Project Contract

**Student Name:**

Joshua Jackson

**P-number:**

p16179167

**Programme:**

Computer Games Programming

**Email address:**

p16179167@my365.dmu.ac.uk

**Project Title:**

Politico – Building a Cross Platform, Component Driven Game for the Modern Web

**Project Proposer:**

N/A (Myself)

**Supervisor:**

Dr. Arjab Singh Khuman (Archie Singh) – arjab.khuman@dmu.ac.uk

**Introduction (max. 100 words):**

You start off as the PM of the United Kingdom and throughout your tenure as Prime Minister you must make Yes/No decisions that will have either a positive or negative effect on the different attributes of your nation by varying degrees. Your goal is to keep those attributes (Financial, Population Happiness, Domestic & Foreign Political Favor) above 0 for the full length of your premiership. The main principle of the game is the balancing act involved with making a decision and factoring in the impact on the different attributes of your nation.

**Project Background (max. 300 words):**

The main driver for this project is to end up with a game that is fully component based and easy to reason about from a development perspective whilst being easy to pick up from a gameplay perspective. The game itself is simple as most of the actual “*gameplay*” will come from the user’s thought process and how they choose to navigate the balancing act of keeping different attributes of a nation running smoothly whilst essentially only ever interacting with the game in the form of yes/no answers. My desire to work on a project like this stems from my enjoyment of games such as Civilization or Diplomacy where, when broken down into the components that make them up, are very simple and I believe that is part of what makes them enjoyable to play.

The other motivating factor for this project is to build a cross platform game, I believe this can most easily be achieved with a game of this genre as well as through the use of web technologies as the web is easily the most accessible platform and especially with new developments such as Progressive Web Apps, an end-user may not even be able to tell the difference between native and the web, allowing our code to be truly *write once, run everywhere*.

**Aims (max. 100 words):**

* To deliver a cross platform, component driven game using modern web technologies such as React and TypeScript that can be played completely within a web browser (Chrome, Firefox, Edge) on any device.
* Make use of extensive Unit Testing for every feature that is implemented.
* Make use of core tenets of an agile development methodology such as Sprints, Epics and Retrospectives.
* Discuss the use of modern web technologies such as React and TypeScript and their benefit over using vanilla technologies such as JavaScript and the HTML DOM.
* Discuss the alternatives to the technologies I chose and why I decided to use what I am using.

**Objectives (max. 200 words):**

* Create a functioning strategy game for the web using React and TypeScript that can be played completely within a web browser
* Create a library of re-usable UI components
* Deliver a game that works on modern Desktops, Tablets and Mobile Devices all using a single codebase
* Create unit tests for every feature that must pass before any code can be committed to the git repository
* Run the project in phases of work that are made up of Sprints lasting a set amount of time with a list of features that should be implemented in that time.
* Run a Sprint Retrospective at the end of each sprint documenting what went well, what didn’t go well and what went okay but could be improved resulting in actions for the next sprint
* Document, using a culmination of all the sprint retrospectives, what I have learned, what development decisions didn’t go well and how I’d improve them in the future.
* Create a detailed Gantt Chart that documents the sprint cycles in relation to the project in order to keep track of all the stages the project went through.
* Implement a UI that allows the user to make yes/no decisions and to clearly see the impact their decision had on their nation’s attributes
* Implement a UI which is a map of the UK split into provinces/regions that allows the user to click on a province and see what that province contributes to their overall population happiness rating and which way the province leans on political issues.
* Create a documenting evaluating the competition in terms of technologies used and why they may or may not be better than what I decided to go with.

**Deliverables (max. 100 words):**

|  |  |  |
| --- | --- | --- |
| **Submission** | **Due Date** | **Deliverables** |
| Project Start Forms | 01/11/2019 (Week 5) | * Project Contract * Global Checklist * Ethical Review Form |
| First Deliverable (25%) | 10/01/2020 (Week 15) | * Literature Review * Functional Requirements * Indicative Test Plan * System Design Documentation * Implementation Report |
| Final Deliverable (75%) | 01/05/2020 (Week 31) | * Main Report (35%) * Viva (15%) * The System (40%) * Management Meetings (10%) |

**Resources and Constraints (max. 100 words):**

**Resources**

* A Windows or Mac computer capable of running the Visual Studio Code Text Editor and a web browser
* Access to the latest version of the React.js framework (<https://reactjs.org/>)
* Access to the latest version of the TypeScript compiler (<https://www.typescriptlang.org/>)
* Potential access to an open source pre-made SVG map of the UK
* Access to a wireframing/design tool such as Figma to design the user interface before implementation
* Access to tools that will allow me to deploy both the finished game and development version of the game to the web for public use. (Potentially static hosting such as Github Pages, or something like AWS, Netlify etc)
* The use of tools such as Ngrok (<https://ngrok.com/>) to allow me to test my localhost on different devices

**Constraints**

* Lost interest in games development and have re-focused that into web development. This may come back to bite me as I find making a game boring, although I hope that the heavy drive for web technologies might counteract this.
* The need to constantly work on personal projects outside of University to remain employable after university may take some of my time away from the final year project.
* The learning curve when working with new technologies could potentially be a constraint on the amount of time I have to complete the project.

**Sources of Information (max. 100 words):**

* I will make extensive use of the TypeScript documentation
* I will make use of Parliament data (<http://www.data.parliament.uk/>) to get information on political bills that have been voted on in the past in order to gain inspiration for list of decisions the user will have to make.
* I will make use of the research in this article (<https://yougov.co.uk/topics/politics/articles-reports/2019/08/13/left-wing-vs-right-wing-its-complicated>) to aid me in deciding what decisions affect which political leaning provinces and to what degree. (Getting a better idea of what is left-leaning and what is right-leaning)
* Potentially more resources from the internet (following research) that give inspiration for potential decisions that should be made within the game.
* Dr Archie Khuman (supervisor) in the event that I decide to explore Fuzzy Logic further as an application within my project
* Journals and Conference Proceedings for use within the Literature Review

**Risk Analysis (max. 100 words):**

* **The loss of data** – There is obviously the potential to lose my work. I plan on mitigating this by making constant use of source control both locally and at a remote location (Github) as well as weekly backups of my source control.
* **The game becoming too large in scope** – The main draw of the gameplay involves keeping track of dependencies, this obviously makes its way into development. I would need to be aware of the potential of the game to become too large in scope for me to complete in time. This is mitigated in part via the use of development sprints and sprint reviews but I will also aim to create a dependency tree/graph that is constantly monitoring each feature and making sure it doesn’t have too many dependencies that can spiral out of control.

**Schedule of Activities (max. 300 words):**

*\*The schedule of activities is subject to change and will be updated at the interim deadline once the direction of the project is clearer. - See attached Sprint plan for a more detailed breakdown of tasks for October, November and December.*

|  |  |
| --- | --- |
| **November** | * Setup project structure and all required build tools * Carry out initial research on development techniques that will be used * Setup a basic game core * Design the UI in all possible states within Figma * Implement the base Saving/Loading system * Implement all the documented attributes and functionality around increasing/decreasing them * Write relevant unit tests for all the functionality so far (ongoing process) * Run multiple sprint retrospectives to make the project aims/tasks clearer * Implement the very base UI (Start/End screens, On-screen attributes) * Implement a very rudimentary version of the decisions system |
| **December** | * Create a test plan for the first deliverable using knowledge gained from November * Gather all relevant research material for the Literature Review * Further implement the decision system * Implement the political leaning system in a basic form so that decisions can actually have an effect * Run multiple sprint retrospectives to make the project aims/tasks clearer * Create the Epics for the next few months with more detailed tasks given the knowledge from the previous two Epics (November/December) |
| **January** | * Complete all the documentation for the First Deliverable * Further implement the political leaning functionality with specific provinces having individual “viewpoints” and leanings to make decisions more dynamic * Implement the rest of the UI. (dummy UI where functionality doesn’t exist yet) * Tie all loose ends together and polish the functionality that has been implemented * Run some basic playtesting sessions on the base mechanics and gather feedback |
| **February** | * Carry out detailed testing following the test plan, including both human testing and unit tests. * Settle on all of the potential decisions that can come up within the game. * Through testing, ensure the game runs on all of the outlined platforms and scales correctly. * Ensure the game can be played fully in its current state (both win and lose) * Work on documentation * Balance the game wherever possible through local playtesting |
| **March** | * Polish the game wherever it feels unpolished. * Run an extensive playtesting session and gather feedback particularly on the gameplay itself rather than technical feedback. * Work on documentation * Balance the game where required based on feedback from playtesting |
| **April +** | * Add in none game-breaking features such as smooth UI animations * Finalise all documentation both technical and academic * Evaluate the game in its current form next to the initial objectives and, using all of the sprint retrospectives, document where the direction has changed and why it has changed. |

Student Signature: ***Digitally Signed***

Supervisor Signature: ***Digitally Signed***

Date: N/A