2D Tanks Game

Software Design Document

Tyler Carr

March 6, 2020

Modified April 20, 2020

CS 225, Spring 2020

Embry-Riddle Aeronautical University

Daytona Beach campus

1 Aerospace Boulevard

Daytona Beach, FL 32114

**Introduction:**

The 2D Tanks Game is a turn-based strategy game [1] that mimics the popular flash game Big Battle Tanks [2]. The player will control a tank from a side view and can navigate their tank left and right along a randomly generated hilly terrain, trying to shoot projectiles at the opponent tank to eliminate them. The opponent tank is controlled by the computer and can have different levels of difficulty. Flash games gained popularity when Adobe purchased FutureSplash Animator [3] and created Adobe Flash [4], giving developers access to tools to create interactive animations or games [5]. Big Battle Tanks is one of many of those games. However, new games are now much more easily accessed via smartphones and are being built with new technologies, so since the popularity of Flash has died out, Adobe will be removing support for Flash later this year [6].

**Problem Description:**

The 2D Tanks Game is a game where one player takes turns with an opponent, which is controlled by an algorithm, moving their tank on a terrain of hills and firing projectiles at one another until one tank runs out of health. The player can choose the difficulty of the opponent before the game begins. If the difficulty is set to hard, the projectiles grow more powerful and accurate, and vice versa if the difficulty is set to easy. The difficulties of the opponent depend on Java inheritance, with both levels of difficulty having a different method to move the tank and fire projectiles randomly but with increasing accuracy. The game starts the first round with two tanks at full health on a randomly generated terrain. The terrain is a curvy line that goes across the screen and follows an equation generated randomly from the specifications seen in Appendix B. The player starts near the left side of the screen (), and the opponent starts near the right side of the screen (). The player starts first. The player is able to navigate their tank left and right across the terrain with the respective arrow keys, and they are able to move their tank’s cannon counterclockwise with the up arrow key and clockwise with the down arrow key. The game will handle the exceptions that may occur when the player attempts to move their tank off of the map (above the maximum set width or below 0) or tries to move their tank’s cannon inside of the tank (with the range of acceptable cannon angles being between -90° and 270°). Both tanks will only be able to move a certain distance each round. The tank shoots a projectile when the spacebar is pressed, and this projectile will follow the rules of projectile motion [7] until it reaches the same height as the terrain. See Appendix A for more information on projectile movement. After the projectile is shot, the tank cannot move until their next turn. If it hits the other tank or itself, it does damage to the tank it hits, subtracting from the current health of that tank. The amount of damage is a fixed amount depending on whether it was a “critical” hit or a regular hit. A critical hit is defined by a landed projectile that has a x coordinate within 2 units of the center of the tank, and a regular hit is within 2 to 15 units of the center of the tank. After the damage is done, then it is the opponent’s turn, which is outlined in Appendix C. The game continues until either one tank runs out of health or 50 rounds pass, where the game will end in a draw. If the player wins, they get a certain amount of in-game money depending on their performance, which is outlined in Appendix D. This money can then be used to upgrade their tank in the “upgrades” panel. See Appendix E for more information about upgrades. After the winner is done upgrading, they can play another round or quit.

**User Stories:**

1. As a user I want to have an account that I can come back to later.
   1. Completed when the user can input a username that is saved to a file and can load this information from the main menu to start a game with.
2. As a user I want to move my tank across the terrain.
   1. Completed when the tank moves left when the left arrow key is pressed, and right when the right arrow key is pressed (incremented by x+1 or x-1 every time).
3. As a user I want to move my cannon up and down.
   1. Completed when the tank’s cannon moves counterclockwise when up the arrow key is pressed and clockwise when the down arrow key is pressed. The angle of the cannon will be stored in a variable.
4. As a user I want to shoot my cannon.
   1. Completed when the cannon shoots a projectile at the angle defined by moving the cannon clockwise or counterclockwise that follows the equations in Appendix A.
5. As a user I want to upgrade my tank.
   1. Completed when the user can sort through list of available upgrades on the upgrade screen and click on each one to attempt to purchase it.
6. As a user I want to start the game when the start button is pressed.
   1. Completed when the start menu goes away when the start button is pressed and is replaced by a window where terrain and tanks will go.
7. As a user I want to see game statistics in a menu when the game is in progress.
   1. Completed when information is shown in a panel at the top of the screen while a game is in progress, including names and health.
8. As a user I want my tank to have full health upon game start.
   1. Completed when both tanks have maximum health (10 health + multiplier) upon game start.
9. As a developer I want to generate random terrain when game starts.
   1. Completed when a random terrain of hills is created upon game start.
10. As a developer I want to display player’s tank on the left side of the terrain upon game start.
    1. Completed when player’s tank appears at fixed x location (x=10) on the left side of the screen on top of the terrain upon game start.
11. As a developer I want to display opponent’s tank on the right side of the terrain upon game start.
    1. Completed when opponent’s tank appears at fixed x location () on the right side of the screen on top of the terrain upon game start.
12. As a developer I want to make sure tanks stay on top of the terrain when moving.
    1. Completed when the tanks stay on top of the terrain when moving left and right, adjusting the y coordinate according to the current x value with the random equation used to generate the hills.
13. As a developer I want to subtract health from a tank when a projectile hits a tank.
    1. Completed when the tank loses a certain number of health points when a projectile hits it, depending on whether it was critical or regular (see following user stories).
14. As a developer I want to force user to wait for when it is their turn.
    1. Completed when the player is unable to perform actions unless it is their turn (can’t move tank, can’t shoot, etc).
15. As a developer I want to differentiate between a critical hit and regular hit.
    1. Completed when a tank loses more health (5 health) when a projectile is right on top of the tank (within 2 units of the tank’s x coordinate) and loses less health (2 health) when it is slightly to the side of the tank (within 3 and 15 units of the tank’s x coordinate).
16. As a developer I want to end the game when one tank runs out of health.
    1. Completed when the game ends with a winner when one tank falls below the elimination threshold (health=0) after a turn is completed.
17. As a developer I want to end the game in a draw when the turn number exceeds a certain number.
    1. Completed when the game ends in a draw when both tanks are above the damage threshold after a turn is completed but the number of turns exceeds a set number (turn=50).
18. As a developer I want to award the winning player a certain amount of in-game money after the round ends.
    1. Completed when, if the player won, they get a certain amount of in-game money depending on their performance (number of rounds taken to eliminate the other tank, damage taken, etc) after the game ends.
19. As a developer I want to display the upgrades screen when a round ends.
    1. Completed when the game panel is replaced by the upgrades panel after the game ends, which displays all available tank upgrades.
20. As a developer I want to check player in-game balance when a player attempts to purchase an upgrade.
    1. Completed when a player receives an upgrade if they have sufficient funds and receives an error if they don’t have sufficient funds when attempting to upgrade something from the upgrades panel.
21. As a developer I want to save tank upgrades to a file.
    1. Completed when all tank upgrades are saved to a file after each upgrade is made.
22. As a developer I want player in-game money saved to a file.
    1. Completed when the player’s in-game money is saved to a file after each round.

**Problem Solution:**



When the program begins, the TankGUI class first waits for the user to select a difficulty level, enter a username, and then start the game. All of this information is passed on to the Manager class, which handles the game mechanics. Once the game begins, the Manager class creates a random curvy terrain using the equation as described in Appendix B. Then, it creates one Player object with the multipliers retrieved from a file if a username was entered. Either one “Easy” or one “Hard” difficulty opponent object is also created, depending on the user’s selection. Then the game loop runs in the MainPane class. In the loop, the player goes first, and takes turns every other turn with the opponent until one of the tanks runs out of health or 50 turns are surpassed. When it is their turn, the user can use the arrow keys to maneuver their tank across the map until they have moved the maximum distance that is allowed, and they are able to fire their cannon anytime during their turn. After they fire their cannon and it lands, then it is the opponent’s turn, in which the actions are automated with some randomness where the Easy difficulty opponent is less accurate than the Hard difficulty opponent. One thing to note is that any time the y position is needed (for example, when calculating when a projectile hits the ground, or what the y position is when a tank is moved left or right), a method involving the equation in Appendix B is used, which is in a terrain object.

User stories satisfied by each class:

* Manager: 1, 5, 13, 15, 17, 18, 21, 22
* TankGUI: 6
* MainPane: 2, 3, 4, 5, 7, 10, 11, 14, 16, 17, 18, 19, 20, 21, 22
* Tank: 2, 3, 4, 8, 13, 15, 18, 20, 21, 22
* Projectile: 15
* Terrain: 9, 12

**References:**

1. Frum, Larry. The Rebirth of Turn-Based Strategy Games (CNN), 2012. [Online] Available from: [www.cnn.com/2012/07/23/tech/gaming-gadgets/rebirth-turn-based-strategy-games/index.html](http://www.cnn.com/2012/07/23/tech/gaming-gadgets/rebirth-turn-based-strategy-games/index.html) [Accessed 12 Feb 2020].
2. BigWig Media. Big Battle Tanks (2dplay), 2017. [Online] Available from: 2dplay.com/#/game/big-battle/tanks [Accessed 12 Feb 2020].
3. Web Design Museum Team. FutureSplash Animator (Web Design Museum), 2020. [Online] Available from: [www.webdesignmuseum.org/old-software/macromedia-flash/futuresplash-animator](http://www.webdesignmuseum.org/old-software/macromedia-flash/futuresplash-animator) [Accessed 12 Feb 2020].
4. Adobe. Adobe Flash Player, 2020. [Online] Available from: get.adobe.com/flashplayer/about [Accessed 12 Feb 2020].
5. Mouli, Mohua. In loving memory of flash games (The Daily Star), 2018. [Online] Available from: [www.thedailystar.net/bytes/news/loving-memory-flash-games-1653250](http://www.thedailystar.net/bytes/news/loving-memory-flash-games-1653250) [Accessed 12 Feb 2020].
6. Adobe Corporate Communications. Flash & The Future of Interactive Content (Adobe Blog), 2017. [Online] Available from: theblog.adobe.com/adobe-flash-update [Accessed 12 Feb 2020].
7. Shene, Ching-Kuang. Programming Example: Projectile Motion (MTU), 1997. [Online] Available from: pages.mtu.edu/~shene/COURSES/cs201/NOTES/chap02/projectile.html [Accessed 12 Feb 2020].

**Appendix A:**

The tank projectiles will follow the following equations for the x and y coordinates, updating until the y level of the terrain is reached and then stopping.

* 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.

**Appendix B:**

The terrain will be created at the beginning of each round by generating random values a, b, and c, which are defined below. This equation will then be used to calculate the height of the terrain any time that it is needed.

* 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.

**Appendix C:**

The user will be able to choose the difficulty of their opponent before the game begins. For both “easy” and “hard” difficulties, the opponent’s movement will be a random amount between 5 and the maximum number of units that they are allowed to move with their upgrades in the direction of the player tank. The hard difficulty opponent uses multiple different factors to decide how to aim its cannon, such as the y coordinate 15 units to the left and right (to detect hills), the horizontal distance between the two tanks, the current projectile velocity multiplier, and more. It will have random multipliers between the values of 5 and 10 applied at the beginning of each new round. The easy difficulty opponent uses these same concepts, but a random value between -20 and 20 degrees is added on to the angle, hill detection is omitted, and no multipliers are applied to make it easier to defeat. If the player is within the calculated random distance of the player tank, then the opponent doesn’t move at all. The maximum amount of spaces to move is 50. For both opponents, the angle that the cannon needs to be adjusted to in order to land a projectile directly on the player will be calculated. If the angle turns out to be an illegal value, then the angle is changed to the closest maximum value and then fired. For example, if the angle is -100 degrees, it would be changed to -90 degrees because that is the closest limit.

**Appendix D:**

When the player wins a round, the amount of in-game money they receive will be calculated with the following 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

**Appendix E:**

The upgrades panel displays all of the available tank upgrades in a menu after the game ends. If the player has enough in-game money, they can purchase their desired upgrade. The available upgrades and how they are applied are as follows:

* Health: The number of upgrades a user has is added on to the default health of 10.
* Initial projectile velocity: The number of upgrades a user has is doubled and then added on to the default velocity of 75.
* Projectile damage to tanks: When a projectile hits a tank, the number of upgrades a user has is multiplied by 0.1, incremented by 1, and then that sum is multiplied by the default damage amount.
* Tank travel distance per round: The number of upgrades a user has is multiplied by 10 and then added on to the default travel distance of 50.

Upgrades are stored in a file and are associated with a username. Each time the user purchases an upgrade, one is added to the total number of times they’ve upgraded that particular item. During the next game, the total number of times they’ve upgraded each item is added on to the default amount of each item upon game start. For example, if the user had upgraded their health three times already and purchases another health upgrade, for the next game, their health would be the default health value plus four extra points. Upgrades are retrievable upon initial program start by entering a username into a text field. The maximum amount of upgrades that is allowed per item is 10.