Requirements and Analysis Document for Tank Wars

Version: Iteration 1

Date: 2018/03/26

Authors: Patricia Zabecka, Carl Lundborg, Thomas Jinton, Adam Kjäll, Leyla Arale.

(This version overrides all previous versions.)

Introduction

The purpose of this application is to entertain the user(s) with a simple but fun game based on the retro version of the artillery game *Tank Wars*.

Requirements

User Interface

The game Tank Wars will have a menu and it will be the first thing you see when you start the game application. The menu will have basic settings and optional buttons where you can choose what the conditions for the game will be like. Using the the keyboard you can navigate around the menu and when you have entered the given conditions that you want for the game, you can start the game by pressing the Start-button.

After the game is started you will see the playing field. There will be a landscape with a terrain and the tanks will be located on that terrain. A pop-up dialog will appear every time a gameround is finished, the dialog will show the players´ scores and a next-button that will take the player to the next gameround.

Functional Requirements

The player will be able to do the following:

- Choose and set the game according to provided optional game settings at the start menu. The game will have the following options:

    - Select difficulty
        - Easy, medium and hard are the available difficulties.
        - The hard option provides the gameplay with a intense wind that makes it harder to move a tank in a certain direction.
        - The intensity of the wind increases depending what difficulty is chosen.
        - The wind direction and intensity will be displayed in the gameplay.
    - Select number of game rounds.
        - The user can select between 1-5 rounds.
        - Default number of round is 3.
    - Select number of players to play against.
        - The player can choose between playing against other players or players generated by the computer (AI).
        - The default number of player is 2.
        - The player can customize the tank by choosing a color for the tank and a name that will be displayed during the gameplay when it is the player’s turn.

Default settings will be provided so that the game can be started immediately without affecting the gameplay. All options will be selected by default so that the player can start the game without having to select anything.

Each player controls a tank.

- The tank can be moved forward and backwards only if the player has enough fuel.
- The fuel will be displayed with numbers in the gameplay and change in relation to how much you try to move your tank, when there is no more fuel you can still aim and shoot with the tank.
- The player can aim and fire the tank even if there is no fuel left.
- The tank can shoot.
- The shot can destroy other tanks but also the terrain giving it a new shape.
- The shot will travel in the direction it is aimed and with a certain velocity.
- The terrain changes its shape.
- A new terrain is drawn at the start of each game round.
- The shape of the terrain changes during the game according to where the players shoot.
- Every tank has health points.

A tank can be damaged by:

- A shot fired by another tank.
- If the tank shoots itself somehow.
- A tank dies when there is no terrain.
- Getting stuck in a deep terrain area.
- The damage will be shown by decrease in health points.
- If a tank's health points equals to zero the tank will die.

At the end of each game round players are awarded for kills and wins.

- After the last rounds the sum of the collected points from every round is calculated for each player. The player with the biggest number of points is the winner of the game.
- Killing other tanks awards the player with points.
- No time constraints for the rounds.
- The game is turn based. The next player is chosen by the application according to a randomly generated order.
- New turn after each shot.
- Player can end the game during a round.
- A gameplay can be saved.
- A round ends when only one tank is left on the field.
    - This tank will be the winner of the round.

Non-functional Requirements

A Pop up box is displayed when a game round is finished, the box informs the players about the point distribution and the winner of the current gameplay. Other non-functional requirements are listed and specified under the following categories:

Usability

- The instructions for the game are easy, almost intuitive.
- If a player is not familiar with the game it should not be hard to understand how to play the game.

- Easy language.
- Only a few interfaces.

Supportability

- Help im terms of instructions explaining how to play the game will be available.

Performance

- The game will run without any problems or interference. The players actions will not cause the game to crash or slow down.
- The game will have a certain tempo/speed that is relevant for its 2D animation. Playing the game will be expected to work.

Maintainability

- If something has an error or that it is incomplete then those problems will be fixed eventually. Possible improvements to the program can also be implemented and the game can be expanded.

Testability

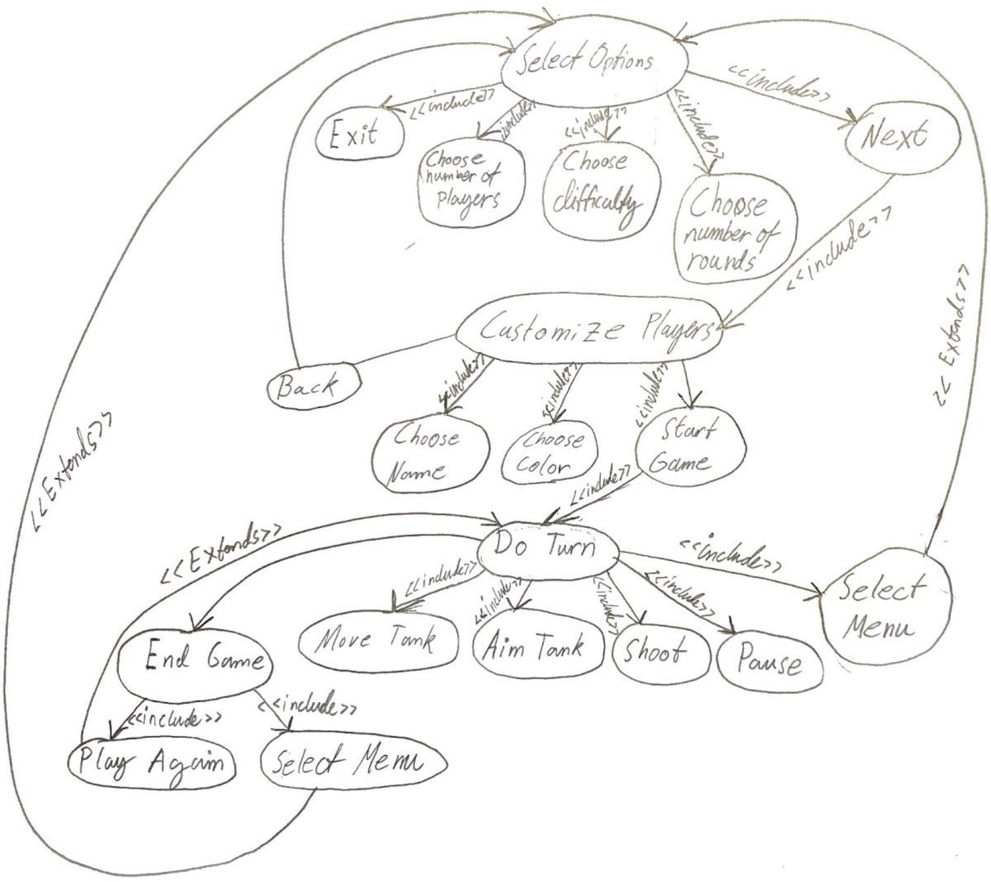
- The code will be possible to test and also modify if needed.

Availability

- The application would be available for both Windows and Mac users.
- Implementation
- The code will be written in Java.

Use Cases

The following use case diagram shows the different use cases that the player will be involved with when playing the game.



Use Case listing

Select Options

When selecting options the following use cases are available for the player to interact with:

- Choose number of players.
- Choose difficulty.
- Choose number of rounds.
- Press “Nex”t.
- Press “Exit”.

Customize Players

When customizing players for the game the following use cases are available for the player to interact with:

- To main menu - the player starts the game.
- Press back.
- Choose name of a specific player.
- Choose color for a specific player.

Do Turn

During the actual gameplay the player could do the following:

- Move the tank.
- Aim the tank.
- Pause the game.
- Shoot - the player fires the tank.
- Press “Main Menu”.

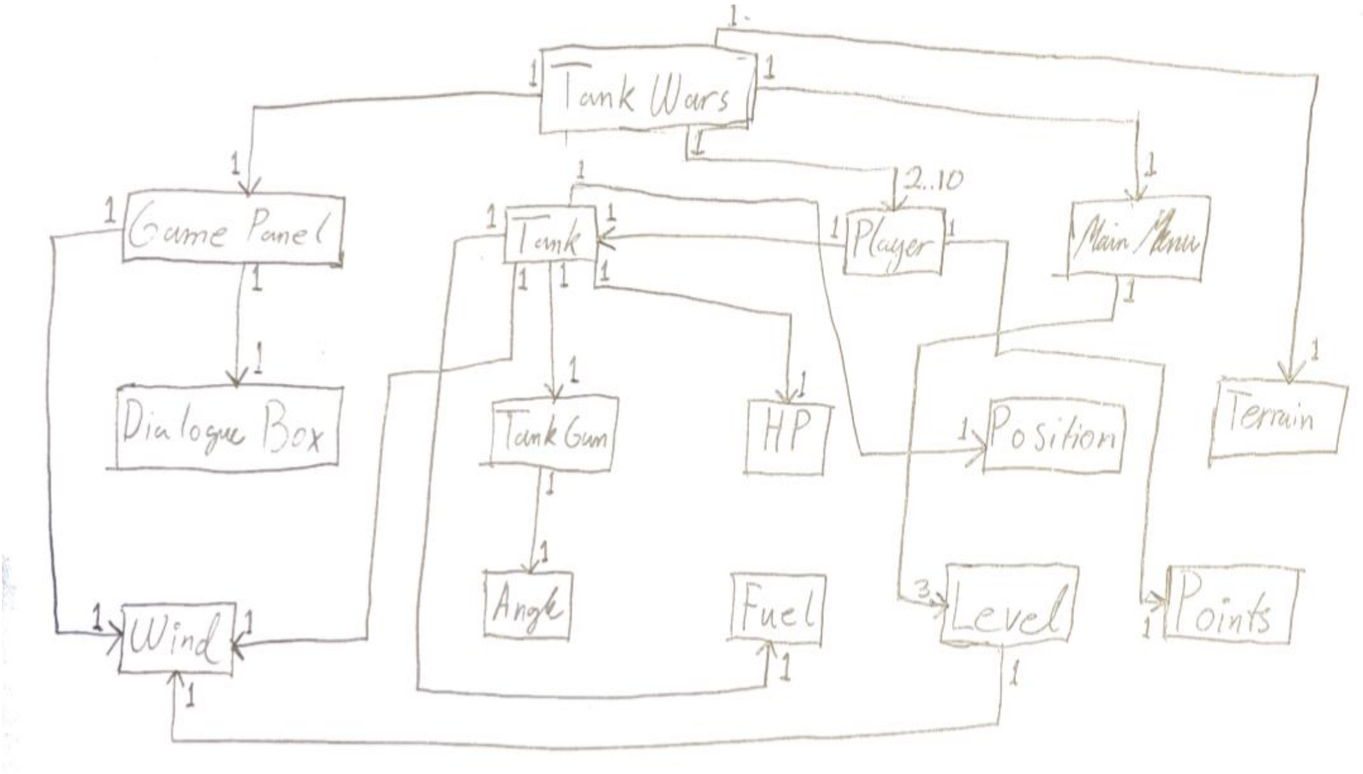
End Game

When the game ends the user could do the following:

- Press “Play again”.
- Press “Main Menu”.

See Appendix A for a more detailed summary of the cases and a table describing their flow of events.

Domain Model



Class responsibilities

TankWars

Overall representation of the game. The game-loop will be taken care of in this class, therefore all game objects will be created here.

Player

Represents the player. A player has a score, a name and a color.

GamePanel

Represents the frame that will be displaying information during the gameplay such as the direction of the wind.

DialogBox

Information about the player’s points will be stored here so that it could be displayed when a game ends.

Tank

Represents the tank that the player can control. It has a position, health points, a fuel level and the tank can move. The tank holds also a weapon that is represented by the TankGun class.

TankGun: Representation of the tank’s gun. The gun can be fired and aimed. The gun could be aimed according to different angles, therefore this class holds an Angle class.

Angle

Represents the angle of the tank’s weapon.

HP

Represents a tank’s health points, these points will decrease after a tank has been damaged.

Fuel

A representation of the fuel that is needed for a tank to be able to move.

Level:

The game has three available levels: easy, medium and hard. This class holds the Wind class because the intensity of the wind changes depending on which level the player has selected.

Position

A tank has a position which is represented in terms of x and y coordinates.

Points

The player can collect points during the game. Points of each game round will be summed at the end of the game.

Wind

The wind has a direction and an intensity that makes it harder for the tank to move.

Terrain

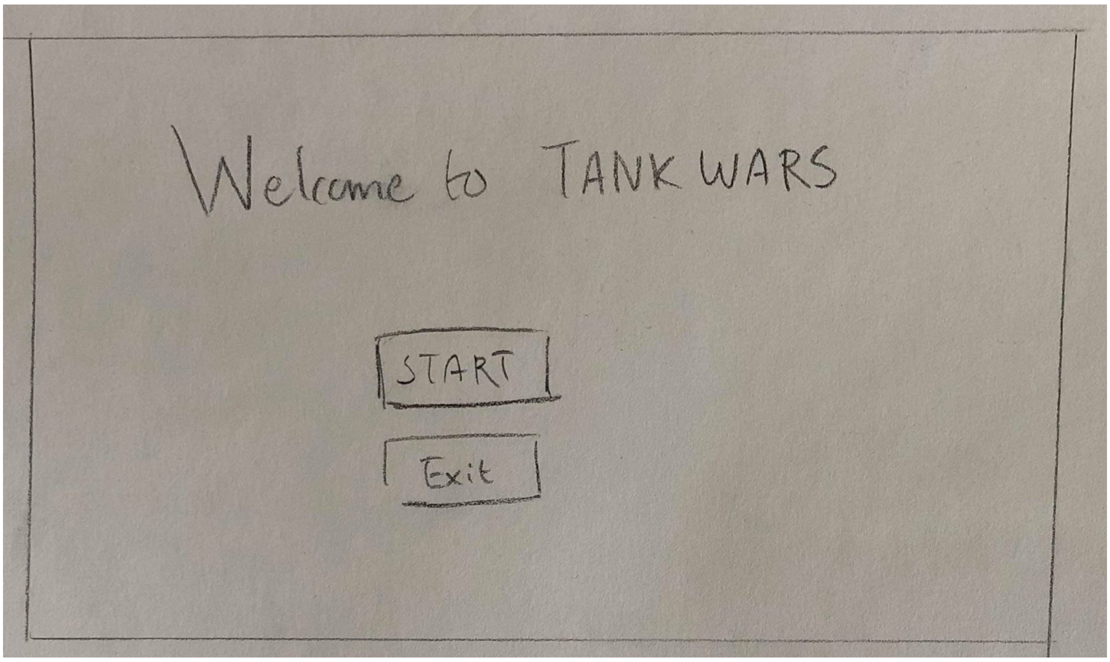
The landscape for the tanks to move on and to interact with each other.

MainMenu
Holds all information that the start menu will show for the user, for instance the level of the game.

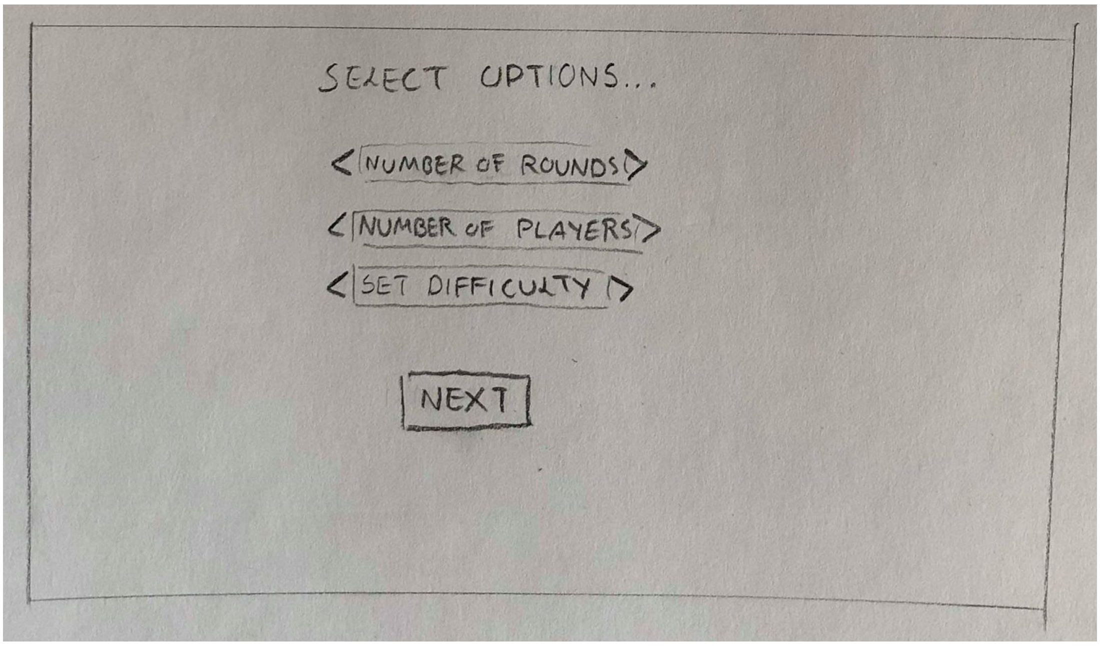
Prototype

A simple sketch of how the interface of the game application will look like. The application has 5 scenes in total.

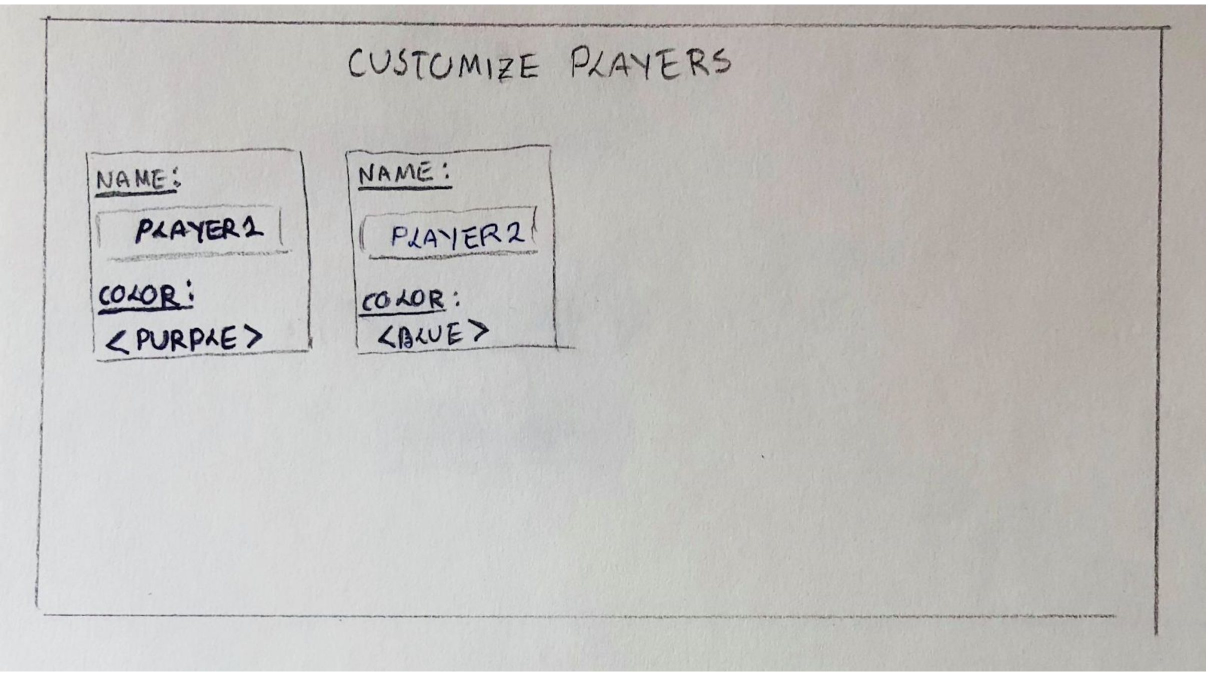
The first scene represents the start screen that appears when the player starts the application.



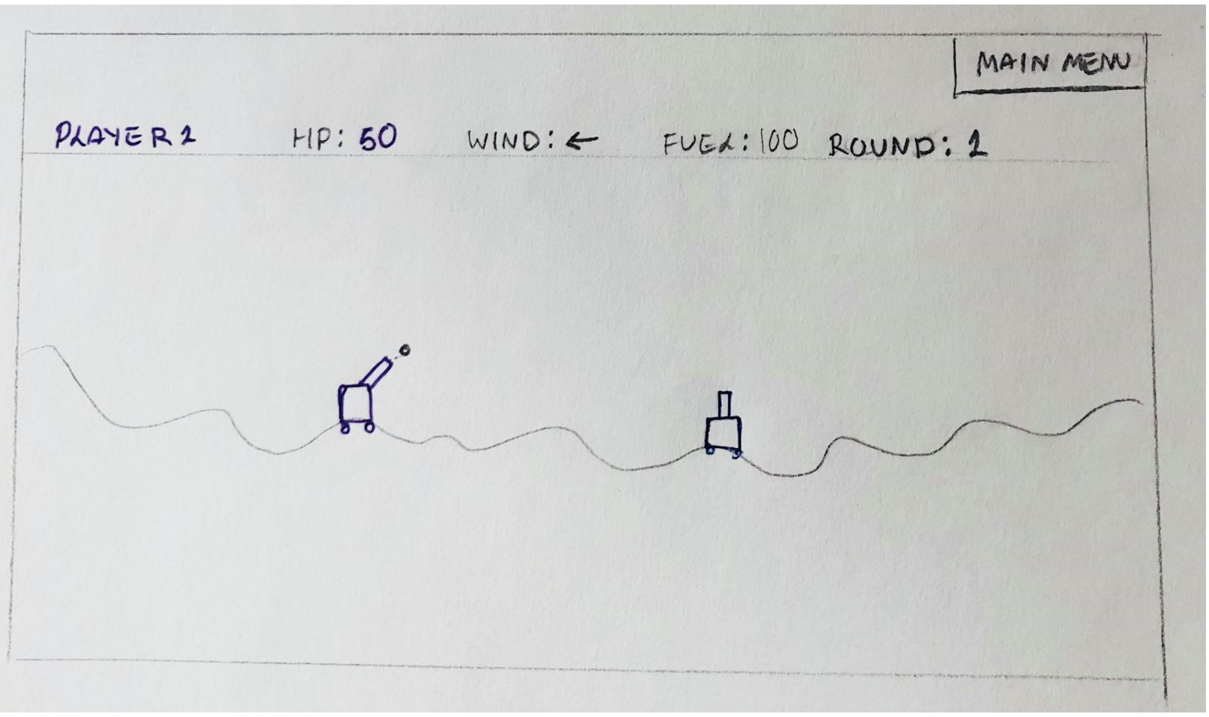
The second scene allows the user to select some options for the game.



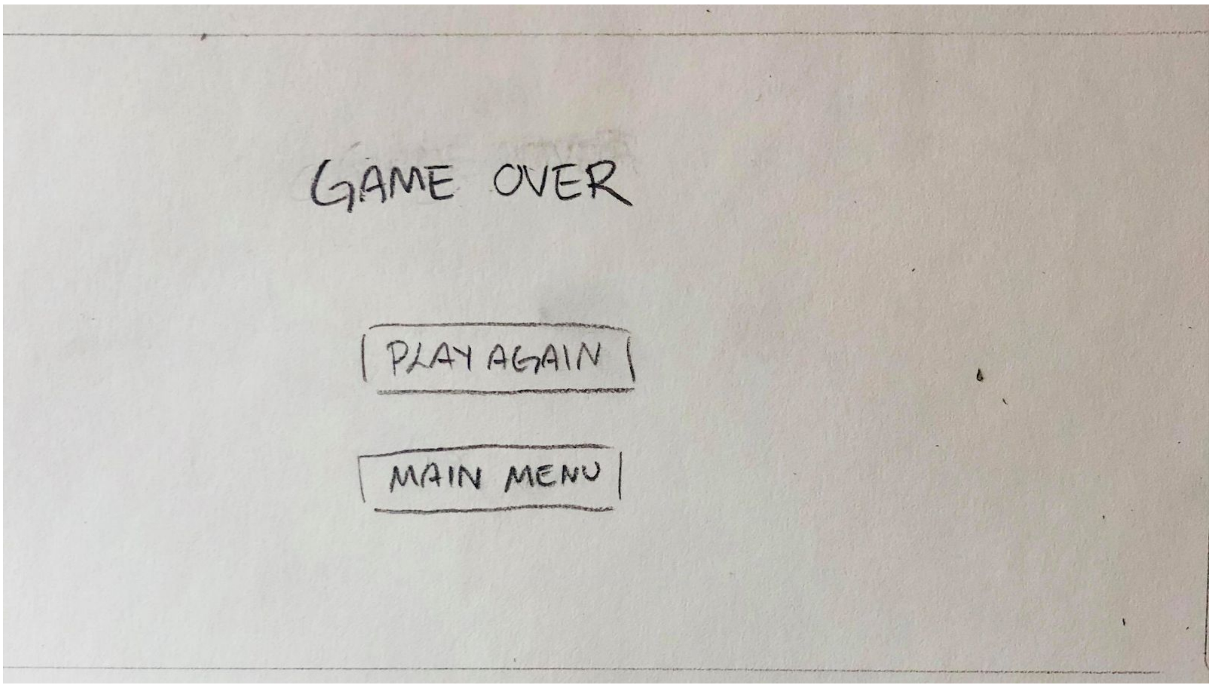
After selecting the options the player is taken to another screen where some basic player customization can be done such as changing the layer's name and color.



The gameplay will look something like this. The current player will be recognized by the chosen color and name. Useful information such as health points and the wind direction will be displayed at the top of the game world.



When all the game rounds are finished the player will be taken to this scene.



Appendix A - Use Cases

Use Case: To Main Menu

Summary: The player starts the game application.

Priority: High

Extends: Start program.

Includes: --

Participators: The player.

Normal flow of events

	Actor	System
1		
2		

Alternative flow

	Actor	System
1		
2		
3		

Use Case: Back

Summary: The player can return to main menu.

Priority: High

Extends: Select Options.

Includes: Customize Player

Participants: The player.

Normal flow of events

	Actor	System
1	The player navigates with arrow keys up/down to select “Back”.	
2		The “Back” gets highlighted
3	The player presses enter to select “Back”	
4		The “Customize player” window gets replaced with “Main menu” window.

Use Case: Choose players color

Summary: The player can select their own color.

Priority: High

Includes: Customize Player.

Participants: The player.

Normal flow of events

	Actor	System
	The player navigates with arrow keys up/down to select “Choose color”.	
		The option “Choose color” gets highlighted
1	The player presses enter key to enter the option “Choose color”.	
3		A random default color is shown as text and the option box will change to that color.
4	The player presses enter key to choose the selected color.	
5		The option “Choose color”’s highlight gets removed

Alternate flow of events

	Actor	System
1	The player presses enter key to select the option “Choose color”.	

2		The option “Choose color” gets highlighted
3	The player presses enter key to enter the option “Choose color”.	
4		A random default color is shown as text and the option box will change to that color.
5	The player presses left or right arrow key to choose between the available colors	.
6		Display which color is chosen as text and also changes the option box to the selected color
7	The player presses enter key to choose the shown color.	.
		The option “Choose color”’s highlight gets removed

Use Case: Choose players’ name

Summary: The player can select their own name.

Priority: High

Includes: Select Option.

Participants: The player.

Normal flow of events: The player chooses default name.

	Actor	System
1	The player navigates with arrow keys up/down to select “Choose name”.	
2		The option “Choose name” gets highlighted
3	The player presses enter key to enter the option “Choose name”.	
4		A default name is shown as text under “Choose name”.
5	The player presses enter key to accept the default name	
6		The option “Choose name” is no longer highlighted.

Alternate flow of events: The player types in a name.

	Actor	System

1	The player navigates with arrow keys up/down to select “Choose name”.	
2		The option “Choose name” gets highlighted
3	The player presses enter key to select the option “Choose name”.	
4		A default name is shown as text.
5	The player types in their name with the keyboard.	
6		The default name is now replaced with the one the player typed.
7	The player presses enter key to confirm their name.	
8		The “Choose name” option is no longer highlighted.

Use case: Aim Tank

Summary: The user presses the arrow keys (“up”, “down”) and the tank will aim in desired direction.

Priority: High

Includes: DoTurn

Participants: The player

Normal flow of events: The player aims the tank gun.

	Actor	System
1	Player holds down the up/down arrow keys	
2		Tank gun moves in the desired direction left or right
3		

Alternate flow:
The player tries to aim down/up but the tank gun is already at minimum/maximum angle.

	Actor	System
1	Player holds down the up/down arrow keys	
2		Tank gun stays at 90 degrees angle left or right

Use case: Move

Summary: The user presses the arrow keys (“left”, “right”) and the tank will move in desired direction.

Priority: High

Includes: DoTurn

Participants: The player.

Normal flow of events: The player moves the tank.

	Actor	System
1	Player holds down the left/right arrow keys	
2		Tank moves left or right on the terrain where the game is played
3		Fuel level decreases by numbers in a field showed at the top right corner

Alternate flow:
The player tries to move the tank, however the steepness in terrain doesn’t allow it to be moved.

	Actor	System
1	Player holds down the left/right arrow keys	
2		Tank stays at the same position

Use case: Pause

Summary:
The player presses the pause button and the game pauses.

Priority: High

Extends: --

Includes: DoTurn

Participants: The player

Normal flow of events: The player presses the pause button.

	Actor	System
1	Player presses “pause button” with left mouse button.	
2		Game freezes and the text “Game Paused” is shown on top of game field.

Use case: Select Menu

Summary: The user presses the “main menu” button and the select options window comes up.

Priority: High

Extends: --

Includes: DoTurn

Participants: The player

Normal flow of events:
The player presses the “main menu” button.

	Actor	System
	Player presses “Main Menu” button with left mouse button.	
1		A dialogue pops up with the text: “Are you sure that you want to quit current game?” and two buttons: “yes” and “no”.
2	Player presses “yes” button with left mouse button.	
		Pop-up window disappears and a transition animation plays by showing the select options window with a “curtain-dragging” effect.

Use case: Shoot

Summary: The user presses the shoot-key (unknown for now), the tank will shoot and the game will check if something was hit.

Priority: High

Extends: StartGame (RunGame)

Includes: DoTurn

Participants: The player

Normal flow of events:
The player fires and hits the ground but misses the targeted enemy tank.

	Actor	System
1	Player presses shoot-key {space}.	
2		**Shoot** animation plays.
3		Checks if an **enemy tank** was hit.
4		The **shot** removes the **terrain** in the specified area.
6.		Animates the new **shape** of the ground.

Alternate flows:
The player fires and an enemy tank is hit and dies. There are still other enemy tanks left.

	Actor	System
1	Player presses shoot-key.	
		Shoot animation plays.
2		Enemies HP is decreased by numbers, which shows in a box in the top right corner
4		An enemy tank was hit - play “die” animation - and remove tank from field.
5.		Increase player’s points by numbers.
9.		Next players turn

Alternate flows:
The player fires and an enemy tank is hit and there are no other enemy tanks left.

	Actor	System
1	User presses shoot-key.	
		Shoot animation plays.
2		Enemies HP is decreased by numbers, which shows in a box in the top right corner
4		An enemy tank was hit - play “die” animation - and remove tank from field.
5.		Increase player’s points by numbers.
9.		Game ends, winner is shown in a pop up dialog on top of game field.

Alternate flow:
The player presses the “main menu” button.

	Actor	System
	Player presses “Main Menu” button with left mouse button.	
1		A dialogue pops up with the text: “Are you sure that you want to quit current game?” and two buttons: “yes” and “no”.
	Player presses “no” button with left mouse button	
		Dialogue window disappears.

Use case: Play again

Summary: The user presses the “play again” button and the round with the same options restarts.

Priority: High

Extends: Do Turn

Includes: End Game

Participants: The player

Normal flow of events:
The player presses the “play again button”

	Actor	System
		“play again” button is marked with red color.
1	Player chooses the “play again” by pressing enter.	
2		Game over window plays transition animation by being dragged upwards out of view. The game plan lies under the window.

Use case: Select Menu

Summary: The user presses the “main menu” button and the select options window appears.

Priority: High

Extends: Select Options

Includes: End Game

Participants: The player

Normal flow of events:
The player presses the “main menu” button.

	Actor	System
	Player goes to “main menu button” by pressing right arrow.	
1		Previous option button changes color from red to white. Main menu button changes from white to red.
2	Player chooses the “main menu” by pressing enter.	
		Game over window plays transition animation by being dragged upwards out of view. The “select option”

		window lies under the window.

Use Case: Choose number of players

Summary: The user can select the number of plays for the game.

Priority: High

Includes: Select Options.

Participants: The player.

Normal flow of events:
The player increases number of players.

	Actor	System
1.		“<select players>” is colored.
2.		Default number of players is selected to 2.
3.	The player presses the right-key to increase number of players. The key is pressed two times	
4.		Selected number of players changes to 4.

Alternative flow of:
The player decreases the number of players to one.

	Actor	System
1.		“<select players>” is colored.
2.		Default number of players is selected to 2.
3.	The player presses the left-key to decrease number of players. The key is pressed once.	
4.		Selected number of players changes to 10.

Alternative flow of
The player increases number of players and then decreases the number.

	Actor	System
1.		“<select players>” is colored.
2.		Default number of players is selected to 2.
3.	The player presses the right-key to increases number of players. The key is pressed twice.	
4.		Selected number of players changes to 4.
5.	The player presses the left-key to decreases number of players. The key is pressed once.	
6.		Selected number of players changes to 3.

Alternative flow of
The player doesn't want to change the number of players.

	Actor	System
1.		“<select players>” is colored.
2.		Default number of players is selected to 2.
3.	The player is content with this number and moves forward to the next option by pressing the down key.	

Use Case: Choose Difficulty

Summary: The user can select the number of plays for the game.

Priority: High

Includes: Select Options

Participators: The player.

Normal flow of events

	Actor	System
1	Player selects the option “Difficulty”.	
2		The options changes color from white to red. Default difficulty “Easy” is displayed.

Alternative flow

	Actor	System
1	Player selects the option “Difficulty”.	
2		The options changes color from white to red. Default difficulty “Easy” is displayed.
3	Player presses down the right arrow-key	
		The difficulty changes from Easy to Medium, from Medium to Hard and from Hard to Easy.

Use case: Next

Summary: The user presses the next key and is redirected to the tank’s customization page.

Priority: High

Extends: Select Options

Includes: Customize Player

Participants: Thel player

Normal flow of events:
The player presses the next key and the customization page appears.

	Actor	System
1	Player scrolls/navigates to the next-key	
2		Next-key changes color
3	Player presses the next-key	

4		Select options menu disappears
6.		Customization page appears

Use case: Choose number of rounds

Summary: The player can choose the number of rounds the game will be played.

Priority: High

Includes: Customize Player

Participants: The player

Normal flow of events:
The player presses the right arrow keys left or right to choose from 1 to 10 rounds.

	Actor	System
1	Player presses right arrow key	
2		Displays the next number within the range [1-10]. If players continues pressing right arrow key the number displayed will loop back around to 1 after it reaches 10.

Alternate flow:
The player presses left arrow key.

	Actor	System
1	Player presses left arrow key	
2		Displays the previous number within the range [1-10]. If players continues pressing left arrow key the number displayed will loop back around to 10 after it reaches 1.

Use case: Exit

Summary: The user scrolls down to the “exit game” option and shuts down the program.

Priority: High

Includes: Select options

Participants: The player

Normal flow of events:
Player presses exit game and the game shuts down.

	Actor	System
1	Player uses down arrow key to scroll down to the button “exit game”.	
2		Every button in menu goes from white to red, and back to white as the player scrolls through the options.
3	Player is at the button “exit game” and presses enter key.	
		The game shuts down completely.