Browser Multiplayer Game

Supervisor: Lau Vincent
Member: Yip Long Yin 3035270745
## Table of Contents

1. **Introduction** ................................................................................................................................. 3
   1.1. Background ............................................................................................................................. 3
   1.2. Existing Product ....................................................................................................................... 4
2. **Objective** ......................................................................................................................................... 5
3. **Methodology** ............................................................................................................................... 6
   3.1. Client Side .............................................................................................................................. 6
   3.2. Server Side ............................................................................................................................. 6
   3.3. Production System ................................................................................................................ 6
4. **Feasibility Assessment** .................................................................................................................. 7
   4.1. Insufficient CPU and Memory ............................................................................................. 6
   4.2. Unexpected Errors and Bugs ............................................................................................... 6
   4.3. Out of Budget ....................................................................................................................... 6
5. **Schedule and Milestones** ........................................................................................................... 8
6. **Reference** ...................................................................................................................................... 9
1. Introduction

1.1 Background

There has been an increase in popularity for playing online games on the Internet. Gaming in both mobile and PC (personal computer) market has been growing rapidly and it is estimated to have reached an astonishing number of 2.6 billion audience (The UK Interactive Entertainment Association, 2017). The global gaming market is definitely a great opportunity.

Along with the growing trend of gaming, there is also a new type of games which can be played in browser, namely HTML5 games. Some of the popular games are Agar.io and Slither.io. For instance, “Agar.io” was the seventh search term in Google (Jon, 2016), which was slightly behind "Ronda Rousey" and "Paris". For both Agar.io and Slither.io, there were more than 300 million monthly players during their peak (Rank2Traffic, 2018) (Rank2Traffic, 2018), as well as countless of YouTube videos related to them having over 10 million views. It is clear that HTML5 games are being a trend and people enjoy playing them.

Apart from its huge popularity, HTML5 games are also great at being cross-platform and cross-device. In order words, the same HTML5 game can be played in your smart phones, Desktop or Laptop computers. The cost of maintenance and update is efficient compared to traditional games where a different version is needed for each device.
1.2 Existing Product

Currently, there are some existing browser games available in the market for children. Some of the most notable ones are Animal Jam, Adventure Quest Worlds and Mope.io.

Animal Jam is a browser RPG (Role-Playing Game) created by National Geographic as a fun and educational game for children to learn about animals, plants, habitats (National Geographic Kids, 2018). At the same time, children can meet new friends, socialize and chat in English (with filter of misspelled and bad words) in the game. There are also many mini-games available in Animal Jam for kids to play with, like chess and mysterious puzzles to solve. These features made Animal Jam one of the most popular online game among children and parents.

AdventureQuest Worlds is a children-friendly MMORPG (Massively Multiplayer Online Role-Playing Game) about hero fighting with monsters (AdventureQuest Worlds, 2018). The graphic style is 2D cartoonish. Although there are weapons and monsters in the game, it is completely kid-friendly as there are no blood nor violent scenes. The game has different storylines and quests, which allows children to have fun while learning English.

Mope.io, again, is an online game about animals targeting at children. The game is a huge, realistic simulation about food chain. Monkey eats bananas and Lion eats mice. The game allows their players to experience with different animals and survive in the nature. Players are required to make good use of their natural habitats such as lakes, swamp, forests, etc.
2. Objective

The project aims to develop a multiplayer browser game.

The game will be real-time multiplayer, which means players can see others’ actions with minimal delay. Players will be able to interact with each other in different ways. The server should be able to handle 100 concurrent users.

Users should be able to access the game with different browsers (e.g. Google Chrome, Safari) and screen size.

The user interface and game theme will be of 2D cartoonish graphic style as our target audience will be children of age between 5 and 13.

There will be two main focuses for the project. Educational and fun. In order to make the game impactful, we will make sure our game is attractive to children, our target audience. In addition, certain educational value should be delivered by our game.
3. Methodology

3.1. Client Side

This section will be responsible for presenting the game to the user, with the help of user interface, image and graphical representation of the game objects (e.g. monsters and terrains) and functions (e.g. Health bar).

A mix of different programming languages will be used to create the client side. The programming languages include: Javascript, HTML and CSS.

All of the game images, graphic and sound will also be in this section for optimal performance.

3.2. Server Side

This section will be responsible for most, if not all, of the calculations and algorithms of the game. Algorithms such as Collision Detection and Physics System will be implemented and executed here.

This is to ensure the correctness and uniformity of data, as putting all of the crucial calculation of data in one place will remove all sort of error resulted by delay.

A mix of different programming languages and tools will be used to create the server side, including Javascript, Node.js, Socket.io, Express and MongoDB.

We have chosen Javascript over other programming languages, such as C++, for building backend server. This is because Javascript is compatible with Node.js, which is easy, efficient and has many useful tools for building web applications like Socket.io and Express.

3.3. Production System

This section will be responsible for keeping the game available to the public. The production system’s job is to ensure the 100% uptime of the game, as well as the game running smoothly without problems.

The production system consists of a Virtual Private Server (Linux Ubuntu 18.04), with some running software including Nginx, Process Manager (PM2) and MongoDB.

Lastly, a domain name will also be included in this section, which is essential for the publishing of the project.
4. Feasibility Assessment

It is likely that a number of problems and challenges will occur during the project. This section is about the problems with which we are likely to encounter and the solution we suggest.

4.1. Insufficient CPU and Memory

Firstly, it is possible that our game server will not be able to handle a large amount of players and objects, resulting in the failure of the program.

We will allocate additional time for the optimization of algorithm in order to maximize the efficiency of code. In addition, the program will be inspected to ensure the success of the program.

4.2. Unexpected Errors and Bugs

Extra time will be assigned for debugging and fixing errors. Moreover, there will be a periodical backup for different stages of the project, to smoothen the debugging process.

4.3. Out of Budget

As the budget provided is limited (i.e. $1000), and the virtual machine alone cost $700, it is likely that a problem of insufficient budget will be resulted.

To solve the problem, we will be covering the additional cost of the project to make sure the project can be completed on schedule.
### 5. Schedule and Milestones

<table>
<thead>
<tr>
<th>Date</th>
<th>Deliverables of Phase 1 (Inception)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sept 30</td>
<td>• Detailed project plan</td>
</tr>
<tr>
<td></td>
<td>• Project web page</td>
</tr>
<tr>
<td>Oct</td>
<td>Conceptualize the details of the game</td>
</tr>
<tr>
<td></td>
<td>• Gameplay</td>
</tr>
<tr>
<td></td>
<td>• Mechanics</td>
</tr>
<tr>
<td>Nov</td>
<td>GUI and graphic design</td>
</tr>
<tr>
<td></td>
<td>• In-game User Interface</td>
</tr>
<tr>
<td></td>
<td>• Main Menu and Other Sections</td>
</tr>
<tr>
<td></td>
<td>• Object Images</td>
</tr>
<tr>
<td>Dec</td>
<td>Construction</td>
</tr>
<tr>
<td></td>
<td>• Input/Output System of Players</td>
</tr>
<tr>
<td></td>
<td>• Basic Game Mechanics (e.g. movement, chat, collision)</td>
</tr>
<tr>
<td></td>
<td>• Construction of Objects in-game (e.g. monsters, weapons)</td>
</tr>
<tr>
<td></td>
<td>• Addition of Sound Effect and Music</td>
</tr>
<tr>
<td></td>
<td>• Production System</td>
</tr>
<tr>
<td>Jan 20</td>
<td>Deliverables of Phase 2 (Elaboration)</td>
</tr>
<tr>
<td></td>
<td>• Preliminary implementation</td>
</tr>
<tr>
<td></td>
<td>• Detailed interim report</td>
</tr>
<tr>
<td>Feb-April</td>
<td>Optimization and Testing</td>
</tr>
<tr>
<td>April 14</td>
<td>Deliverables of Phase 3 (Construction)</td>
</tr>
<tr>
<td></td>
<td>• Finalized tested implementation</td>
</tr>
<tr>
<td></td>
<td>• Final report</td>
</tr>
</tbody>
</table>
6. Reference


