

CSC 130 Final Project

Mitchell Nursey

Project Details:

In this project we have created a demonstration project to showcase the features of WebSockets and how they can be used to collect live data from users and create real time interaction. For the demonstration we created a simple uncomplete multiplayer game in which a player can move and attack in a 2D top down environment. Using WASD and the left mouse button the player can completely control their character. Player's movements and actions are synchronized between all clients connected to the game server. The game server was written in python, it acts as a dummy host for all the clients to connect and relay information to each other.

Problem and Specification:

We created this project for two main reasons. First, we were interested in learning how to create real time asynchronous events and information sharing between web clients without using methods such as AJAX. Second, we also wanted to be able to collect live information from users such as mouse position, key clicks, users interaction on the page, window size, etc. This would allow us to easily collect information from users that could be analysed and studied. A reason for collecting this data would be to visualize how users use our website and applications, therefore we could increase user experience.

Design:

The data currently being sent to the server is the player position, rotation, and if the player is attacking or not. We have structured our data into JSON which is being modified and relayed to each client by the server. It would be trivial to send mouse clicks, key presses, and other useful information to our server but that is out of scope of this demonstration project.

Testing:

Our project has been tested on several browsers and different devices. The difficulty we found was the reliability of our project depends on the age of the browser it was tested on. Older browsers either did not have support for the HTML Canvas element or the JavaScript WebSockets. On the other hand, support was found for almost all modern up to date browsers. Unfortunately the world renown web browser Netscape does not support either WebSockets or the HTML Canvas. Our demo was created for desktop and laptop devices and has minimal mobile support due to game we created requiring a mouse and keyboard.

Initially we created our game server in Java, but that was a very poor decision as the implementation and bug testing was extremely time consuming. To remedy this issue we switched to Python, this helped significantly as we were able to deploy our server quickly with minimal lines of code.

To test the project open two web browser windows. Adjust the windows so each is visible on the monitor. Then go to web.uvic.ca/~mnursesey/attack/ in each window and move around your character in the game world. All character's movements should be synchronized across all browser windows. Use WASD to move, Left Mouse to attack, and the Mouse Position is the direction the character looks.

Documentation:

Our project has shown that it is easy to collect data and gather real time information from the web. This could be used for a plethora of applications such as data science, machine learning, web design, and video game development.

Taking this project further we could implement and create several diverse applications.

One avenue we could explore further is creating online games. Using a web browser as a platform for a game would allow us to target a broad audience without worrying too much about compatibility between systems. It would also allow our games to be lightweight and very accessible to users since they do not have to download or maintain any software. Diving into this route would pose significant challenges. Many things provided by preexisting game engine software we would have to implement ourselves. We would have to spend significant time and resources developing a code base that could easily be replaced by using tools such as Unity.

Another approach would be develop our data collection further. Being able to track and visualise how our users are using our applications and site is invaluable data. The data could provide information on how we should improve our application, what features are most used, and how to improve user interaction. An issue that might be problematic is securely handling and maintaining the data generated. The data would be anonymous but would still provide information such as IP addresses and anything the user enters while using our application.

Implementation:

To create the demo game we leveraged the HTML Canvas element and used several event listeners to track key presses. We created several JavaScript scripts each compartmentalized by their use. Getting the multiplayer aspect in our game to work required that we setup a server in Python for the JavaScript WebSockets to send and receive data from.

Demo: web.uvic.ca/~mnursesey/attack/

Github: github.com/mnursesey/attack-web-game