Project Report

# Objectives

The main objective of our project was to tie together various technologies which were new to us, in order to create a simple game application. We intentionally chose the simple game of tic-tac-toe so that we could focus on learning the new technologies rather than on implementation of difficult game logic.   
Research Problems Addressed

There technically was not a research problem addressed, as we were not trying to solve a problem which had never been solved before. Rather, we were addressing the educational goals of the individual team members: to gain more experience working on a SCRUM team, to explore several open source technologies, and to learn in a practical way how to apply some of the technologies discussed in class lectures.

# Approach

The project was attempted as a means of connecting various microservices to each other with JSON communication between them to accomplish a common goal.

# Methods for Evaluation of Results

The application was evaluated using mostly manual testing and automation testing where the cost of automation was low and the capacity was readily available. In the end we only automated testing on some of the Java code, as deadlines made the risk of untested code more acceptable.

# Architecture and Functional Components of Prototype System

Because we are using various webservers, to run our prototype system, we need to take several steps to fire up the various servers. We tried to standardize our install locations and database locations throughout our team so that we could all start up the system in the same way. We still ran into issues with different group members running different versions of our applications which meant we had some minor adjustments when moving code between computers using version control.

First, the MongoDb server needs to be started so that we have a database to store our player and game data. Due to the nature of MongoDb and its default configuration, no project specific configuration was needed to complete initial usability. If security became a concern in the project, MongoDb would need to be configured and secured.

Second, the node.js server needs to be started. This service provides the ability to send text-based messages to the other users of the application. The chat service operates using the socket.io library available in the node package manager. The chat service also uses the socket.io-cookie package for handling our client cookies. This works by having each client create a web socket connection to the chat service when loading the main webpage of the application. The service listens for messages to arrive then does a broadcast of the message to all known clients to display.

Third, Authentication provides a mean to determine the identity of the users of the application. It does this by the traditional practice of having each user provide a user name and password to the application to verify identity. Each user is assigned a token which is a randomly generated value that will continue to identify the user in future access attempts. A user’s user details can be queried from the service by using their token. The ability to look up a user by token is used by the other services to get the username of the client connecting.

Fourth, the GameServer tracks the state of games of tic-tac-toe. The GameServer provides a means to query the state of a given game of tic-tac-toe, retrieve a list of all games of tic-tac-toe, creation of games of tic-tac-toe, and the ability to submit a move to a tic-tac-toe game. The service provides a simple implementation of the rules of tic-tac-toe and will restrict move submission to follow those rules. Any user is able to query and submit moves for a game, however only the user whose turn the game is currently on will be able to successfully submit a move to advance the game.

Finally, the gateway application running in an Apache Tomcat Application Server provides a convenient browser-based user interface for accessing the other services. This application responds to http requests with html web pages and javascript utilities that provide a means for users to interact with a game of tic-tac-toe and other users through chat or challenges.

# Three Most Interesting Contributions of Project Design and/or Implementation

## Microservices

The utilization of a microservice architecture provided a level of complexity that we were not used to as a team. Which service was handling which aspect and when it was handling was not easy to follow or track as they each had their own domain of control. In the end it provided a nice environment as you could edit and restart individual applications and not wait for the whole project to restart if you only changed one service.

## WebGL

WebGL provided a lot of complexity that even after we got it “working” we still are not confident in our ability to make it do the sophisticated renderings that it is known to provide. It was an interesting experience to learn about perspective and how the camera works. Learning how the matrix math applied to rendering of shapes provided complexity that took significant time to figure out. This technology seems very useful as it offloads rendering work to the GPU instead of doing it on the CPU.

## node.js

When the group started this project we knew very little of JavaScript and node.js provided a means to learn more. We found that node.js provides a quick means for implementing services by utilizing its comprehensive package library. Although we only ended up using two packages, we explored a few others during development. We found that the framework of web sockets using emits and event listeners took a while to get a good grasp on, but they provide a means for continuous communication between client and server.

# What you have learned through the hand-on experience of doing this project

Starting off with a small minimum viable product scope and then expanding as resources allow, allowed the team to accomplish its initial objectives and then evaluate future objectives and if they fit into the scope of the project.

We picked a small set of technologies and specialized into them and then shared our learnings and guides with the rest of the team so that we all didn’t have to do all the work of investigating on our own.

Subway and only Subway is open in the Oakland Center every Saturday and Sunday since we were on campus doing this project. ☺

# What concepts and techniques you learned in class are used in the current project design

Servlets provided us a basis for the rest of the Java aspects of our program as they let us understand the underlying concepts of moving data over HTTP. JSPs provided us a framework for displaying information in a web browser in a stylized fashion.

We utilized microservices to separate concerns and functionality within the project space. This pattern is growing in popularity along with the notion of platform as a service. This is due to the ability to change individual components of a system and redeploy or scale them individually.

Node.js was covered in our panel discussion and so we had some familiarity with it already. Our small familiarity with it and the tutorials we found online provided us a good basis for implementing the functionality we wanted.

JSON provides a means of data exchange that avoids the overhead of definitions of XML and provides the fluidity to change on the fly and get the software to a functional state.

# What concepts and techniques you learned in class can be considered for extension of your current project

We wanted to apply semantic web techniques to our application but did not have sufficient time from when we had formalized our knowledge of semantic web to the end of the project’s timeline to realize our goals. We would have liked to have exported our database records of users and the games that they have played into an Ontology using FOAF’s to use a user’s email address as their unique identifier. This would provide a means for a social network or other application to track a user and the games of tic-tac-toe that they have played.