also will use an image library like pypng in order to read map data like collision information since the server needs to do all the modifying of game state. I will also use the json module in order to serialize data and store config files. If necessary, at the end I may write out a proper packet specification for better network performance.

For the client side of my project, I plan on using the javascript and HTML5 canvases in order to render the game. Javascript comes with support for websockets by default. In order to make working with canvases easier, I will use EaselJS, which allows me to create containers, draw objects, and create a drawing loop very easily. I will use PreloadJS in order to load assets into javascript on the fly. I will use the library mousetrap to respond to keyboard events without worrying about keycodes. I may also end up using TweenJS to create smooth animations.