2D Tanks Game

Requirements and Test Document

Tyler Carr

March 6, 2020

Modified Apr 15, 2020

CS 225, Spring 2020

Embry-Riddle Aeronautical University

Daytona Beach campus

1 Aerospace Boulevard

Daytona Beach, FL 32114

**Introduction:**

The purpose of a requirements document is to show the functionality that a specific program will have. Requirements outline the scope of the project, showing the programmers exactly what is needed to be done and preventing a client from asking for more than what is officially agreed upon [1].

The goal of this project in particular is to create an engaging game that users will enjoy and want to come back and play again. The game is a side-view of a hilly terrain with two tanks, one being controlled by the player and the other by the computer. The user will take turns with the opponent, and when it is their turn, they can move their tank left and right, aim their cannon, and fire at the opponent tank. Once one tank runs out of health, the other wins. The game will feature user accounts with in-game money and tank upgrades in an attempt to engage the user and achieve the overall goals. Please refer to the Software Design Document for a more detailed explanation of the 2D Tanks Game.

**Background information:**

**Terms:**

* Tank: The player’s tank that the player can control. It is an object that contains user upgrades which are applied on game start. The player can move the tank left and right the number of spaces that is allowed, adjust the cannon angle between -90 and 270 degrees, and fire their cannon when it is their turn.
* Opponent: The “computer-controlled” tank that the player is trying to defeat. There are two difficulties, easy and hard. The hard opponent will aim as accurate as possible with its algorithm, and the easy opponent will utilize that same algorithm but apply a random value between -20 and 20 degrees to the angle to make it easier to defeat. The hard opponent will have random multipliers between 5 and 10 applied to it at the beginning of each round.
* Round: A round is one game that is played from when the game is started until the current game ends. The player can play as many rounds as they desire. The upgrade menu is displayed between rounds.
* Terrain: The curvy line that represents hills that the tanks move around on. The terrain is different for every round. Refer to the Formulas section for how the line is generated. Tanks and projectiles stay on top of the terrain line.
* Cannon: The cannon resembles a cannon from an actual tank. In the game, it can be moved clockwise and counterclockwise with the down and up arrow keys respectively between -90 and 270 degrees.
* Projectile: The projectile is shot when a tank fires their cannon, creating a new projectile object. For the user, that would be when the space bar is pressed during their turn. The projectile follows the Kinematic Equations listed in the Formulas section below until the terrain is reached, where the angle is defined by the angle the user moves their cannon to.
* Tank upgrades: Tank upgrades can be purchased when the “Upgrades” screen is displayed in between rounds. Only the user can purchase upgrades, but the opponent has new randomly selected upgrades each new round. Upgrades can only be purchased if the user has enough in-game money. When an upgrade is purchased, in-game money is deducted from the user’s balance, and the number of times they have upgraded is applied to their tank upon the start of the following rounds. For example, if the user has upgraded their traveling distance 3 times, they will be able to move 3 extra units every round in addition to the default amount. Tank upgrades are saved to a file and can be retrieved upon game start. The available “multipliers” in the program are: health, projectile velocity, projectile damage, and distance. You can only upgrade each item a maximum of 10 times.
* xMax: The width of the screen which defines the limit on the right side of the terrain that projectiles and tanks cannot pass.
* Default Health: 10
* Default Projectile Velocity: 50
* Default Distance to Move per Turn: 50
* Default Projectile Damage: 2 (regular), 5 (critical)

**Formulas:**

Kinematic Equations (for projectiles)

* Equation for updating y coordinate:
* Equation for updating x coordinate:
* : Initial speed of the projectile (a set value).
* t: Current time since the projectile was launched.
* g: Acceleration of gravity constant (9.81).
* and Initial x and y positions of the tank when projectile is shot.
* : Angle at which the cannon was moved to when projectile is shot.

Random Terrain Equation

* Equation:
* a: Randomly generated double between 100 and 250.
* b: Randomly generated double between 0.0025 and 0.007.
* c: Randomly generated double between 0 and 5.

Award Money Equation

* health: Health that the user ends the round with
* startingHealth: Default health that the user starts with (10)
* healthMultiplier: Number of upgrades that the user has made to their tank health
* totalTurns: Total amount of turns that is allowed per round
* turns: Turns that the user ends the round with

**Requirements:**

Below are the requirements for the 2D Tanks Game. They outline the functionality of the program and set in stone the exact things that need to be done in order for the project to be complete. Each requirement features one or more user stories, which give examples of what the requirement should look like when somebody is actually using the program.

1. The program shall apply a user’s saved upgrades from a file to the user tank if a user enters a username before the game begins.
   1. User Stories: 1
2. The program shall generate a new terrain at the beginning of each round that follows the Random Terrain Equation in the Formulas section above.
   1. User Stories: 9
3. The user’s tank’s x coordinate shall be 10 units from the left side of the screen (x=10) upon game start.
   1. User Stories: 10
4. The opponent’s tank’s x coordinate shall be 10 units from the right side of the screen (x=xMax-10) upon game start.
   1. User Stories: 11
5. Both tanks shall input the initial x coordinate into the Random Terrain Equation in the Formulas section above to retrieve the y coordinate that shall be used upon game start.
   1. User Stories: 10, 11
6. The user shall be able to move their tank 1 unit plus the number of movement multipliers the user has to the left when the left arrow key is pressed with the y coordinate updating according to the terrain equation.
   1. User Stories: 2, 12, 14
7. The user shall be able to move their tank 1 unit plus the number of movement multipliers the user has to the right when the right arrow key is pressed with the y coordinate updating according to the terrain equation.
   1. User Stories: 2, 12, 14
8. The user shall be able to increase their tank’s cannon angle by 1 degree when the up arrow key is pressed.
   1. User Stories: 3, 14
9. The user shall be able to decrease their tank’s cannon angle by 1 degree when the down arrow key is pressed.
   1. User Stories: 3, 14
10. The user shall be able to fire their tank’s cannon when the space bar is pressed, which creates a new projectile object.
    1. User Stories: 4, 14
11. The projectile shall follow kinematic equation of a body in air, which is defined in the Formulas section above, until the y coordinate is less than or equal to the y coordinate of the terrain, which is obtained from inputting the x coordinate into the terrain equation.
    1. User Stories: 4
12. The projectiles shall decrease the health of a tank by 2 points plus the number of projectile damage upgrades that the user has if the x coordinate is within 3 to 5 units of the tank’s x coordinate.
    1. User Stories: 13
13. The projectiles shall decrease the health of a tank by 5 points plus the number of projectile damage upgrades that the user has if the x coordinate is within 0 to 2 units of the tank’s x coordinate.
    1. User Stories: 13, 15
14. The program shall end the current game when the health points for either the player or the opponent tank is equal to or less than zero.
    1. User Stories: 16
15. The program shall end the current game in a draw after 10 rounds regardless of the health of either tank.
    1. User Stories: 17
16. The program shall give money to the user if their health is above zero after a game ends with the Award Money Equation in the Formulas section listed above.
    1. User Stories: 18
17. The user shall be able to purchase tank upgrades if they have sufficient funds.
    1. User Stories: 5, 20
18. The program shall save tank upgrades to a file when a tank upgrade is made.
    1. User Stories: 21, 22

**Test Cases:**

This table includes three test cases for five of the requirements listed above, making a total of 15 test cases. Each test case defines a set of inputs that are to be entered into the program and specifies the result that is expected upon fulfilling the conditions. When a test case is completed, the observed result is logged and it is noted whether or not the observed result matched the expected result. This is extremely helpful for the programmer as they are working on their project because it helps them to think about and fix conditions that may potentially cause their program to fail or cause the user to perform an action that shouldn’t be allowed.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Req’t**  **ID** | **Test**  **Case**  **ID** | **Initial**  **Conditions**  **And Input** | **Expected Behavior**  **Or Output** | **Actual**  **Behavior**  **Or Output** | **Pass**  **Fail** |
| 1 | A | Username left blank and start game button pressed | An error shall occur indicating that the user needs to either input a username or create an account. | The error occurs. | Pass |
| 1 | B | Valid username entered (username: “test”, whose information exists in the file with all multipliers at 1) and start game button pressed | A player tank object should be created with all multipliers set to 1. | A player tank object is created with all multipliers set to 1 | Pass |
| 1 | C | Invalid username entered (username: “test2”, whose information does not exist in the file) and a round is started | An error should be displayed to the user indicating that the user did not exist. There should be an option to try again. | The error occurs, and the user can try again. | Pass |
| 6 | A | Left arrow key is pressed and then immediately depressed with the tank at x=10 with the terrain equation and multiplier set to 4 | The tank’s x coordinate is 5 and the y coordinate is 2.823 | The tank’s x coordinate is 5.0 and the y coordinate is 2.822566568204671 | Pass |
| 6 | B | Left arrow key is pressed and then immediately depressed with the tank at x=0 with the terrain equation and multiplier set to 0 | The tank’s x coordinate is 0 (the same) and the y coordinate is 2.054 (the same) | The tank’s x coordinate is 0.0 and the y coordinate is 2.053786266843076 | Pass |
| 6 | C | Left arrow key is pressed and then immediately depressed with the tank at x=51 with the terrain equation and multiplier set to 0 | The tank’s x coordinate is 50 and the y coordinate is 114.945 | The tank’s x coordinate is 50.0 and the y coordinate is 114.94381324735993 | Pass |
| 7 | A | Right arrow key is pressed and then immediately depressed with the tank at x=10 with the terrain equation and multiplier set to 0 | The tank’s x coordinate is 11 and the y coordinate is 3.901 | The tank’s x coordinate is 11.0 and the y coordinate is 3.900566122589183 | Pass |
| 7 | B | Right arrow key is pressed and then immediately depressed with the tank at x=500 (which is xMax) with the terrain equation and multiplier set to 0 | The tank’s x coordinate is 500 (the same) and the y coordinate is 22.799 (the same) | The tank’s x coordinate is 500.0 and the y coordinate is 22.79894445553151 | Pass |
| 7 | C | Left arrow key is pressed and then immediately depressed with the tank at x=51 with the terrain equation and multiplier set to 0 | The tank’s x coordinate is 52 and the y coordinate is 115.536 | The tank’s x coordinate is 52.0 and the y coordinate is 115.53680334672059 | Pass |
| 8 | A | Up arrow key pressed and then immediately depressed with the angle at 0 degrees. | The tank’s angle changes to 1 degree. | The tank’s angle changes to 1.0 degrees | Pass |
| 8 | B | Up arrow key pressed and then immediately depressed with the angle at 45 degrees. | The tank’s angle changes to 46 degrees. | The tank’s angle changes to 46.0 degrees | Pass |
| 8 | C | Up arrow key pressed and then immediately depressed with the angle at 270 degrees. | The tank’s angle changes to 270 degrees. | The tank’s angle changes to 270 degrees. | Pass |
| 9 | A | Down arrow key pressed and then immediately depressed with the angle at 180 degrees. | The tank’s angle changes to 179 degrees. | The tank’s angle changes to 178.99999999999997 degrees | Pass |
| 9 | B | Down arrow key pressed and then immediately depressed with the angle at 45 degrees. | The tank’s angle changes to 44 degrees. | The tank’s angle changes to 43.99999999999962 degrees | Pass |
| 9 | C | Down arrow key pressed and then immediately depressed with the angle at -90 degrees. | The tank’s angle stays at -90 degrees. | The tank’s angle stays at -90 degrees. | Pass |

**References:**

Gilmore, Stephen. Software Requirements (The University of Edinburgh), 2003. [Online] Available from: [www.inf.ed.ac.uk/teaching/courses/ip/CS2Ah0405-SoftwareRequirements.pdf](http://www.inf.ed.ac.uk/teaching/courses/ip/CS2Ah0405-SoftwareRequirements.pdf) [Accessed 22 Feb 2020].