

## SE 3XA3: Requirements Document

Table 1: Revision History

Date	Developer	Change	Revision
September 29	Jiahao Li	Initial Draft	0
September 29	Pavithran Pathmarajah	Initial Draft	0
September 29	Viren Patel	Initial Draft	0

## Contents

<b>1 Project Drivers:</b>	<b>2</b>
1.1 Purpose of the Project . . . . .	2
1.1.1 The User Business or Background of the Project Effort: .	2
1.1.2 Goals of the Project: . . . . .	2
1.2 Stakeholders: . . . . .	2
1.2.1 Client: . . . . .	2
1.2.2 Customer: . . . . .	2
1.2.3 Other Stakeholders: . . . . .	2
1.2.4 Hands-On Users of the product: . . . . .	3
1.2.5 Personas: . . . . .	3
1.2.6 Priorities Assigned to Users: . . . . .	4
1.2.7 User Participation: . . . . .	4
1.2.8 Maintenance Users and Service Technicians: . . . . .	4
<b>2 Project Constraints:</b>	<b>4</b>
2.1 Mandate Constraints: . . . . .	4
2.1.1 Solution Constraints: . . . . .	4
2.1.2 Implementation Environment of the Current System: . . .	5
2.1.3 Partner and Collaborative Applications: . . . . .	5
2.1.4 Off-The-Shelf-Software: . . . . .	5
2.1.5 Anticipated Workspace Environment: . . . . .	5
2.1.6 Schedule Constraints: . . . . .	5
2.1.7 Budget Constraints: . . . . .	5
2.1.8 Enterprise Constraints: . . . . .	5
2.2 Naming Conventions and Terminology . . . . .	6
2.3 Relevant Facts and Assumptions: . . . . .	6
<b>3 The Scope of the Work</b>	<b>6</b>
3.1 Current Situtation . . . . .	6
3.2 Context of the work . . . . .	7
3.3 work partitioning . . . . .	7
<b>4 Business Data Model and Data Dictionary</b>	<b>7</b>
4.1 Business data Model . . . . .	7
4.2 Data Dictionary . . . . .	8
4.2.1 Artificial Intelligence . . . . .	8
4.2.2 Keyboard Input . . . . .	8
4.2.3 Tank . . . . .	8
4.2.4 Walls . . . . .	8
4.2.5 Base . . . . .	9
4.2.6 Tiles . . . . .	9
4.2.7 Projectiles . . . . .	9
4.2.8 End Game . . . . .	9

<b>5</b>	<b>Scope of the Product:</b>	<b>10</b>
5.1	Product Boundary . . . . .	10
5.2	Product use case Table . . . . .	12
5.3	Individual product use cases . . . . .	13
<b>6</b>	<b>Functional Requirments:</b>	<b>14</b>
<b>7</b>	<b>Non-Functional Requirements:</b>	<b>15</b>
7.1	Look and Feel Requirements: . . . . .	15
7.1.1	Appearance Requirements: . . . . .	15
7.1.2	Style Requirements: . . . . .	16
7.2	Usability and Humanity Requirements: . . . . .	16
7.2.1	Ease of use requirements: . . . . .	16
7.2.2	Personalization and Internationalization Requirements: . . . . .	16
7.2.3	Learning Requirements: . . . . .	16
7.2.4	Understandability and Politeness Requirements: . . . . .	16
7.2.5	Accessibility Requirements: . . . . .	16
7.3	Performance: . . . . .	16
7.4	Operational and Environmental Requirements: . . . . .	16
7.5	Maintainability and Support Requirements: . . . . .	17
7.6	Security Requirements: . . . . .	17
7.7	Cultural Requirements: . . . . .	17
7.8	Legal Requirements: . . . . .	17
<b>8</b>	<b>Project Issues:</b>	<b>17</b>
8.1	Open Issues: . . . . .	17
8.2	Off the shelf solutions: . . . . .	17
8.2.1	Ready Made: . . . . .	17
8.2.2	Reusable Components: . . . . .	18
8.2.3	Products that can be copied: . . . . .	18
8.3	New Problems: . . . . .	18
8.4	Tasks: . . . . .	18
8.5	Risks: . . . . .	18
8.6	Costs: . . . . .	18
8.7	User Documentation/Training: . . . . .	18
8.7.1	User Documentation Requirements: . . . . .	18
8.7.2	Training Requirements: . . . . .	18
8.8	Waiting Room: . . . . .	19

## 1 Project Drivers:

### 1.1 Purpose of the Project

#### 1.1.1 The User Business or Background of the Project Effort:

The objective of this project is to expand the accessibility options for the famous Tanks game. The game we will be working on is currently a stand-alone program which requiring a Java interface to compile or any other means of manual compilation. This can be an issue for a lot of users as they might not have the necessary knowledge or have compatibility problems with their platform. Our plan to transform this Java Program to a web-based application will rid of any issues and allow users to experience playing the game without any difficulties. What motivates us to go on with this project is the will to be a part of the game development community which is a power station providing a great number of options for people seeking entertainment through video games. This game is not a serious problem and is not necessarily a significant business opportunity for our host. This is because there are various versions of the Tanks game already available online for users at the present moment. This however is only theory based because the host popularity and traffic has a great impact on how significant this game is in terms of business.

#### 1.1.2 Goals of the Project:

The ultimate service goal we are trying to achieve is to make easy access of the Tanks game for the users. We will be using JavaScript as our programming language in order to make the game runnable on a website. We will then decide on a host that will best suite our product and help popularize it.

### 1.2 Stakeholders:

#### 1.2.1 Client:

The client for our product is the titleholder of the website which will be hosting our completed version of the Tanks game. Their role will be to review the final version of our product and share an interest in gaming entertainment.

#### 1.2.2 Customer:

Our customer will be the general public seeking gaming entertainment. The typical customer would have access to the Internet and a computer platform (laptops or desktops). Although our product is suitable for all age groups, children and teenagers are expected to make up the majority of our customers.

#### 1.2.3 Other Stakeholders:

Developers:

- JSTanks is the team of developers for this project. All developers will play a role in redeveloping the Tanks game; transforming it from a Java application to a website friendly game easily available to all potential customers.

#### Other Software Developers

- These are members of the general public, but what separates them from the rest is their interest in game/software development. This includes people who may be in our shoes; looking to redevelop an open source software or they could be complete beginners wanting to get into developing software using our product as a learning example.

#### 1.2.4 Hands-On Users of the product:

##### Teenagers and University / College Students

- Use games as a means of entertainment
- Novice in game development
- Masters of gaming media and its consumption
- Access to the product, majority age group 11 - 24

#### 1.2.5 Personas:

**Name:** Luke Cage

**Age:** 20

**Job:** Student

**Family:** Oliver Cage - Father, Felicity Cage - Mother

**Hobbies:** Educational Politics

**Residence:** North Bay, ON, Canada

**Favourite Food:** Poutine

**Favourite Music:** EDM ? Alan Walker

**Likes:** Tanks, Action, Flash Games

**Dislikes:** High tuition costs of post-secondary education

**Preferred Holiday:** Niagara Falls, Canada

**Attitude to Technology:** Positive

**Attitude to Money:** Extremely Positive

### **1.2.6 Priorities Assigned to Users:**

- Key Users: Client, Customers
- Secondary Users: other developers, general public
- Unimportant Users: general public

### **1.2.7 User Participation:**

No participation is required from any users

### **1.2.8 Maintenance Users and Service Technicians:**

JSTanks

The Developers Team will be responsible for maintaining and changing the source code for this product. They will also communicate with the client in order to ensure the website has considerate uptime.

## **2 Project Constraints:**

### **2.1 Mandate Constraints:**

#### **2.1.1 Solution Constraints:**

Description: The game shall be compatible and be able to run on Mozilla Firefox 49.0.1 and -Google Chrome 53.0.2785.143 browsers.

Rationale: The game will not be updated for any previous versions that might cause issues with the execution of the game client.

Fit Criterion: The game shall be available to users with the mentioned browser versions or any other versions which are compatible to run the game.

Description: The game shall only use HTML, CSS, and JavaScript for implementation and execution.

Rationale: The client will not use any other web related software and will have the correct versions of HTML, CSS, and JavaScript.

Fir Criterion: The users shall be able to run the game with no issues given they have the right version of JavaScript on their platform.

### 2.1.2 Implementation Environment of the Current System:



### 2.1.3 Partner and Collaborative Applications:

The product requires a text editor for the HTML, CSS, and JavaScript implementation to take place in. Because it is a web based software, it requires a browser with JavaScript and a server provided by the client. There are no other partner and collaborative applications for this product.

### 2.1.4 Off-The-Shelf-Software:

- Text editor i.e. Notepad++
- A web browser i.e. Mozilla Firefox, Google Chrome
- JavaScript for implementation

### 2.1.5 Anticipated Workspace Environment:

There is no specific working environments for this product, it can be used as long as there is a computer desktop or a laptop (portability) and an internet connection.

### 2.1.6 Schedule Constraints:

The given deadline for the finished product is December 8th, 2016. In addition, a proof of concept demonstration requires a minimum amount of implementation done to be presented on the week of October 17, 2016.

### 2.1.7 Budget Constraints:

There are no budget constraints to the development of this product. However, a goal imposed upon ourselves as a team is to use absolutely no money in the making of this product.

### 2.1.8 Enterprise Constraints:

There are no enterprises involved in the product development. This game is free of charge and requires nothing more than a computer platform and an internet connection.

## 2.2 Naming Conventions and Terminology

Acronym/Abbreviation	Meaning
JSTanks	Team Name
JSTanks	Project Name
JS	JavaScript
HTML	Hypertext mark up language
CSS	Cascading style sheets
git	Git Lab
API	Application program interface
GUI	Graphical user interface
AI	Artificial intelligence
PC	Personal computer

## 2.3 Relevant Facts and Assumptions:

The project is base on an existing application called tanks which is an open source with 684 lines of code. The aim is to convert such game from the local PC version to the website version using JS which allow users to enjoy the game without downloading.

The game is intend to be run on the web page, so we assume that modern computers which are capable of running latest browsers is available to users. Since the game take the keyboard as the input tool, a keyboard is also required to play the game.

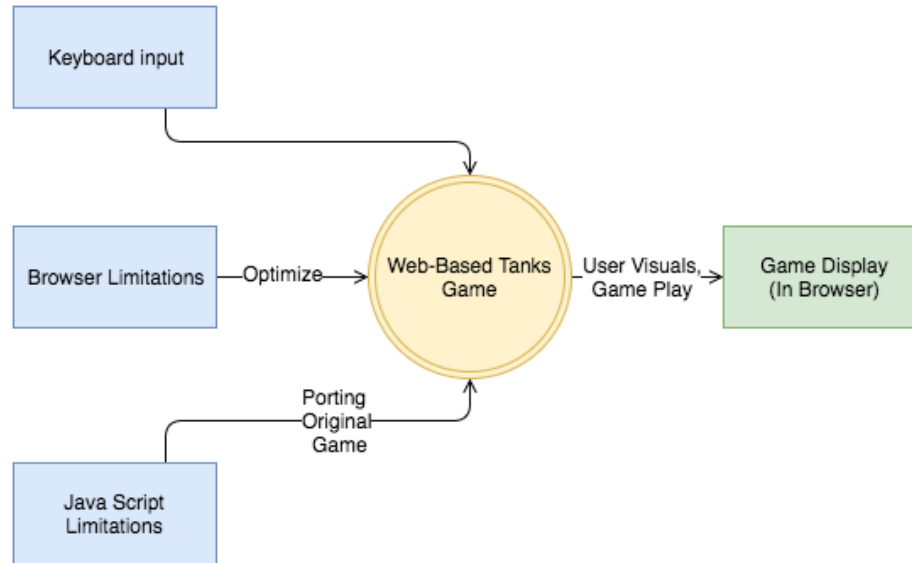
# 3 The Scope of the Work

## 3.1 Current Situtation

Currently computer systems based games run solely on the system itself and require the end-user to download and install dependencies for each game they would like to play. These games are designed for few systems and require particular specs to operate. The process that JS Tanks plan to modify is the above to make games platform independent and hardware independent.



### 3.2 Context of the work

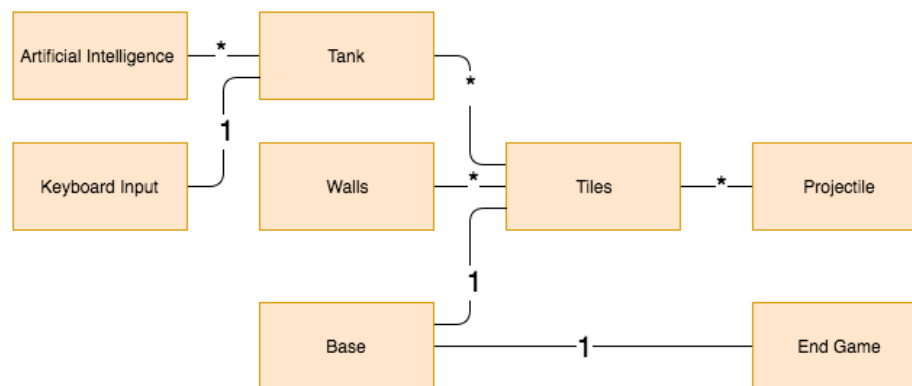


### 3.3 work partitioning

BUC Cases do not apply

## 4 Business Data Model and Data Dictionary

### 4.1 Business data Model



## **4.2 Data Dictionary**

### **4.2.1 Artificial Intelligence**

Attributes: Difficulty of AI

Relationships: Connected to the Tank class such that once it is initiated as a tank, location, health and direction can be tracked

Inputs: Tank locations, Player location, Base Location

Outputs: Tank directional movement, Tank Attacks

### **4.2.2 Keyboard Input**

Attributes: Current and last, inputs

Relationships: Connected to the Tank class such that once a player makes a move it can be passed onto there respective tank

Inputs: Keyboard input

Outputs: Tank directional movement, Tank attacks

### **4.2.3 Tank**

Attributes: Location, movement, attack

Relationships: Connected to the Tiles class to track if it can move or if there is a wall, base or tank in its way

Inputs: Directional Movement, Tiles beside the tank tile

Outputs: Moved to location

### **4.2.4 Walls**

Attributes: Location

Relationships: Connected to the Tiles class such that its health can be tracked and such that tanks can recognize there is a wall

Inputs: None

Outputs: None

#### **4.2.5 Base**

Attributes: Location

Relationships: Connected to the Tiles class such that its health can be tracked and such that tanks can recognize where the base is

Inputs: Destroyed

Outputs: End-Game

#### **4.2.6 Tiles**

Attributes: Health, track adjacent tiles

Relationships: Connected to all three tile types and acts as a game board

Inputs: moved locations, projectile attack on a tile

Outputs: Health, destroyed tile, adjacent tiles, launch attack

#### **4.2.7 Projectiles**

Attributes: Direction, strength

Relationships: Connected to tiles, such that tile health can be updated

Inputs: Attack launched in direction

Outputs: Tile hit by attack

#### **4.2.8 End Game**

Attributes: None

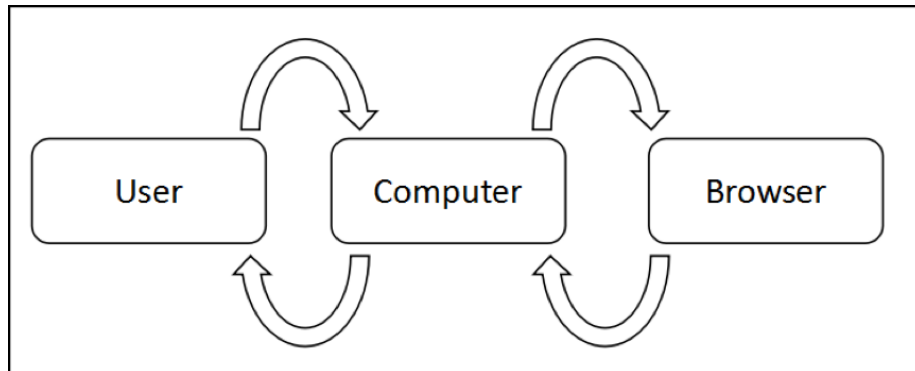
Relationships: Attached to Base which when destroyed ends the game

Inputs: Game Over

Outputs: Score and menu

## 5 Scope of the Product:

### 5.1 Product Boundary





## 5.2 Product use case Table

PUC NO	PUC Name	Actor	Input	Output
1	Start the new game	User	Click on the menu	Refreshing of the web page with the start of a new game
2	Pause the game	User	Click on the menu	The pause state of the game
3	Continue the game	User	Click on the menu	The activation of the game from the pause state
4	Choose the level	User	Click on the menu	Different speed mode of the game
5	See the introduction	User	Click on the menu	A window with introduction of the game in it
6	Quit the game	User	Click on the menu	The end state of the game
7	Move the tank	User	Press specified keys on the keyboard	The movement of the tank
8	Fire	User	Press specified keys on the keyboard	The launch of a bullet
9	Hit the enemy	User	Action of the tank operated by the user	The disappearance of the enemy
10	Hit the brick wall	User	Action of the tank operated by the user	The disappearances of both the bullet and the brick wall
11	Hit the steel wall	User	Action of the tank operated by the user	The disappearance of the bullet
12	Hit by the enemy	AI	Action of the tank operated by the AI	The decrement of the blood in the blood bar
13	Home base destroyed	AI	Action of the tank operated by the AI	The message of game over
14	Cross over the heart	User	Action of the tank operated by the user	The increment of the blood in the blood bar
15	Kill all enemies	User	Action of the tank operated by the user	The message of win

### 5.3 Individual product use cases

1. When the user clicks ?start a new game? in the menu, the whole web page gets refreshed which makes all stuff in the game back to their initial positions.
2. When the user clicks ?pause? in the menu, the game come into a pause state. All stuff in the game freeze and stay in their temporal position until the button ?continue? in the menu is clicked.
3. When the user clicks ?continue? in the menu, the game is activated from the pause state that all stuff run as the routine if it was the pause state before the click. And the click of ?continue? cause no effect if it was not the pause state before the click.
4. When the user click the different ?level? in the menu, the speed of tanks controlled by the AI changes in the game. The higher the level is, the higher the speed is.
5. When the user clicks ?introduction? in the menu, the web page pops up a window with the information of the game in it.
6. When the user clicks ?quit? in the menu, the game comes to the end state that the GUI turns into black with only the menu on it.
7. When the user presses the ?up?, ?down?, ?left? or ?right? key on the keyboard, the tank controlled by the user moves to the exact direction according to the key. If the user keep pressing the key, the tank keeps moving until it hits the wall or the boundary of the map.
8. When the user press the ?F? key on the keyboard, the tank controlled by the user launches a bullet. The bullet keeps moving and disappears when it hits another tank, a wall or the boundary of the map.
9. When the bullet launched by the user's tank hits another tank controlled by the AI, that tank and the bullet disappear all together at the same time.
10. When the bullet hits a brick wall, the wall and the bullet disappear all together at the same time.
11. When the bullet hits a steel wall, the bullet disappears and the steel wall keep existing in the same position.
12. When the tank controlled by the user or the home base is hit by the bullet which is launched by the tank controlled by the AI, one tenth of blood in the blood bar decreases.
13. When the blood bar is empty which means that the home base is destroyed, the web page pops up a window showing that the game is over.

14. When the tank controlled by the user cross over a heart, one-tenth of blood increases in the blood bar. The blood does not increase if the blood bar is full.
15. When the tank controlled by the user is the only tank in the map, the web page pops up a window showing that the user wins the game, and then the game come to the end state.

## 6 Functional Requirments:

- The executable HTML file shall create a new browser window.
- The HTML shall be executed by a browser with JavaScript functionality.
- The game shall have a standby state in which it waits for user input.
- The menu with four sections which are ?game?, ?pause/continue?, ?level?and ?introduction? shall shows up in the standby state.
- The sub menu of ?game?section which has choices of?start a new game?and ?quit?shall shows up when the ?game?section is clicked.
- The sub menu of ?pause/continue?section which has choices of?pause?and ?continue?shall shows up when the ?pause/continue?section is clicked.
- The sub menu of ?level?section which has choices of?level 1?, ?level 2?, and ?level 3?shall shows up when the ?level?section is clicked.
- The game shall be reset and start when ?start a new game?is clicked.
- The game shall come to the end state that the GUI turns into black with only the menu on it when ?quit?is clicked.
- the game shall come to the pause state when ?pause?is clicked.. All stuff in the game shall freeze and stay in the temporal positions in the pause state.
- All stuff frozen by the ?pause?button in the game shall be activated and back into the routine when ?continue?is clicked.
- The click of the?continue?button shall have no effect if the game is not in the pause state.
- The AI shall control tanks to move and fire randomly when the game starts.
- The moving speed of tanks controlled by the AI shall change when ?level 1? , ?level 2?or ?level 3?is clicked. The higher the level is, the higher the speed it.



- The level 1 shall be chosen as the default speed of tanks when the game starts.
- The window with the information of the game in it shall pop up when introduction is clicked.
- The tank controlled by the user shall move left, right, up or down when the left, right, up or down key on the keyboard is pressed. The tank shall keep moving when the key is kept pressing until the tank hits the wall or the boundary of the map.
- A bullet shall be launched and move along the direction that the tank faces to when the F on the keyboard is pressed. One bullet shall be launched every time that user press F.
- When the bullet launched by the user's tank hit another tank, that tank shall disappear with the bullet at the same time.
- When the bullet launched by the user's tank hit a brick wall, the brick wall shall disappear with the bullet at the same time.
- When the bullet launched by the user's tank hit a steel, the bullet shall disappear and the steel wall shall keep staying at the same position.
- When the bullet launched by the tank of the AI hits the user's tank or the home base, one-tenth of blood in the blood bar shall decrease.
- When the user's tank cross over a heart, one-tenth of blood shall increase in the blood bar. The blood shall not increase if the blood bar is full.
- A window shall pop up to show that the game is over and the game shall come to the end state if the blood bar is empty.
- A window shall pop up to show that the user wins the game and the game comes to the end state if the user's tank is the only one left in the map.

## 7 Non-Functional Requirements:

### 7.1 Look and Feel Requirements:

#### 7.1.1 Appearance Requirements:

The game shall acknowledge the client hosting it on a website by showing its credentials upon starting up. The game should be visually attractive and have appealing colour schemes. The game will have many tanks other than the user tank. The player tank will be coloured differently and all AI tanks will be coloured the same to make it easier for the player to tell the difference. There shall be brick sprites which make up the player's home base and graphics for bullets fired from all tanks.

### **7.1.2 Style Requirements:**

JS Tanks shall colour the tanks and the background so that they do not make it hard for the user to focus on the screen. The game shall have a menu with different options that the player can go through. i.e. Play, Instructions, Quit.

## **7.2 Usability and Humanity Requirements:**

### **7.2.1 Ease of use requirements:**

JS Tanks being a very simplistic game shall be very ease to play for all users. The interface as well as the controls shall be very basic, but at the same time, the gameplay should be satisfying for the player.

### **7.2.2 Personalization and Internationalization Requirements:**

This is not applicable to this game.

### **7.2.3 Learning Requirements:**

The user requires no further knowledge than to be able to open the browser.

### **7.2.4 Understandability and Politeness Requirements:**

The user should know simple English to surf through the menu and read instructions.

### **7.2.5 Accessibility Requirements:**

The user should be able to access the game from any platform with a web browser supporting JavaScript.

## **7.3 Performance:**

JS Tanks should be able to operate with minimal hardware specific requirements for the basis of the project is stray away from specific and high end hardware. The game should be able to operate with little input latency and no notable frame rates.

## **7.4 Operational and Environmental Requirements:**

The game should run on Chromium web-browsers and firefox browsers, across operating systems most notably Windows and OSX systems. The game should run within the web browser and require no connection to a server for processing.

### **7.5 Maintainability and Support Requirements:**

The code for the game should be simple and easy to understand and breakdown to help those maintaining the code or others whom plan to learn from it or build their own Java Script application

### **7.6 Security Requirements:**

The game should not access nor compromise user data, nor should it access or compromise the hosting entities data.

### **7.7 Cultural Requirements:**

JS Tanks shall be available only in English and contain no directly offensive remarks to any culture or nation.

### **7.8 Legal Requirements:**

The game must abide by Canada's Anti-Spam Legislation.

The game should try to meet the standards set by Canada's Accessibility Legislation.

## **8 Project Issues:**

### **8.1 Open Issues:**

Currently there are two issues plaguing our development of this project. Firstly the members of JSTanks are not particularly affluent in Java Script which is slowing down development as the team must learn as it develops. Secondly, deriving from the first issue the team is unsure of Java Script in its ability to create and handle objects, this issue will be looked into quickly; if Java Script is unable to create or handle objects then a more procedural approach will need to be followed.

### **8.2 Off the shelf solutions:**

#### **8.2.1 Ready Made:**

Ready made solutions which can be used are simple parsers to convert to Java Script but to do this, the java must first be parsed into another language for no direct Java to Java Script parser is available and by doing multiple language conversion efficiency can easily be lost.

### **8.2.2 Reusable Components:**

The components of the original project cannot be re-used due to the language and platform barrier, but the existing algorithms can be broken down and re-used but this is to be determined.

### **8.2.3 Products that can be copied:**

Currently there is no full product which may be duplicated but what can be copied are Java Script tutorials for specific functions which can be merged together to create an acceptable result.

## **8.3 New Problems:**

For Revision 1

## **8.4 Tasks:**

- Break down original project into workable components
- Convert Java components into algorithms and then into Java Script
- Merge components together to make fluent and efficient game

## **8.5 Risks:**

The only potential risk the development process might have is that we are learning new languages; HTML, CSS, and JavaScript. This learning period will take some space in our Development timeline and might under specific circumstances cause delays to the project schedule. This in turn, could compromise the quality of the final product. i.e. poor gameplay.

## **8.6 Costs:**

Not applicable as there are no set constraints for cost.

## **8.7 User Documentation/Training:**

### **8.7.1 User Documentation Requirements:**

Not Applicable.

### **8.7.2 Training Requirements:**

The game menu shall have an option giving a brief explanation of game and controls.

### **8.8 Waiting Room:**

- In-Game Audio
- Mobile edition
- Multiplayer