CptS 322 - Fall 2017

Term Project

Sean Kallaher, Hugh McGough, Connor Wright

I. Overview

For your term project in CptS322, you will build a web application for playing Diplomacy online. (see Section -2 for the project description). Alternatively, you may propose your own project idea (contingent on the approval of instructor; proposals due Wednesday Sep13, 11:59pm).

- You will be working in teams of 3 students. This will allow you to experience how to work as a team, and will also give you an idea of the issues that arise in software development groups. Each team will be able to choose the development languages, environments and tools that they want to use. You are welcome to use the same framework you used for your warm-up project.

- One member in your team should serve as the team liaison – not necessarily the leader, but will be responsible for the regular communications of your team with the course staff.

- Each team will work on refining the requirements, constructing the specification, designing the architecture and the individual modules, and of course the implementation and testing, both manual and automated.

- Your team will present the final project to the class at the end of the semester and will share the lessons they learned from the project.

- You will also demonstrate your final project to the instructor and the course TAs.

- Your individual performance will be peer-reviewed by your team mates.

- The project is structured in several stages, as described in Project Assignments (Section-IV).

II. Project Description

The board game *Diplomacy* is a fun one; however, it requires players to be in the same physical area.

A web based board game based on the classic board game *Diplomacy* (rules can be found at http://faculty.washington.edu/majeski/426/sim1.html).

You will build a web application that:

- Allows users to create an account for use on this website

- Allows users to join games

- Play the game *Diplomacy* and send messages to other players to coordinate moves.

- Allow users to check their statistics

Homepage

1. Outline for the rules of diplomacy
2. Option to log in
3. Option to register for an account
4. Option to join the queue for a game (Will redirect to login if not signed in)
5. Option to display user statistics

Login Page

1. Place for user to enter username
2. Place for user to enter password
3. Place for user to recover their password
4. Link to register for an account

Register Page

1. Prompt user for email
2. Prompt user for username
3. Prompt user for password

Game Page

1. Contains a map of the board populated with game pieces
2. User should be able to communicate with other players via text
3. Each game piece is clickable and will prompt user for what action should be taken (eg Attack, Defend, etc.)
4. After action is received further options will be available to choose from depending on what was chosen from before
5. The user should be able to submit all their orders to the server
6. The user should be able to revise their orders before the result of the moves are calculated
7. Game board is updated after resolution of all moves from previous round.
8. Repeat to 3 until game is over.

III. Data Models:

1. User information table (Statistics, username, email, etc.)
2. Encrypted login information

IV. Project Timeline:

i. Requirements document: 6%

1. Once we form the teams you will have to start right away getting the team working together and collecting the requirements. In this stage you only need to elaborate the details of the

requirements for the first iteration of your code, and include a summary of the requirements for future iterations.

2. You must start using GitHub to manage your plan.

3. Since there are only couple weeks after you form the team until the full Requirements and Specification Doc is due, it is important for the team to start meeting right away, because it takes time to figure out what the team will be building.

ii. Iterations: The main part of the project will be done in several two-week iterations. In each iteration you will have to work on several tasks

− Iteration 1: (12%)

− First draft of your Design Document (6%)

− Write a short progress report briefly summarizing the project progress and tests performed. This document should have links to your repo and deployed application.

− A completion of a running version of your code with a partial set of features. All code should be uploaded to GitHub repo before the iteration deadline. (4%)

− Testing support and tests for the implemented features. Every iteration should include unit tests and functional tests. All test code should be uploaded to GitHub repo before the iteration deadline. (2%)

− Iteration 2: (7%)

− Update your Design Document to reflect your current accomplishments and your plans for iteration 2 (1%)

− Write a short progress report briefly summarizing the project progress, tests performed, and code coverage analysis (starting iteration2). This document should have links to your repo and deployed application.

− A completion of a running version of your code with additional set of features. All code should be uploaded to GitHub repo before the iteration deadline. (4%)

− Testing support and tests for the implemented features. Every iteration should include unit tests and functional tests. Starting iteration 2 you should also include some automated tests. All test code should be uploaded to GitHub repo before the iteration deadline. (2%)

− Iteration 3: (8%)

− Update your Design Document to reflect your current accomplishments and your plans for iteration 3 (1%)

− Write a short progress report briefly summarizing the project progress, tests performed, and code coverage analysis. This document should have links to your repo and deployed application.

− A completion of a running version of your code with additional set of features. All code should be uploaded to GitHub repo before the iteration deadline. (4%)

− Testing support and tests for the implemented features. Every iteration should include unit tests and functional tests. You should also include some automated tests. All test code should be uploaded to GitHub repo before the iteration deadline. (2%)

− Final demo: (8%)

− At the end of the semester each team will demo their project to the instructor and the course TA.

− Presentation: (3%)

− During the last three lectures of the semester, each team will present 5-minute long presentations. You will demo your project so-far and teach your classmates about lessons learned. See Aection- for more details.

V. Project Deliverables:

1. Project Requirements Specifications:

There is no standard for requirements or specifications documents and in fact many organizations blend aspects of requirements, and specification in a single document. We will use a simple template for requirements and specification. Inevitably in preparing this document you will describe of how the system should interact with the outside world.

Please see the “Project Requirements Specifications” document template posted on Blackboard.

2. Design Document:

This will be a living document. For the first iteration you will fill in the document with the design details as you can see them before the first iteration. In subsequent iterations you will expand this document.

A template for the “Design Document” will be available on Blackboard.

3. Progress Report:

For each iteration, write a short progress report covering the topics specified below.

• Main Difficulties: What were the main difficulties so far? You should consider both technical and organization issues.

• Features: What features were completed? Were there any features you did not implement as planned? Are you pushing some features to later iterations, and if so, why?

• Tests: What tests did you prepare for this iteration, and what are they covering? What features are you not testing yet? Did you use any test frameworks?

o Note: Every iteration should include unit tests and functional tests

• Code Coverage: Use a code coverage tool to generate a code coverage report. Detailed instructions on how to perform code coverage will be provided.

• A template of the “Progress Report” will be available on Blackboard.

4. Your code and Tests:

In each iteration you need to commit a running version of your code along with your test code to the master branch of your repo. All code should be uploaded to GitHub repo before the iteration deadline. There should be a link to your repo in your progress report.

5. Presentation:

At the end of the semester, each team will make a presentation to the whole class in one of the lecture slots. The goal of these presentations is to allow the whole class to learn from the experience of all groups.

Each presentation should be delivered by the entire team. We will take attendance in these lectures. Everybody attending will fill in a feedback form that includes scores for the presentation. The presentation will be graded based on the course staff evaluation of how well prepared and useful the presentation is, along with the peer feedback submissions.

The presentation should be about 5 minutes long and should have 2 parts:

• An overview of your project, perhaps with a very short demo of the current state. The chance of a demo breaking increases with the number of people in the audience. We recommend recording the demo as a backup.

• Some of the lessons you have learned. Please do not be afraid to say what went wrong; being frank and drawing educational conclusions from your experience is what we are looking for.

You will have total 5 mins for your presentation. Please keep it short and up to the point.

VI. Submission Instructions:

All project documents including “Project Requirements Specifications”, “Design Document”, iteration “Progress Reports”, should be committed to the GitHub repo and uploaded to Blackboard.

All code and test case implementations should be committed to the GuitHub repo only.

Your GitHub repo should have the following branch structure :

− master:

o Final version of your code;

o (under “tests” directory) all your test code;

o (under documents directory) Project Requirements Specifications document; final version of your Design Document.

− iIterationX:

o All the code you produced in iteration X;

o (under “tests” directory) your test code for iteration X;

o (under documents directory) Version X of your Design Document; Progress Report for iteration X