SEG 3904 Project Proposal

Project Title: Full-Stack Socket Programming Web Application

SEG 3904 – Innovation Research Project
Winter 2019
Supervisor: Professor Daniel Amyot – damyot@uottawa.com
Date: December 14, 2018.

Dominic Roy-Stang – droys011@uottawa.ca
Student #7483706

OVERVIEW

The purpose of this project is to learn about the process of full-stack development and socket programming by developing a simple multiplayer game that runs on the browser. The game would be a modified multiplayer version of the classic game *pong*. The project's focus is on becoming familiar with full-stack development in the context of real-time shared applications such as Google Docs. Hence, the web app will not use a game development framework, and will instead only rely on a minimalistic 2D physics engine library for the game scene. The application will use socket programming via the Socket.io library to handle the real-time events.

Learning Outcomes

My experience with web development is very limited. Hence, most of the technologies used will be new to me. By the end of this project, I will have learned the following:

- Back-end application development
- Front-end development with React
- Deploying a website that is publicly accessible and uses HTTPS
- Bidirectional stream communication with Socket.io and Express
- Principles of concurrency and networked physics
- Usage of a language that is not taught in courses at uOttawa: JavaScript
- Using docker in the context of web development
- Proper issue tracking, source control, and other general software engineering activities

Technologies

- **HTML** a standard markup language to create webpages
- **SASS** a stylesheet language that extends regular CSS
- **JavaScript** a common programming language for web applications
- **Node.js** a JavaScript runtime that permits running JavaScript for back end services
- **Express** a Node.js framework used for the creation of the back end server
- **Socket.io** a real-time, bidirectional and event-based communication system
- **Matter.js** a 2D rigid body JavaScript physics engine for the web
- **Docker** a tool to create and deploy applications using containers
- **Google Cloud Platform** a cloud provider for website hosting

Resources

1. https://gafferongames.com/post/introduction_to_networked_physics/
2. https://modernweb.com/building-multiplayer-games-with-node-js-and-socket-io/
3. https://scotch.io/@blizzerand11/docker-for-web-developers-all-you-need-to-know

DELIVERABLES

Deliverable	**Weight**
Initial application description and requirements.	15%
Hello World assessment of technologies.	15%
First version of software checked against use cases selected for first release.	15%
Inspection to determine the quality of the use of the programming language and patterns/frameworks.	15%
Working final version checked against both functional and non-functional requirements.	15%
Proper use of source control and issue tracking tools.	5%
Report with final requirements, architecture, challenges/resolutions, code structure overview, self-assessment of learning, and future work items.	20%

Test Aspects

I will be using *Jest* as the testing framework for the program. The following areas will be covered by tests in the test suite:

- **Socket Service**
    - Testing one to many concurrent connections.
    - Testing sending and receiving messages.
- **REST Service**
    - Testing properly formed requests to endpoints.
    - Testing malformed requests to endpoints.
- **Game State (networked physics)**
    - Testing the creation of multiple games when more than two connections are active.
    - Comparing client position and server position for the same rigidbody
    - Comparing game states across clients in the same game and the server
- **UI Testing**
    - Jest snapshots will be used to compare the visual appearance of pages with a reference image.

WORK PLAN (142 hours)

The following table serves as an overview of the work that will be accomplished every week. The number of hours listed includes time spent fixing the actions listed based on professor comments that will be given during meetings.

Week	**Meet?**	**Action**	**Hours**
1	Y	Project plan, expectations, technologies, and environment set-up	12
2	Y	Application description and requirements	12
3	Y	Hello World version with use of all technologies including GCP	24
4	Y	Development of a skeleton for the website and back-end	12
5	N	Implementation of a selected set of features/scenarios	15
6	N	Implementation of a selected set of features/scenarios	12
7	Y	Implementation of a selected set of features/scenarios [Version 1]	15
8	Y	Implementation of remaining features/scenarios	15
9	Y	Implementation of remaining features/scenarios & code cleanup	10
10	Y	Report draft and software inspection	10
11	Y	Report and final software delivered.	5