

# SE 3XA3: Test Report TankWar

Team #212, Genius  
Di Wu, 400117248, wud43  
Jiahao Zhou, 400082351, zhouj56  
Xinyu Huang, 400120376, huangx65

April 6, 2020

# Contents

<b>1</b>	<b>Functional Requirements Evaluation</b>	<b>1</b>
1.1	Area of Testing1 for Tank Movements . . . . .	1
1.2	Area of Testing2 for Tank Attacking . . . . .	6
1.3	Area of Testing3 for Tank Properties . . . . .	9
1.4	Area of Testing4 for Gaming Process . . . . .	10
1.5	Area of Testing5 for Map Editing . . . . .	14
1.6	Area of Testing6 for Game Rules . . . . .	18
<b>2</b>	<b>Nonfunctional Requirements Evaluation</b>	<b>30</b>
2.1	Look and Feel . . . . .	30
2.1.1	Visual Appearance Testing . . . . .	30
2.2	Usability and Humanity . . . . .	33
2.2.1	Ease of Use Testing . . . . .	33
2.2.2	Learning and Understandability Testing . . . . .	34
2.3	Performance . . . . .	35
2.3.1	Speed and Latency Testing . . . . .	35
2.3.2	Accuracy Testing . . . . .	36
2.3.3	Reliability and Availability Testing . . . . .	37
2.3.4	Robustness Testing . . . . .	38
2.3.5	Capacity Testing . . . . .	38
2.4	Operational and Environmental . . . . .	39
2.4.1	Physical Environment Testing . . . . .	39
2.4.2	Productization and Release Testing . . . . .	40
2.5	Maintainability and Support . . . . .	41
2.5.1	Supportability Testing . . . . .	41
2.6	Security . . . . .	42
2.6.1	Integrity and Privacy Testing . . . . .	42
2.7	Cultural . . . . .	43
2.7.1	Cultural Testing . . . . .	43
<b>3</b>	<b>Comparison to Existing Implementation</b>	<b>43</b>
<b>4</b>	<b>Unit Testing</b>	<b>43</b>
<b>5</b>	<b>Changes Due to Testing</b>	<b>48</b>
<b>6</b>	<b>Automated Testing</b>	<b>48</b>

<b>7</b>	<b>Trace to Requirements</b>	<b>49</b>
7.1	Traceability Between Test Cases and Requirements . . . . .	49
<b>8</b>	<b>Trace to Modules</b>	<b>53</b>
<b>9</b>	<b>Code Coverage Metrics</b>	<b>58</b>

## List of Tables

1	<b>Revision History</b> . . . . .	ii
2	Traceability Matrix for Functional Requirement . . . . .	50
3	Traceability Matrix for Non-functional Requirement . . . . .	51
4	Traceability Matrix for Unit Test Case . . . . .	52
5	Traceability Matrix for Functional Requirement . . . . .	55
6	Traceability Matrix for Non-functional Requirement . . . . .	56
7	Traceability Matrix for Unit Test Case . . . . .	57

## List of Figures

1	Coverage Test Screenshot . . . . .	59
---	------------------------------------	----

Table 1: **Revision History**

Date	Version	Notes
27 March 2020	1.0	Create the first version of the Test Plan
2020	1.1	Finish all the parts of the Test Report

This document provides a report that describes the results of the testing for the TankWar game. The test result of functional and nonfunctional requirements, and unit testing are presented in the file. The traceability of this test report to requirements and modules are showed. Note that all test cases conducted in this report are described in the Test Plan document. Please refer to the Test Plan for the details.

# 1 Functional Requirements Evaluation

## 1.1 Area of Testing1 for Tank Movements

There are four sections in Tank Movements: going up, going down, going left, and going right. The test cases of each sections are similar. Here is the test results for going up section.

### Test for Going Up

#### 1. MT-GU-1

Initial State: After the game starts.

Input: "W" key held by player 1.

Expected result: Player 1's tank goes up, which is shown on the screen.

Actual result: The screen shows the action that player 1's tank going up.

Result: pass

#### 2. MT-GU-2

Initial State: After the game starts.

Input: "W" key pressed by player 1.

Expected result: Player 1's tank goes up by 1 unit of length, which is shown on the screen.

Actual result: Player 1's tank goes up by 1 unit of length. The screen shows the action simultaneously.

Result: pass

### 3. MT-GU-3

Initial State: After the double-life tank is shot by a bullet and the ultimate skill is activated.

Input: "W" key held by player 1.

Expected result: Player 1's double-life tank still goes up, which is shown on the screen.

Actual result: Player 1's double-life tank still goes up. The screen shows the action simultaneously.

Result: pass

### 4. MT-GU-4

Initial State: After the double-life tank is shot by a bullet and the ultimate skill is activated.

Input: "W" key pressed by player 1.

Expected result: Player 1's double-life tank still goes up by 1 unit of length, which is shown on the screen.

Actual result: Player 1's double-life tank still goes up by 1 unit of length. The screen shows the action simultaneously.

Result: pass

### 5. MT-GU-5

Initial State: After the game starts.

Input: Upward Arrow key held by player 2.

Expected result: Player 2's tank goes up, which is shown on the screen.

Actual result: Player 2's tank goes up. The screen shows the action simultaneously.

Result: pass

### 6. MT-GU-6

Initial State: After the game starts.

Input: Upward arrow key pressed by player 2.

Expected result: Player 2's tank goes up by 1 unit of length, which is shown on the screen.

Actual result: Player 1's tank goes up by 1 unit of length. The screen shows the action simultaneously.

Result: pass

#### 7. MT-GU-7

Initial State: After the double-life tank is shot by a bullet and the ultimate skill is activated.

Input: Upward arrow key held by player 2.

Expected result: Player 2's double-life tank still goes up, which is shown on the screen.

Actual result: Player 2's double-life tank still goes up. The screen shows the action simultaneously.

Result: pass

#### 8. MT-GU-8

Initial State: After the double-life tank is shot by a bullet and the ultimate skill is activated.

Input: Upward arrow key pressed by player 2.

Expected result: Player 2's double-life tank still goes up by 1 unit of length, which is shown on the screen.

Actual result: Player 2's double-life tank still goes up by 1 unit of length. The screen shows the action simultaneously.

Result: pass

The rest of the test cases are also completed and get the expected results. The techniques of testing the rest three sections are similar to the Going Up section above, which is to test player 1's operation keys and player 2's operation keys separately. Also the moving ability of double-life tank is tested after getting shot by one bullet.

## Test for Walls and Boundaries

### 1. MT-WB-1

Initial State: After the game starts.

Input: Player 1's tank is facing to a brick wall and there is no distance between the tank and the wall. Player 1 presses the corresponding key from "WASD" keys in order to make the tank move towards the brick wall.

Expected result: Player 1's tank does not move.

Actual result: Player 1's tank stops.

Result: pass

### 2. MT-WB-2

Initial State: After the game starts.

Input: Player 1's tank is facing to a iron wall and there is no distance between the tank and the wall. Player 1 presses the corresponding key from "WASD" keys in order to make the tank move towards the iron wall.

Expected result: Player 1's tank does not move.

Actual result: Player 1's tank stops.

Result: pass

### 3. MT-WB-3

Initial State: After the game starts.

Input: Player 2's tank is facing to a brick wall and there is no distance between the tank and the wall. Player 2 presses the corresponding key from arrow keys in order to make the tank move towards the brick wall.

Expected result: Player 2's tank does not move.

Actual result: Player 2's tank stops.

Result: pass

#### 4. MT-WB-4

Initial State: After the game starts.

Input: Player 2's tank is facing to a iron wall and there is no distance between the tank and the wall. Player 2 presses the corresponding key from arrow keys in order to make the tank move towards the iron wall.

Expected result: Player 2's tank does not move.

Actual result: Player 2's tank stops.

Result: pass

#### 5. MT-WB-5

Initial State: After the game starts.

Input: Player 2's tank is facing to the boundary of the map and there is no distance between the tank and the boundary. Player 2 presses the corresponding key from arrow keys in order to make the tank move towards the boundary.

Expected result: Player 2's tank does not move.

Actual result: Player 2's tank stops.

Result: pass

#### 6. MT-WB-6

Initial State: After the game starts.

Input: Player 1's tank is facing to the boundary of the map and there is no distance between the tank and the boundary. Player 1 presses the corresponding key from "WASD" keys in order to make the tank move towards the boundary.

Expected result: Player 1's tank does not move.

Actual result: Player 1's tank stops.

Result: pass



## 1.2 Area of Testing<sup>2</sup> for Tank Attacking

### Test for Shooting Bullets

#### 1. TAT-SB-1

Initial State: After the game starts.

Input: "J" key pressed by player 1.

Expected result: Player 1's tank shoots a bullet.

Actual result: Player 1's tank shoots a bullet.

Result: pass

#### 2. TAT-SB-2

Initial State: After the game starts.

Input: "J" key held by player 1.

Expected result: Player 1's tank shoots bullets continuously.

Actual result: Player 1's tank shoots bullets continuously.

Result: pass

#### 3. TAT-SB-3

Initial State: After the game starts.

Input: ",," key pressed by player 2.

Expected result: Player 2's tank shoots a bullet.

Actual result: Player 2's tank shoots a bullet.

Result: pass

#### 4. TAT-SB-4

Initial State: After the game starts.

Input: ",," key held by player 2.

Expected result: Player 2's tank shoots bullets continuously.

Actual result: Player 2's tank shoots bullets continuously.

Result: pass

5. TAT-SB-5

Initial State: After the double-life tank get shot for the first time.

Input: ",," key held by player 2.

Expected result: Player 2's double-life tank still shoots bullets continuously.

Actual result: Player 2's double-life tank still shoots bullets continuously.

Result: pass

6. TAT-SB-6

Initial State: After the double-life tank get shot for the first time.

Input: "J" key held by player 1.

Expected result: Player 1's double-life tank still shoots bullets continuously.

Actual result: Player 1's double-life tank still shoots bullets continuously.

Result: pass

7. TAT-SB-7

Initial State: After the double-life tank get shot for the first time.

Input: ",," key pressed by player 2.

Expected result: Player 2's double-life tank still shoots a bullet.

Actual result: Player 2's double-life tank still shoots a bullet.

Result: pass

8. TAT-SB-8

Initial State: After the double-life tank get shot for the first time.

Input: "J" key pressed by player 1.

Expected result: Player 1's double-life tank still shoots a bullet.

Actual result: Player 1's double-life tank still shoots a bullet.

Result: pass

### **Test for Using Ultimate Skills**

#### **1. TAT-UUS-1**

Initial State: After the game starts.

Input: "K" key pressed by player 1.

Expected result: Player 1's tank use ultimate skills.

Actual result: Player 1's tank use ultimate skills.

Result: pass

#### **2. TAT-UUS-2**

Initial State: After the game starts.

Input: "." key pressed by player 2.

Expected result: Player 2's tank use ultimate skills.

Actual result: Player 2's tank use ultimate skills.

Result: pass

#### **3. TAT-UUS-3**

Initial State: After the game starts.

Input: "K" key held by player 1.

Expected result: Player 1's tank use ultimate skills continuously.

Actual result: Player 1's tank use ultimate skills continuously.

Result: pass

#### 4. TAT-UUS-4

Initial State: After the game starts.

Input: "." key held by player 2.

Expected result: Player 2's tank use ultimate skills continuously.

Actual result: Player 2's tank use ultimate skills continuously.

Result: pass

### 1.3 Area of Testing<sup>3</sup> for Tank Properties

#### Test for Double-Life Tank

##### 1. TPT-DLT-1

Initial State: After the game starts.

Input: Player chooses double-life tank before entering the game and the player presses "K" key or "." key to activate the ultimate skill.

Expected result: The double-life tank is still alive when the ultimate skill is activated and the tank gets shot simultaneously.

Actual result: The double-life tank is still alive.

Result: pass

#### Test for High-Speed Tank

##### 1. TPT-HST-1

Initial State: After the game starts.

Input: Player chooses high-speed tank before entering the game and the player presses "K" key or "." key to activate the ultimate skill.

Expected result: During the tank movement, the tank rushes for 1 seconds after the ultimate skill is activated. The direction can be changed by the player during the rushing process by pressing another key on the keyboard.

Actual result: The ultimate skill only available for 1 second. During the time, the speed of tank becomes faster and direction can be changed during rushing time. After 1 second, the speed of tank returns normal.

Result: pass

### **Test for Double-Bullet Tank**

#### **1. TPT-DBT-1**

Initial State: After the game starts.

Input: Player chooses double-bullet tank before entering the game and the player presses "K" key or "." key on the keyboard to activate the ultimate skill.

Expected result: During attacking, the double-bullet tank shoots two bullets simultaneously when the ultimate skill activates.

Actual result: The double-bullet tank shoots two bullets simultaneously, one with a faster speed and another with a slower speed.

Result: pass

## **1.4 Area of Testing4 for Gaming Process**

### **Test for PVP Mode Selection**

#### **1. GPT-PPMS-1**

Initial State: Player 1 opens the software.

Input: Player 1 selects PVP mode using ENTER key as the button of confirming from the keyboard.

Expected result: After opening the software, there should be three modes for player to choose. If the player chooses PVP mode, there should be the interface for PVP tank selection shown on the screen.

Actual result: After pressing the Enter key, the interface of tank selection is shown.

Result: pass

### **Test for PVE Mode Selection**

#### **1. GPT-PEMS-1**

Initial State: Player 1 opens the software.

Input: Player 1 selects PVE mode using Enter key as the button of confirming from the keyboard.

Expected result: After opening the software, there should be three modes for player to choose. If the player chooses PVE mode, there should be the interface for PVE tank selection shown on the screen.

Actual result: After pressing the Enter key, the interface of tank selection is shown.

Result: pass

### **Test for Map Editing Mode Selection**

#### **1. GPT-MEMS-1**

Initial State: Player 1 opens the software.

Input: Player 1 selects map editing mode using Enter key as the button of confirming from the keyboard.

Expected result: After opening the software, there should be three modes for player 1 to choose. If the player 1 chooses map editing mode, there should be the interface for map selection shown on the screen.

Actual result: After pressing the Enter key, the interface of choosing map is shown.

Result: pass

### **Test for PVP Map Loading**

#### **1. GPT-PPML-1**

Initial State: Player 1 opens the software and selects PVP mode using Enter key as the button of confirming from the keyboard.

Input: Player 1 selects one of the PVP maps in the PVP map selecting section.

Expected result: The selected PVP map shows on the screen.

Actual result: The selected PVP map shows on the screen.

Result: pass

### **Test for PVE Map Loading**

#### **1. GPT-PEML-1**

Initial State: Player 1 opens the software and selects PVE mode using Enter key as the button of confirming from the keyboard.

Input: Player 1 selects one of the PVE maps in the PVE map selecting section.

Expected result: The selected PVE map shows on the screen.

Actual result: The selected PVE map shows on the screen.

Result: pass

### **Test for PVP Tank Selection**

#### **1. GPT-PPTS-1**

Initial State: Player 1 opens the software, selects PVP mode by using Enter key as the button of confirming from the keyboard.

Input: Player 1 chooses one from three different kinds of tanks by using Enter key as the button of confirming from the keyboard.

Expected result: The corresponding tank shows on the selected PVP map.

Actual result: The corresponding tank shows on the selected PVP map.

Result: pass

#### **2. GPT-PPTS-2**

Initial State: Player 1 opens the software, selects PVP mode by using Enter key as the button of confirming from the keyboard.

Input: Player 2 chooses one from three different kinds of tanks using Enter key as the button of confirming from the keyboard.

Expected result: The corresponding tank shows on the selected PVP map.

Actual result: The corresponding tank shows on the selected PVP map.

Result: pass

### **Test for PVE Tank Selection**

#### **1. GPT-PETS-1**

Initial State: Player 1 opens the software, selects PVE mode by using Enter key as the button of confirming from the keyboard.

Input: Player 1 chooses one from three different kinds of tanks by using Enter key as the button of confirming from the keyboard.

Expected result: The corresponding tank shows on the selected PVE map.

Actual result: The corresponding tank shows on the selected PVE map.

Result: pass

#### **2. GPT-PETS-2**

Initial State: Player 1 opens the software, selects PVE mode by using Enter key as the button of confirming from the keyboard.

Input: Player 2 chooses one from three different kinds of tanks using Enter key as the button of confirming from the keyboard.

Expected result: The corresponding tank shows on the selected PVE map.

Actual result: The corresponding tank shows on the selected PVE map.

Result: pass



## 1.5 Area of Testing<sup>5</sup> for Map Editing

### Test for Adding Brick Wall for PVP Map

#### 1. ME-PVPAB-1

Initial State: The map editing for PVP mode is running, and there is empty space in front of the player's tank.

Input: "J" key pressed.

Expected result: A brick wall is added right in front of the player's tank.

Actual result: A brick wall is added right in front of the player's tank.

Result: pass

#### 2. ME-PVPAB-2

Initial State: The map editing for PVP mode is running, and the player's tank is facing to a brick wall.

Input: "J" key pressed.

Expected result: Nothing is changed.

Actual result: The map editing for PVP mode will be run and the player's tank will be moved to face to and with no distance with a existing brick wall, then the "K" key will be pressed and check if nothing is changed.

Result: pass

#### 3. ME-PVPAB-3

Initial State: The map editing for PVP mode is running, and the player's tank is facing to a iron wall.

Input: "J" key pressed.

Expected result: Nothing is changed.

Actual result: Nothing is changed.

Result: pass

#### 4. ME-PVPAB-4

Initial State: The map editing for PVP mode is running, and the player's tank is facing the boundary of the map.

Input: "J" key pressed.

Expected result: Nothing is changed.

Actual result: Nothing is changed.

Result: pass

### **Test for Adding Iron Wall for PVP Map**

#### 1. ME-PVPAI-1

Initial State: The map editing for PVP mode is running, and there is empty space in front of the player's tank.

Input: "K" key pressed.

Expected result: A iron wall is added right in front of the player's tank.

Actual result: A iron wall is added right in front of the player's tank.

Result: pass

#### 2. ME-PVPAI-2

Initial State: The map editing for PVP mode is running, and the player's tank is facing to a brick wall.

Input: "K" key pressed.

Expected result: Nothing is changed.

Actual result: Nothing is changed.

Result: pass

#### 3. ME-PVPAI-3

Initial State: The map editing for PVP mode is running, and the player's tank is facing to a iron wall.

Input: "K" key pressed.

Expected result: Nothing is changed.

Actual result: Nothing is changed.

Result: pass

#### 4. ME-PVPAI-4

Initial State: The map editing for PVP mode is running, and the player's tank is facing to the boundary of the map.

Input: "K" key pressed

Expected result: Nothing is changed.

Actual result: Nothing is changed.

Result: pass

### **Test for Deleting Wall for PVP Map**

#### 1. ME-PVPDW-1

Initial State: The map editing for PVP mode is running, and the player's tank is aim at a brick wall.

Input: "L" key pressed

Expected result: The brick wall is deleted from the map.

Actual result: The brick wall is deleted from the map.

Result: pass

#### 2. ME-PVPDW-2

Initial State: The map editing for PVP mode is running, and the player's tank is aim at a iron wall.

Input: "L" key pressed

Expected result: The iron wall is deleted from the map.

Actual result: The iron wall is deleted from the map.

Result: pass

### 3. ME-PVPDW-3

Initial State: The map editing for PVP mode is running, and the player's tank is aim at nothing.

Input: "L" key pressed

Expected result: Nothing is changed in the map.

Actual result: Nothing is changed in the map.

Result: pass

#### **Test for Adding Brick Wall for PVE Map**

For PVE map, the testing strategy is same as PVP map. Firstly, the ability of adding brick should be tested, including the situations about adding the brick on a free space, adding the brick on the area with another brick or iron, and adding the brick on the boundary of map. Secondly, the ability of adding iron should be tested, including the situations about adding the iron on a free space, adding the iron on the area with another brick or iron, and adding the iron on the boundary of map. Lastly, deleting ability should be tested, including deleting a brick wall, deleting a iron wall, and pressing "L" facing a free space. The results for those test cases in PVE map are same as expected.

#### **Test for Saving Maps for PVP**

##### 1. ME-PVPSM-1

Initial State: The player is in map editing mode and a map is created.

Input: Press the "Enter" key, then enter a valid file name and press "Enter" key again.

Expected result: A file stands for the map created is shown in the folder for PVP maps.

Actual result: A file stands for the map created is shown in the folder for PVP maps.

Result: pass

## **Test for Saving Maps for PVE**

### **1. ME-PVESM-1**

Initial State: The player is in map editing mode and a map is created.

Input: Press the "Enter" key, then select the existing files and press "Enter" key again.

Expected result: A file stands for the map created is shown in the folder for PVE maps.

Actual result: A file stands for the map created is shown in the folder for PVE maps.

Result: pass

## **1.6 Area of Testing<sup>6</sup> for Game Rules**

### **Test for General Rules about the Wall**

#### **1. GR-GRW-1**

Initial State: After the PVP game starts.

Input: Player 1 presses "J" key from the keyboard to shoot a bullet towards the brick wall and the bullet hits the brick wall.

Expected result: The brick wall is destroyed by the bullet. The bullet and the brick wall disappear simultaneously.

Actual result: The brick wall is destroyed by the bullet. The bullet and the brick wall disappear simultaneously.

Result: pass

#### **2. GR-GRW-2**

Initial State: After the PVE game starts.

Input: Player 1 presses "J" key from the keyboard to shoot a bullet towards the brick wall and the bullet hits the brick wall.

Expected result: The brick wall is destroyed by the bullet. The bullet and the brick wall disappear simultaneously.

Actual result: The brick wall is destroyed by the bullet. The bullet and the brick wall disappear simultaneously.

Result: pass

### 3. GR-GRW-3

Initial State: After the PVE game starts.

Input: Player 2 presses "," key from the keyboard to shoot a bullet towards the brick wall and the bullet hits the brick wall.

Expected result: The brick wall is destroyed by the bullet. The bullet and the brick wall disappear simultaneously.

Actual result: The brick wall is destroyed by the bullet. The bullet and the brick wall disappear simultaneously.

Result: pass

### 4. GR-GRW-4

Initial State: After the PVP game starts.

Input: Player 2 presses "," key from the keyboard to shoot a bullet towards the brick wall and the bullet hits the brick wall.

Expected result: The brick wall is destroyed by the bullet. The bullet and the brick wall disappear simultaneously.

Actual result: The brick wall is destroyed by the bullet. The bullet and the brick wall disappear simultaneously.

Result: pass

### 5. GR-GRW-5

Initial State: After the PVP game starts.

Input: Player 1 presses "J" key from the keyboard to shoot a bullet towards the iron wall and the bullet hits the iron wall.

Expected result: The iron wall is not destroyed by the bullet. The bullet disappear.

Actual result: The iron wall is not destroyed by the bullet. The bullet disappear.

Result: pass

#### 6. GR-GRW-6

Initial State: After the PVE game starts.

Input: Player 1 presses "J" key from the keyboard to shoot a bullet towards the iron wall and the bullet hits the iron wall.

Expected result: The iron wall is not destroyed by the bullet. The bullet disappear.

Actual result: The iron wall is not destroyed by the bullet. The bullet disappear.

Result: pass

#### 7. GR-GRW-7

Initial State: After the PVE game starts.

Input: Player 2 presses "," key from the keyboard to shoot a bullet towards the iron wall and the bullet hits the iron wall.

Expected result: The iron wall is not destroyed by the bullet. The bullet disappear.

Actual result: The iron wall is not destroyed by the bullet. The bullet disappear.

Result: pass

#### 8. GR-GRW-8

Initial State: After the PVP game starts.

Input: Player 2 presses "," key from the keyboard to shoot a bullet towards the iron wall and the bullet hits the iron wall.

Expected result: The iron wall is not destroyed by the bullet. The bullet disappear.

Actual result: The iron wall is not destroyed by the bullet. The bullet disappear.

Result: pass

### **Test for General Rules about Buff**

#### **1. GR-GRB-2**

Initial State: After the PVE game starts.

Input: Two players enter the PVE game successfully.

Expected result: Different buffs randomly appear in the different locations of the map during a random time period.

Actual result: Different buffs randomly appear in the different locations of the map during a random time period.

Result: pass

#### **2. GR-GRB-6**

Initial State: After the PVE game starts.

Input: The player gets the moving-speed-enhanced buff during the battle.

Expected result: The player's tank gets a higher speed after getting the moving-speed-enhanced buff. The moving-speed-enhanced buff disappear after the player's tank gets it.

Actual result: The player's tank gets a higher speed after getting the moving-speed-enhanced buff. The moving-speed-enhanced buff disappear after the player's tank gets it.

Result: pass

#### **3. GR-GRB-7**

Initial State: After the PVE game starts.

Input: The player gets the wall-enhanced buff during the battle.



Expected result: The player's home base is surrounded by iron walls instead of brick walls. The wall-enhanced buff disappears after the player's tank gets it.

Actual result: The player's home base is surrounded by iron walls instead of brick walls. The wall-enhanced buff disappears after the player's tank gets it.

Result: pass

#### 4. GR-GRB-8

Initial State: After the PVE game starts.

Input: The player gets the bullet-speed-enhanced buff during the battle.

Expected result: Bullets shot by the player's tank gets a higher moving speed. The bullet-speed-enhanced buff disappears after the player's tank gets it.

Actual result: Bullets shot by the player's tank gets a higher moving speed. The bullet-speed-enhanced buff disappears after the player's tank gets it.

Result: pass

### Test for General Rules about Lifes

#### 1. GR-GRL-1

Initial State: After the PVP game starts.

Input: Player 1 controls the tank using "WASD" keys from the keyboard and drives the tank towards the bullet shot by the player 2's tank for three times. The bullet hits player 1's tank three times.

Expected result: Player 1's tank relives for the first and second time after hit by the bullet shot by player 2's tank. Player 1's tank does not relive after the third-time hit by the bullet.

Actual result: Player 1's tank relives for the first and second time after hit by the bullet shot by player 2's tank. Player 1's tank does not relive after the third-time hit by the bullet.

Result: pass

## 2. GR-GRL-2

Initial State: After the PVP game starts.

Input: Player 2 controls the tank using "WASD" keys from the keyboard and drives the tank towards the bullet shot by the player 1's tank for three times. The bullet hits player 2's tank three times.

Expected result: Player 2's tank relives for the first and second time after hit by the bullet shot by player 1's tank. Player 2's tank does not relive after the third-time hit by the bullet.

Actual result: Player 2's tank relives for the first and second time after hit by the bullet shot by player 1's tank. Player 2's tank does not relive after the third-time hit by the bullet.

Result: pass

## 3. GR-GRL-3

Initial State: After the PVE game starts.

Input: Player 1 controls the tank using "WASD" keys from the keyboard and drives the tank towards the bullet shot by the enemy tanks for three times. The bullet hits player 1's tank three times.

Expected result: Player 1's tank relives for the first and second time after hit by the bullet shot by enemy tanks. Player 1's tank does not relive after the third-time hit by the bullet.

Actual result: Player 1's tank relives for the first and second time after hit by the bullet shot by enemy tanks. Player 1's tank does not relive after the third-time hit by the bullet.

Result: pass

## 4. GR-GRL-4

Initial State: After the PVE game starts.

Input: Player 2 controls the tank using "WASD" keys from the keyboard and drives the tank towards the bullet shot by the enemy tanks for three times. The bullet hits player 2's tank three times.

Expected result: Player 2's tank relives for the first and second time after hit by the bullet shot by enemy tanks. Player 2's tank does not relive after the third-time hit by the bullet.

Actual result: Player 2's tank relives for the first and second time after hit by the bullet shot by enemy tanks. Player 2's tank does not relive after the third-time hit by the bullet.

Result: pass

## **Test for General Rules about Timing**

### **1. GR-GRT-1**

Initial State: After the PVP game starts.

Input: Player 1 and player 2 enter the PVP game successfully.

Expected result: Player 1 and player 2 participate a PVP game for at most 3 minutes, if the two home bases are not destroyed and no player's tank is destroyed for 3 times.

Actual result: The game ends after 3 minutes.

Result: pass

### **2. GR-GRT-2**

Initial State: After the PVE game starts.

Input: Player 1 and player 2 enter the PVE game successfully.

Expected result: Player 1 and player 2 participate a PVE game for at most 3 minutes, if the home base is not destroyed and no player's tank is destroyed for 3 times.

Actual result: The game ends after 3 minutes.

Result: pass

## Test for PVP Rules

### 1. GR-PPR-1

Initial State: After the PVP game starts.

Input: Player 1 and player 2 enter the PVP game successfully.

Expected result: There are two home bases shown on the PVP map located on the left and right hand side of the map.

Actual result: There are two home bases shown on the PVP map located on the left and right hand side of the map.

Result: pass

### 2. GR-PPR-2

Initial State: After the PVP game starts.

Input: Player 1 and player 2 play for 3 minutes. No tank is destroyed three times and no tank destroys the other player's home base during this time period.

Expected result: A "Tie" is shown on the screen as the result of the game.

Actual result: A "Tie" is shown on the screen as the result of the game.

Result: pass

### 3. GR-PPR-3

Initial State: After the PVP game starts.

Input: Player 1's tank destroys player 2's tank three times during the 3-minute period.

Expected result: A "Player 1 Win" is shown on the screen as the result of the game.

Actual result: A "Player 1 Win" is shown on the screen as the result of the game.

Result: pass

#### 4. GR-PPR-4

Initial State: After the PVP game starts.

Input: Player 2's tank destroys player 1's tank three times during the 3-minute period.

Expected result: A "Player 2 Win" is shown on the screen as the result of the game.

Actual result: A "Player 2 Win" is shown on the screen as the result of the game.

Result: pass

#### 5. GR-PPR-5

Initial State: After the PVP game starts.

Input: Player 2's tank destroys player 1's home base during the 3-minute period.

Expected result: A "Player 2 Win" is shown on the screen as the result of the game.

Actual result: A "Player 2 Win" is shown on the screen as the result of the game.

Result: pass

#### 6. GR-PPR-6

Initial State: After the PVP game starts.

Input: Player 1's tank destroys player 2's home base during the 3-minute period.

Expected result: A "Player 1 Win" is shown on the screen as the result of the game.

Actual result: A "Player 1 Win" is shown on the screen as the result of the game.

Result: pass

## Test for PVE Rules

### 1. GR-PER-1

Initial State: After the PVE game starts.

Input: Player 1 and player 2 enter the PVE game successfully.

Expected result: There is one home base shown on the PVE map located on the bottom of the PVE map.

Actual result: There is one home base shown on the PVE map located on the bottom of the PVE map.

Result: pass

### 2. GR-PER-2

Initial State: After the PVE game starts.

Input: Player 1 and player 2 enter the PVE game successfully.

Expected result: There is 5 enemy tanks created on the PVE map after the PVE mode starts.

Actual result: There is 5 enemy tanks created on the PVE map after the PVE mode starts.

Result: pass

### 3. GR-PER-3

Initial State: After the PVE game starts.

Input: Player 1 and player 2 enter the PVE game successfully.

Expected result: Enemy tanks move by themselves.

Actual result: Enemy tanks move by themselves.

Result: pass

### 4. GR-PER-4

Initial State: After the PVE game starts.

Input: Player 1 shoots a bullet to the enemy tank using "J" key from the keyboard. The bullet hits the enemy tank.

Expected result: Enemy tanks is destroyed by the bullet.

Actual result: Enemy tanks is destroyed by the bullet.

Result: pass

#### 5. GR-PER-5

Initial State: After the PVE game starts.

Input: Player 2 shoots a bullet to the enemy tank using "," key from the keyboard. The bullet hits the enemy tank.

Expected result: Enemy tanks is destroyed by the bullet.

Actual result: Enemy tanks is destroyed by the bullet.

Result: pass

#### 6. GR-PER-6

Initial State: After the PVE game starts.

Input: Player 1 and player 2 enter the PVE game successfully.

Expected result: Enemy tanks shoot bullets automatically.

Actual result: Enemy tanks shoot bullets automatically.

Result: pass

#### 7. GR-PER-7

Initial State: After the PVE game starts.

Input: A enemy tank is destroyed.

Expected result: Another enemy tank relives.

Actual result: Another enemy tank relives.

Result: pass

#### 8. GR-PER-8

Initial State: After the PVE game starts.

Input: Players' home base is destroyed by the bullet shot by an enemy tank during the 3-minute period.

Expected result: A "Lose" is shown on the screen as the result of the game.

Actual result: A "Lose" is shown on the screen as the result of the game.

Result: pass

#### 9. GR-PER-9

Initial State: After the PVE game starts.

Input: The enemy tanks destroys player 1's and player 2's tanks three times each by using bullets during the 3-minute period.

Expected result: A "Lose" is shown on the screen as the result of the game.

Actual result: A "Lose" is shown on the screen as the result of the game.

Result: pass

#### 10. GR-PER-10

Initial State: The PVE game starts.

Input: Player 1 and player 2 participate the PVE game for 3 minutes. During the game, the home base is not destroyed and no player's tank is destroyed for 3 times.

Expected result: A "Win" is shown on the screen as the result of the game.

Actual result: A "Win" is shown on the screen as the result of the game.

Result: pass



## 2 Nonfunctional Requirements Evaluation

### 2.1 Look and Feel

#### 2.1.1 Visual Appearance Testing

Description of the visual appearance testing: The test cases were conducted by the three developers from the project as the software testers and a test group of people as the test users from the society with different backgrounds among the age of 12 to 50. The participants in the tests were monitored and instructed to strictly follow the specification of the test cases from the test plan to perform the requested tasks. After finished the requested tasks, each participant would fill a feedback form or write a feedback reflection about the visual appearance testing.

1. Test Name: LF-VA-1 & LF-VA-2

Description: The two test cases are similar. They are just in different initial state with the two mode PVP and PVE. So the tests would conducted by open the PVE mode and PVP mode respectively to play the game.

Initial State: The game is at PVE mode. PVE map is loaded and the players have chose tanks. /The game is at PVP mode. PVP map is loaded and the players have chose tanks.

Input/Condition: Players start the PVE game. /Players start the PVP game.

Expected Output: The screen shows the game in a 2-D drawing style.

Result: Pass.

The drawing style of Tank War is in 2-D. No appearance is violated the required drawing style in both modes.

2. Test Name: LF-VA-3

Initial State: The game is installed on the computer.

Input/Condition: The game is just launched.

Expected Output: The screen shows the background of the game system in black.

Result: Pass.

The background of the game is in black.

3. Test Name: LF-VA-4 & LF-VA-5

Description: The two test cases are similar. They are just in different initial state with the two mode PVP and PVE. So the tests would be conducted by opening the PVE mode and PVP mode respectively to play the game.

Initial State: The game is at PVE mode. PVE map is loaded and the players have chosen tanks. /The game is at PVP mode. PVP map is loaded and the players have chosen tanks.

Input/Condition: Players start the PVE game. /Players start the PVP game.

Expected Output: The screen shows the brick walls and iron walls in a grid map.

Result: Pass.

The brick walls and iron walls are in a grid map in both modes.

4. Test Name: LF-VA-6 & LF-VA-7

Description: The two test cases are similar. They are just in different initial state with the two mode PVP and PVE. So the tests would be conducted by opening the PVE mode and PVP mode respectively to play the game.

Initial State: The game is at PVE mode. PVE map is loaded and the players have chosen tanks. /The game is at PVP mode. PVP map is loaded and the players have chosen tanks.

Input/Condition: Players start the PVE game. /Players start the PVP game.

Expected Output: The screen shows brick walls in orange and iron walls consisted of 4 iron cubes in sliver, while the screen displays a home which looks like an eagle.

Result: Pass.

The brick walls are in orange and iron walls consisted of 4 iron cubes in sliver. The home is displayed and it looks like an eagle in both modes.

5. Test Name: LF-VA-8 & LF-VA-9

Description: The two test cases are similar. They are just in different initial state with the two mode PVP and PVE. So the tests would be conducted by opening the PVE mode and PVP mode respectively to play the game.

Initial State: The game is at PVE mode and the PVE map is loaded. /The game is at PVP mode. PVP map is loaded.

Input/Condition: The two players choose two different types of tanks among double-life tank, high-speed tank, double-bullet tank. Then the players start the PVE game. /The two players choose two different types of tanks among double-life tank, high-speed tank, double-bullet tank. Then the players start the PVP game.

Expected Output: The two different types of tanks show on the screen with different colours.

Result: Pass.

The different types of tanks show in different colours in both modes.

6. Test Name: LF-VA-10 & LF-VA-11

Description: The two test cases are similar. They are just in different initial state with the two mode PVP and PVE. So the tests would be conducted by opening the PVE mode and PVP mode respectively to play the game.

Initial State: The game is at PVE mode and the PVE map is loaded and the players have chosen tanks. /The game is at PVP mode. PVP map is loaded and the players have chosen tanks.

Input/Condition: The players start the PVE game. /Players start the PVP game.

Expected Output: The drawing style of TankWar game is very similar to the drawing style of the original game "Battle City".

Result: Pass.

The survey result from the 50 test users shows that there are 92 percent of the test users would consider that the drawing style of Tankwar is more than 90 percent similar with Battle City in both modes.

## 2.2 Usability and Humanity

### 2.2.1 Ease of Use Testing

Description of the ease of use testing: The test cases were conducted by the three developers from the project as the software testers. The participants in the tests were monitored and instructed to strictly follow the specification of the test cases from the test plan to perform the requested tasks. After finished the requested tasks, each participant would fill a feedback form or write a feedback reflection about the ease of use testing.

#### 1. Test Name: UH-EU-1 & UH-EU-2

Description: The two test cases are similar. They are just in different initial state with the two mode PVP and PVE. So the tests would be conducted by opening the PVE mode and PVP mode respectively to play the game.

Initial State: The game is at PVE mode. PVE map is loaded. /The game is at PVP mode. PVP map is loaded.

Input/Condition: The players start to choose tanks.

Expected Output: The screen displays the photos and descriptions of different types of tanks to give a brief introduction when users are choosing tanks.

Result: Pass.

There are instructions about each type of tanks in both modes.

#### 2. Test Name: UH-EU-3 & UH-EU-4

Description: The two test cases are similar. They are just in different initial state with the two mode PVP and PVE. So the tests would be conducted by opening the PVE mode and PVP mode respectively to play the game.

Initial State: The game is at PVE mode. PVE map is loaded and the players have chosen tanks. /The game is at PVP mode. PVP map is loaded and the players have chosen tanks.

Input/Condition: The players press "Enter" key to start the PVE game. /The players press "Enter" key to start the PVP game.

Expected Output: The rules of the game are displayed on the screen. If the players press the key "Enter", the game will start.

Result: Pass.

The rules of the game are displayed on the screen in both modes. Once the players press the key "Enter", the game starts.

### **2.2.2 Learning and Understandability Testing**

Description of the learning and understandability testing: The test cases were conducted by the three developers and a test group of people as the test users from the society with different backgrounds among the age of 12 to 50. The participants in the tests were monitored and instructed to strictly follow the specification of the test cases from the test plan to perform the requested tasks. After finished the requested tasks, each participant would fill a feedback form or write a feedback reflection about the learning and understandability testing.

#### **1. Test Name: UH-LU-1 & UH-LU-2 & UH-LU-3**

Description: The three test cases are similar. They are evaluating the instructions, game rule explanation, words, and icons in the game. The survey is conducted in the one survey form. So the result is presented together.

Initial State: The game is at PVE mode. PVE map is loaded and the players have chose tanks. /The game is at PVP mode. PVP map is loaded and the players have chose tanks. /The game is just launched on the computer.

Input/Condition: The players press "Enter" key to start the PVE game. The players read the rules of the game and operation instruction displayed on the screen, then the game starts. /The players press "Enter" key to start the PVP game. The players read the rules of the game and operation instruction displayed on the screen, then the game starts. /The players choose one of the 3 modes including PVP, PVE, map editing mode.

Expected Output: The players who are above 12 years old and have basic computer operation knowledge know how to play the game after

read the rules and instruction, and the players would consider the words and icons showed on the screen are easily understandable.

Result: Pass.

After surveying 50 test players who are above 12 years old and have basic computer operation knowledge, 85 percent of the test players knows how to play the game after read the rules and instruction, and the players consider the words and icons showed on the screen are easily understandable.

## **2.3 Performance**

### **2.3.1 Speed and Latency Testing**

Description of the speed and latency testing: The test cases were conducted by the three developers from the project as the software testers. The participants in the tests were monitored and instructed to strictly follow the specification of the test cases from the test plan to perform the requested tasks. After finished the requested tasks, each participant would fill a feedback form or write a feedback reflection about the speed and latency testing.

#### **1. Test Name: PT-SL-1**

Initial State: The game is just launched and the game system is presented on the screen.

Input/Condition: The players could perform any operations in the game system.

Expected Output: The system shall respond to a user operation input less than 100 milliseconds.

Result: Pass.

The respond time for the software to the both players' operations is 32 milliseconds which is far less than 100 milliseconds.

Note: Performance evidence

#### **2. Test Name: PT-SL-2**

Initial State: The game is in map editing mode and the player has created their own map.

Input/Condition: The player presses "Enter" key to save the map and names the map. Then the player presses "Enter" key again to start saving process.

Expected Output: The system shall take less than 5 second to save a map on the local disk.

Result: Pass.

The system only takes 8 milliseconds to save a map on the local disk, which is far beyond the expectation.

Note: Performance evidence

### 3. Test Name: PT-SL-3 & PT-SL-4

Description: The two test cases are similar. They are just in different initial state with the two mode PVP and PVE. So the tests would be conducted by opening the PVE mode and PVP mode respectively to load the map.

Initial State: The game is in the PVP mode. /The game is in the PVE mode.

Input/Condition: The player chooses the map to be loaded.

Expected Output: The system takes less than 5 second to load the PVP map from local disk. /The system takes less than 5 second to load the PVE map from local disk.

Result: Pass.

The system only takes 120 milliseconds to load the PVE map from local disk and 100 milliseconds to load the PVP map from local disk. The actual time for loading is far less than the required time.

Note: Performance evidence

### 2.3.2 Accuracy Testing

Description of the accuracy testing: The test cases were conducted by the three developers from the project as the software testers. The participants in the tests were monitored and instructed to strictly follow the specification of the test cases from the test plan to perform the requested tasks. After finishing the requested tasks, each participant would fill a feedback form or write a feedback reflection about the accuracy testing.

1. Test Name: PT-AT-1

Initial State: The game code is functional and is implemented in python.

Input/Condition: There are floating point numbers used in the code.

Expected Output: Floating point numbers used in the code is in double precision.

Result: Pass.

Floating point numbers used in the code is in double precision.

### **2.3.3 Reliability and Availability Testing**

Description of the reliability and availability testing: The test cases were conducted by the three developers from the project as the software testers and a test group of people as the test users from the society with different backgrounds among the age of 12 to 50. The participants in the tests were monitored and instructed to strictly follow the specification of the test cases from the test plan to perform the requested tasks. After finished the requested tasks, each participant would fill a feedback form or write a feedback reflection about the reliability and availability testing.

1. Test Name: PT-RA-1

Initial State: The game is launched.

Input/Condition: The two players are both engaged in the game.

Expected Output: The system crashes do not exceed 3 times within 6-hour constantly running.

Result: Pass.

After 10 TankWar games are constantly played by 20 test players in group of 2 for 6 hour, the crashing situation does not happen at all.

2. Test Name: PT-RA-2

Initial State: The game is launched.

Input/Condition: The game is in the local computer. The system is installed python and pygame.



Expected Output: The game is launched on the flowing operating system including Windows 10, Mac OS 10.8, and Linux.

Result: Pass.

The TankWar game is fully functional on the the above system.

#### **2.3.4 Robustness Testing**

Description of the robustness testing: The test cases were conducted by the three developers from the project as the software testers. The participants in the tests were monitored and instructed to strictly follow the specification of the test cases from the test plan to perform the requested tasks. After finished the requested tasks, each participant would fill a feedback form or write a feedback reflection about the robustness testing.

1. Test Name: PT-RT-1

Initial State: The game is launched.

Input/Condition: The keys besides the desired input keys mentioned in the functional requirements are triggered.

Expected Output: The program would ignore the undesired game input and only identify the desired input keys.

Result: Pass.

In each screen, the three testers randomly press 100 input keys other than the desired input keys. There is no action performed by the system by triggering the undesired key inputs.

Note: Robustness evidence

#### **2.3.5 Capacity Testing**

Description of the capacity testing: The test cases were conducted by the three developers from the project as the software testers and a test group of people as the test users from the society with different backgrounds among the age of 12 to 50. The participants in the tests were monitored and instructed to strictly follow the specification of the test cases from the test plan to perform the requested tasks. After finished the requested tasks, each participant would fill a feedback form or write a feedback reflection about the capacity testing.

1. Test Name: PT-CT-1 & PT-CT-2

Description: The two test cases are similar. They are just in different initial state with the two mode PVP and PVE. So the tests would be conducted by opening the PVE mode and PVP mode respectively to play the game.

Initial State: The game is in the PVE mode. The PVE map is loaded and the players have chosen their tanks. /The game is in the PVP mode. The PVP map is loaded and the players have chosen their tanks.

Input/Condition: The PVE game is started and two players both perform their operation to the tanks at the same time. /The PVP game is started and two players both perform their operation to the tanks at the same time.

Expected Output: The game is able to support the two players' operations at the same time.

Result: Pass.

The two players both perform their operation to the tanks at the same time. The game is able to support the two players' operations at the same time in both modes.

## 2.4 Operational and Environmental

### 2.4.1 Physical Environment Testing

Description of the physical environment testing: The test cases were conducted by the three developers from the project as the software testers and a test group of people as the test users from the society with different backgrounds among the age of 12 to 50. The participants in the tests were monitored and instructed to strictly follow the specification of the test cases from the test plan to perform the requested tasks. After finishing the requested tasks, each participant would fill a feedback form or write a feedback reflection about the physical environment testing.

1. Test Name: OE-PE-1

Description: The two test cases are similar. They are just in different initial state with the two mode PVP and PVE. So the tests would

conducted by open the PVE mode and PVP mode respectively to play the game.

Initial State: The game is installed on the computer with 2-cores processor and 1GB RAM. Python and pygame is installed too.

Input/Condition: The players launch and play the game.

Expected Output: The game is fully functional on the computer.

Result: Pass.

The PVE and PVP games are fully functional on a computer with 2-cores processor and 1GB RAM.

#### **2.4.2 Productization and Release Testing**

Description of the productization and release testing: The test cases were conducted by the three developers from the project as the software testers and a test group of people as the test users from the society with different backgrounds among the age of 12 to 50. The participants in the tests were monitored and instructed to strictly follow the specification of the test cases from the test plan to perform the requested tasks. After finished the requested tasks, each participant would fill a feedback form or write a feedback reflection about the productization and release testing.

##### **1. Test Name: OE-PR-1**

Initial State: The game is fully functional and is developed well.

Input/Condition: The product is in the delivery state.

Expected Output: The product is wrap up as a file package and can be easily downloaded from Git Lab as an open source project.

Result: Pass.

The software is uploaded to GitLab and the installation instruction is provided.

##### **2. Test Name: OE-PR-2**

Initial State: The game is developed and before releasing.

Input/Condition: The game is in the testing process.

Expected Output: The system is released after full verifying and validation.

Result: Pass.

The system is validated and verified fully.

## **2.5 Maintainability and Support**

### **2.5.1 Supportability Testing**

Description of the supportability testing: The test cases were conducted by the three developers from the project as the software testers. The participants in the tests were monitored and instructed to strictly follow the specification of the test cases from the test plan to perform the requested tasks. After finished the requested tasks, each participant would fill a feedback form or write a feedback reflection about the supportability testing.

#### **1. Test Name: MS-ST-1**

Initial State: The project state is in the software development process and code implementation.

Input/Condition: The naming convention is needed in the software development. The variables in the code need to be created in the code implementation.

Expected Output: The naming convention of the variables and development process are consistent.

Result: Pass.

The software tester conducts a inspection on the code and development documentation to check the naming consistence.

#### **2. Test Name: MS-ST-2**

Initial State: The code of the game is implementing.

Input/Condition: The software developer is implementing the code.

Expected Output: The code implementation include the comments that generates Doxygen documentation.

Result: Pass.

The Doxygen documentation is generated with well commented information.

## **2.6 Security**

### **2.6.1 Integrity and Privacy Testing**

Description of the integrity and privacy testing: The test cases were conducted by the three developers from the project as the software testers. The participants in the tests were monitored and instructed to strictly follow the specification of the test cases from the test plan to perform the requested tasks. After finished the requested tasks, each participant would fill a feedback form or write a feedback reflection about the integrity and privacy testing.

#### **1. Test Name: ST-IP-1**

Initial State: The game is in the map editing mode. The map is created.

Input/Condition: The players try to save the map.

Expected Output: The map is safely saved on the local disk and there is no loss of the information of the map.

Result: Pass.

The software testers design and save 10 maps in either PVE and PVP map creation. There is no information loss in the saved maps comparing with the designed maps.

#### **2. Test Name: ST-IP-2**

Initial State: The game is developed well.

Input/Condition: The game is in testing process.

Expected Output: There is no any mechanism to save personal information from users.

Result: Pass.

In the source code, there is no any mechanism to save personal information from users.

## **2.7 Cultural**

### **2.7.1 Cultural Testing**

Description of the cultural testing: The test cases were conducted by the three developers from the project as the software testers. The participants in the tests were monitored and instructed to strictly follow the specification of the test cases from the test plan to perform the requested tasks. After finished the requested tasks, each participant would fill a feedback form or write a feedback reflection about the cultural testing.

1. Test Name: CT-CT-1

Initial State: The game is developed well and fully functional.

Input/Condition: The game is in the testing process.

Expected Output: The game system is in English with Canadian spelling. The game does not contain any content that could be considered offensive.

Result: Pass.

The game system is in English with Canadian spelling. After software walk-through meeting, any content that could be considered offensive is not found.

## **3 Comparison to Existing Implementation**

None.

Comparison to Existing implementation is not available for our project.

## **4 Unit Testing**

1. Test Name: MS-MV-1

Type : Functional, Static, Automated.

Initial State: A map is designed follow a certain known map file and saved in the map editing mode.

Input: The saved file and the certain known map file.

Expected Output: The contents in two files are equivalent.

Result: Pass.

2. Test Name: UBT-SS-1

Type : Functional, Static, Automated.

Initial State: A bullet object is created.

Input: An integer represents the bullet's speed.

Expected Output: The bullet's speed is changed to the input integer.

Result: Pass.

3. Test Name: UBT-SL-1

Type : Functional, Static, Automated.

Initial State: A bullet object is created.

Input: A boolean value represents the bullet's life.

Expected Output: The bullet's life is changed to the boolean value.

Result: Pass.

4. Test Name: UBT-SST-1

Type : Functional, Static, Automated.

Initial State: A bullet object is created.

Input: A boolean value represents that the bullet is strong or not.

Expected Output: The bullet's value of strong is changed to the boolean value.

Result: Pass.

5. Test Name: UDT-INIT-1

Type : Functional, Static, Automated.

Initial State: A prepared decTime module.

Input: Create a decTime object with an input integer represents the number of seconds.

Expected Output: The number of seconds are changed into the form of hours:minutes:seconds.

Result: Pass.

6. Test Name: UFD-CH-1

Type : Functional, Static, Automated.

Initial State: A food object is created.

Input: Call the function change.

Expected Output: The value of life is changed to True.

Result: Pass.

7. Test Name: UET-ST-1

Type : Functional, Static, Automated.

Initial State: An EnemyTank object is created.

Input: Call the function shoot.

Expected Output: The value of the tank's bullet's life is changed to True.

Result: Pass.

8. Test Name: UMT-ST-1

Type : Functional, Static, Automated.

Initial State: An MyTank object is created.

Input: Call the function shoot.

Expected Output: The value of the tank's bullet's life is changed to True.

Result: Pass.



9. Test Name: UMT-LU-1 & UMT-LU-2

Description: These two tests are similar. The first one is to increase the level by one, and the second one is to test when the level is already maximized.

Type : Functional, Static, Automated.

Initial State: An MyTank object is created.

Input: Call the function levelup.

Expected Output: Nothing is changed if tank's level is 2, otherwise the level is increased by 1.

Result: Pass.

10. Test Name: UMT-LD-1 & UMT-LD-2

Description: These two tests are similar. The first one is to decrease the level by one, and the second one is to test when the level is already minimized.

Type : Functional, Static, Automated.

Initial State: An MyTank object is created.

Input: Call the function leveldown.

Expected Output: Nothing is changed if tank's level is 0, otherwise the level is decreased by 1, and if the level is decreased to 0, the tank's speed is changed to 6 and the bullet's value of strong is changed to False.

Result: Pass.

11. Test Name: UDT-BPS-1

Type : Functional, Static, Automated.

Initial State: An doubleLifeTank object is created.

Input: Call the function bulletproof\_start.

Expected Output: The value of the bullet proof is changed to True.

Result: Pass.

12. Test Name: UDT-BPE-1  
Type : Functional, Static, Automated.  
Initial State: An doubleLifeTank object is created.  
Input: Call the function bulletproof\_end.  
Expected Output: The value of the bullet proof is changed to False.  
Result: Pass.
13. Test Name: UDT-GBP-1  
Type : Functional, Static, Automated.  
Initial State: An doubleLifeTank object is created.  
Input: Call the function get\_bullet\_proof.  
Expected Output: The value of the bullet proof is returned.  
Result: Pass.
14. Test Name: UFT-DB-1  
Type : Functional, Static, Automated.  
Initial State: An fastBulletTank object is created.  
Input: Call the function double\_bullet.  
Expected Output: The two bullets' life are both changed to True.  
Result: Pass.
15. Test Name: UHT-LS-1  
Type : Functional, Static, Automated.  
Initial State: An highSpeedTank object is created.  
Input: Call the function leap\_start.  
Expected Output: The tank's speed is increased by 3.  
Result: Pass.

16. Test Name: UHT-LE-1

Type : Functional, Static, Automated.

Initial State: An highSpeedTank object is created.

Input: Call the function leap\_end.

Expected Output: The tank's speed is changed to 3.

Result: Pass.

## 5 Changes Due to Testing

During the testing, we found that if a tank is shot by a bullet at the initial location, where it will be reborn, that bullet will kill the tank again right after the tank appears until the tank's life is zero. We finally figured out that it's because the bullet's life was not set to false immediately after it collides with the tank and we fixed it.

We also found that when a double life tank is shot by a bullet, the tank's life will be decreased by 2, and the reason is that we put a 'if' statement while it should be 'elif', which made the program run both conditions and decrease the tank's life twice.

There was also another problem that made the player2's tank appear at the player's tank's initial location after it's killed. We found out the reason is that the location to appear after being killed is not distinguished between tank\_T1, and tank\_T2.

## 6 Automated Testing

The unittest is used to do the automated testings. Test cases which mean to cover all the conditions are wrote for those functions can be automated tested. After we designing the inputs for those functions, the outputs are compared with the expected outputs, and if they are equivalent, the test will pass, otherwise it will fail.

See Test Cases: MS-MV-1, UBT-SS-1, UBT-SL-1, UBT-SST-1, UDT-INIT-1, UFD-CH-1, UET-ST-1, UMT-ST1, UMT-LU-1, UMT-LU-2, UMT-LD-1,

UMT-LD-2, UDT-BPS-1, UDT-BPE-1, UDT-GBP-1, UFT-DB-1, UHT-LS-1, UHT-LE-1.

## 7 Trace to Requirements

### 7.1 Traceability Between Test Cases and Requirements

Functional Requirement ID	Priority	Test Case ID
FR1	5	GPT-PPMS-1, GPT-PEMS-1, GPT-MEMS-1
FR2	5	ME-PVPAB-1, ME-PVPAB-2, ME-PVPAB-3, ME-PVPAB-4, ME-PVPAI-1, ME-PVPAI-2, ME-PVPAI-3, ME-PVPAI-4, ME-PVPDW-1, ME-PVPDW-2, ME-PVPDW-3, ME-PVEAB-1, ME-PVEAB-2, ME-PVEAB-3, ME-PVEAB-4, ME-PVEAI-1, ME-PVEAI-2, ME-PVEAI-3, ME-PVEAI-4, ME-PVEDW-1, ME-PVEDW-2, ME-PVEDW-3
FR3	4	ME-PVPSM-1, ME-PVESM-1
FR4	5	MT-GU-1, MT-GU-2, MT-GU-3, MT-GU-4, MT-GU-5, MT-GU-6, MT-GU-7, MT-GU-8, MT-GD-1, MT-GD-2, MT-GD-3, MT-GD-4, MT-GD-5, MT-GD-6, MT-GD-7, MT-GD-8, MT-GL-1, MT-GL-2, MT-GL-3, MT-GL-4, MT-GL-5, MT-GL-6, MT-GL-7, MT-GL-8, MT-GR-1, MT-GR-2, MT-GR-3, MT-GR-4, MT-GR-5, MT-GR-6, MT-GR-7, MT-GR-8.
FR5	5	TAT-UUS-1, TAT-UUS-2, TAT-UUS-3, TAT-UUS-4

Functional Requirement ID	Priority	Test Case ID
FR6	5	TAT-SB-1, TAT-SB-2, TAT-SB-3, TAT-SB-4, TAT-SB-5, TAT-SB-6, TAT-SB-7, TAT-SB-8
FR7	5	GR-GRW-1, GR-GRW-2, GR-GRW-3, GR-GRW-4
FR8	5	GR-GRW-5, GR-GRW-6, GR-GRW-7, GR-GRW-8
FR9	5	MT-WB-1, MT-WB-2, MT-WB-3, MT-WB-4, MT-WB-5, MT-WB-6
FR10	5	GPT-PPTS-1, GPT-PPTS-2, GPT-PETS-1, GPT-PETS-2
FR11	5	GR-PPR-2, GR-PPR-3, GR-PPR-4, GR-PPR-5, GR-PPR-6, GR-PER-8, GR-PER-9, GR-PER-10
FR12	5	GR-GRB-2, GR-GRB-6, GR-GRB-7, GR-GRB-8
FR13	5	GR-PER-3
FR14	5	GR-PER-4, GR-PER-5
FR15	5	GR-PER-7
FR16	5	GR-PER-2, GR-PER-7
FR17	5	GR-PER-1
FR18	5	GR-PPR-1
FR19	5	GR-PER-8, GR-PER-9, GR-PER-10
FR20	5	GR-PPR-2, GR-PPR-3, GR-PPR-4, GR-PPR-5, GR-PPR-6
FR21	5	GR-GRL-1, GR-GRL-2, GR-GRL-3, GR-GRL-4

Table 2: Traceability Matrix for Functional Requirement

Non-Functional Requirement ID	Priority	Test Case ID
LF1	5	LF-VA-1, LF-VA-2
LF2	5	LF-VA-3
LF3	4	LF-VA-4, LF-VA-5
LF4	5	LF-VA-6, LF-VA-7
LF5	5	LF-VA-6, LF-VA-7
LF6	5	LF-VA-8, LF-VA-9
LF7	5	LF-VA-6, LF-VA-7
LF8	5	LF-VA-10, LF-VA-11
UH1	5	UH-EU-1, UH-EU-2
UH2	5	UH-EU-3, UH-EU-4
UH4	5	UH-LU-1, UH-LU-2
UH5	5	UH-LU-3
PR1	5	PT-SL-1
PR2	5	PT-SL-2, PT-SL-3, PT-SL-4
PR4	5	PT-AT-1
PR5	5	PT-RA-1
PR6	5	PT-RA-2
PR7	5	PT-RT-1
PR8	5	PT-CT-1, PT-CT-2
OE1	5	OE-PE-1
OE3	5	OE-PR-1
OE4	5	OE-PR-1
OE5	5	OE-PR-2
MS2	5	MS-ST-1
MS3	5	MS-ST-2
MS4	5	PT-RA-2
SR2	5	ST-IP-1
SR3	5	ST-TP-2
CR1	5	CT-CT-1
CR2	5	CT-CT-1

Table 3: Traceability Matrix for Non-functional Requirement

Unit Test ID	Priority	FR ID
MS-MV-1	5	FR3
UBT-SS-1	5	FR6
UBT-SL-1	5	FR7, FR8
UBT-SST-1	5	FR8
UDT-INIT-1	5	FR19, FR20
UFD-CH-1	5	FR12
UET-ST-1	5	FR14
UMT-ST-1	5	FR6
UMT-LU-1	5	FR12
UMT-LU-2	5	FR12
UMT-LD-1	5	FR12
UMT-LD-2	5	FR12
UDT-BPS-1	5	FR5
UDT-BPE-1	5	FR5
UDT-GBP-1	5	FR5
UFT-DB-1	5	FR5
UHT-LS-1	5	FR5
UHT-LE-1	5	FR5

Table 4: Traceability Matrix for Unit Test Case

## 8 Trace to Modules



Test Case ID	Module ID	Module Name
GPT-PPMS-1, GPT-PEMS-1, GPT-MEMS-1	M5, M6, M7, M8	PvsE, PvsP, mapEditing, main
ME-PVPAB-1, ME-PVPAB-2, ME-PVPAB-3, ME-PVPAB-4, ME-PVPAI-1, ME-PVPAI-2, ME-PVPAI-3, ME-PVPAI-4, ME-PVPDW-1, ME-PVPDW-2, ME-PVPDW-3, ME-PVEAB-1, ME-PVEAB-2, ME-PVEAB-3, ME-PVEAB-4, ME-PVEAI-1, ME-PVEAI-2, ME-PVEAI-3, ME-PVEAI-4, ME-PVEDW-1, ME-PVEDW-2, ME-PVEDW-3	M2, M7, M17, M18, M19	myTankControl, mapEditing, Map, MapEdit- Tank, wall
ME-PVPSM-1, ME-PVESM-1	M7, M17	mapEditing, Map
MT-GU-1, MT-GU-2, MT-GU-3, MT-GU-4, MT-GU-5, MT-GU-6, MT-GU-7, MT-GU-8, MT-GD-1, MT-GD-2, MT-GD-3, MT-GD-4, MT-GD-5, MT-GD-6, MT-GD-7, MT-GD-8, MT-GL-1, MT-GL-2, MT-GL-3, MT-GL-4, MT-GL-5, MT-GL-6, MT-GL-7, MT-GL-8, MT-GR-1, MT-GR-2, MT-GR-3, MT-GR-4, MT-GR-5, MT-GR-6, MT-GR-7, MT-GR-8.	M12, M2	myTank, my- TankControl
TAT-UUS-1, TAT-UUS-2, TAT-UUS-3, TAT-UUS-4	M13, M14, M15, M2	highSpeedTank, doubleLifeTank, fastBulletTank, myTankControl
TAT-SB-1, TAT-SB-2, TAT-SB-3, TAT-SB-4, TAT-SB-5, TAT-SB-6, TAT-SB-7, TAT-SB-8	M12, M2	myTank, my- TankControl
GR-GRW-1, GR-GRW-2, GR-GRW-3, GR-GRW-4	M9, M4	bullet, display
GR-GRW-5, GR-GRW-6, GR-GRW-7, GR-GRW-8	M9, M4	bullet, display
MT-WB-1, MT-WB-2, MT-WB-3, <sup>54</sup> MT-WB-4, MT-WB-5, MT-WB-6	M12, M2	myTank, my- TankControl

Test Case ID	Module ID	Module Name
GPT-PPTS-1, GPT-PPTS-2, GPT-PETS-1, GPT-PETS-2	M3	screen
GR-PPR-2, GR-PPR-3, GR-PPR-4, GR-PPR-5, GR-PPR-6, GR-PER-8, GR-PER-9, GR-PER-10	M3	Screen
GR-GRB-2, GR-GRB-6, GR-GRB-7, GR-GRB-8	M10	food
GR-PER-3	M4	display
GR-PER-4, GR-PER-5	M16, M4	enemyTank, display
GR-PER-7	M4	display
GR-PER-2, GR-PER-7	M4	display
GR-PER-1	M17, M5	Map, PvsE
GR-PPR-1	M17, M6	Map, PvsP
GR-PER-8, GR-PER-9, GR-PER-10	M5	PvsE
GR-PPR-2, GR-PPR-3, GR-PPR-4, GR-PPR-5, GR-PPR-6	M6	PvsP
GR-GRL-1, GR-GRL-2, GR-GRL-3, GR-GRL-4	M4, M5, M6	display, PvsE, PvsP

Table 5: Traceability Matrix for Functional Requirement

Test Case ID	Module ID	Module Name
LF-VA-1, LF-VA-2	M8	main
LF-VA-3	M3	Screen
LF-VA-4, LF-VA-5	M17, M19	Map, wall
LF-VA-6, LF-VA-7	M19	wall
LF-VA-6, LF-VA-7	M19	wall
LF-VA-8, LF-VA-9	M13, M14, M15, M16, M18	highSpeedTank, doubleLifeTank, fastBulletTank, MapEditTank, enemyTank
LF-VA-6, LF-VA-7	M17	Map
LF-VA-10, LF-VA-11	M5, M6	PvsE, PvsP
UH-EU-1, UH-EU-2	M3	Screen
UH-EU-3, UH-EU-4	M3	Screen
UH-LU-1, UH-LU-2	M3	Screen
UH-LU-3	M3	Screen
PT-SL-1	M4, M2	display, my-TankControl
PT-SL-2, PT-SL-3, PT-SL-4	M7, M17	Map, mapEditing
PT-AT-1	M7	Map
PT-RA-1	All the modules	All the modules
PT-RA-2	All the modules	All the modules
PT-RT-1	M7	mapEditing
PT-CT-1, PT-CT-2	M2	myTankControl
OE-PE-1	All the modules	All the modules
OE-PR-1	All the modules	All the modules
OE-PR-2	All the modules	All the modules
MS-ST-1	All the modules	All the modules
MS-ST-2	All the modules	All the modules
PT-RA-2	All the modules	All the modules
ST-IP-1	M17	Map
ST-TP-2	M17	Map
CT-CT-1	All the modules	All the modules
CT-CT-1	All the modules	All the modules

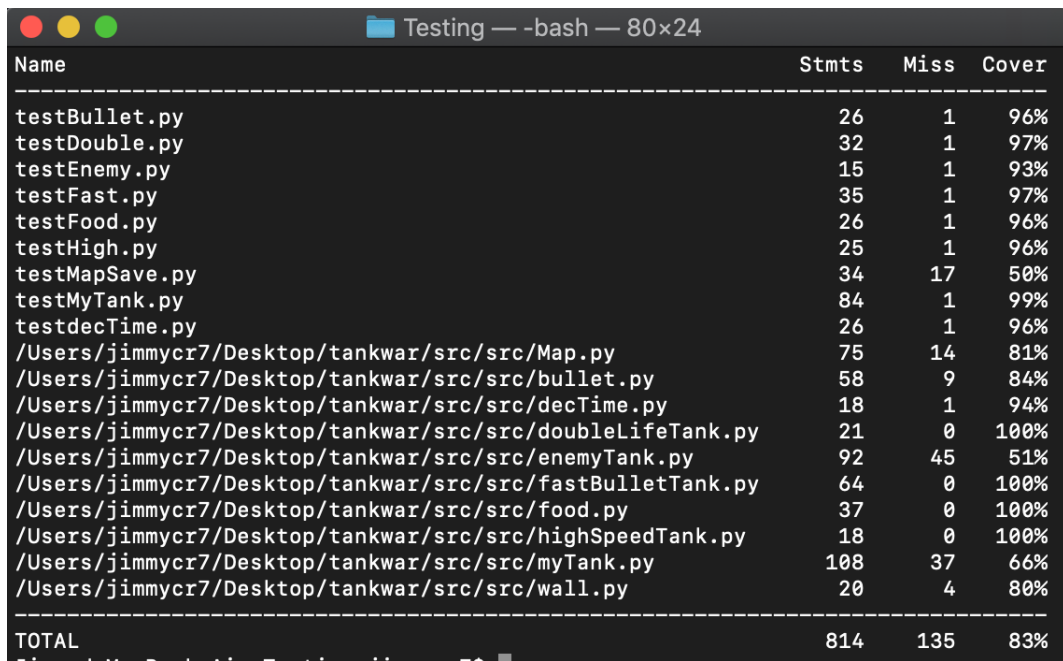
Table 6: Traceability Matrix for Non-functional Requirement

Unit Test ID	Module ID	Module Name
MS-MV-1	M7, M17	mapEditing, Map
UBT-SS-1	M9	bullet
UBT-SL-1	M9	bullet
UBT-SST-1	M9	bullet
UDT-INIT-1	M11	decTime
UFD-CH-1	M10	food
UET-ST-1	M16	enemyTank
UMT-ST-1	M12	myTank
UMT-LU-1	M12	myTank
UMT-LU-2	M12	myTank
UMT-LD-1	M12	myTank
UMT-LD-2	M12	myTank
UDT-BPS-1	M14	doubleLifeTank
UDT-BPE-1	M14	doubleLifeTank
UDT-GBP-1	M14	doubleLifeTank
UFT-DB-1	M15	fastBulletTank
UHT-LS-1	M13	highSpeedTank
UHT-LE-1	M13	highSpeedTank

Table 7: Traceability Matrix for Unit Test Case

## 9 Code Coverage Metrics

We get a overall coverage of 83 percent. Most of the module had been tested and most of the code had been covered in our python unittest cases. The reason why our test cases cannot achieve 100 percent is that our project is a game, therefore some of the functions can only be tested manually. In other words, some functions can be tested only if we start the game and play with it. The detailed model traceability about unit test cases can be found above in the "Trace to Modules" section. The result is shown below in Figure 1: Coverage Test Screenshot.



Name	Stmts	Miss	Cover
testBullet.py	26	1	96%
testDouble.py	32	1	97%
testEnemy.py	15	1	93%
testFast.py	35	1	97%
testFood.py	26	1	96%
testHigh.py	25	1	96%
testMapSave.py	34	17	50%
testMyTank.py	84	1	99%
testdecTime.py	26	1	96%
/Users/jimmycr7/Desktop/tankwar/src/src/Map.py	75	14	81%
/Users/jimmycr7/Desktop/tankwar/src/src/bullet.py	58	9	84%
/Users/jimmycr7/Desktop/tankwar/src/src/decTime.py	18	1	94%
/Users/jimmycr7/Desktop/tankwar/src/src/doubleLifeTank.py	21	0	100%
/Users/jimmycr7/Desktop/tankwar/src/src/enemyTank.py	92	45	51%
/Users/jimmycr7/Desktop/tankwar/src/src/fastBulletTank.py	64	0	100%
/Users/jimmycr7/Desktop/tankwar/src/src/food.py	37	0	100%
/Users/jimmycr7/Desktop/tankwar/src/src/highSpeedTank.py	18	0	100%
/Users/jimmycr7/Desktop/tankwar/src/src/myTank.py	108	37	66%
/Users/jimmycr7/Desktop/tankwar/src/src/wall.py	20	4	80%
TOTAL	814	135	83%

Figure 1: Coverage Test Screenshot