Tank Game Documentation

Summer 2021

Simon Wu

Student ID

CSC413-01

[csc413-su21/csc413-tankgame-swu465: csc413-tankgame-swu465 created by GitHub Classroom](https://github.com/csc413-su21/csc413-tankgame-swu465)

Table of Contents

[1 Introduction 3](#_Toc522827688)

[1.1 Project Overview 3](#_Toc522827689)

[1.2 Technical Overview 3](#_Toc522827690)

[1.3 Summary of Work Completed 3](#_Toc522827691)

[2 Development Environment 3](#_Toc522827692)

[3 How to Build/Import your Project 3](#_Toc522827693)

[4 How to Run your Project 3](#_Toc522827694)

[5 Assumption Made 3](#_Toc522827695)

[6 Implementation Discussion 3](#_Toc522827696)

[6.1 Class Diagram 3](#_Toc522827697)

[7 Project Reflection 3](#_Toc522827698)

[8 Project Conclusion/Results 3](#_Toc522827699)

# Introduction

This term project is a game with two players that fight against each other using tanks.

## Project Overview

We needed to create two tanks that could shoot each other, where the bullets they shoot could collide into walls, and the enemy tank. We also needed to create three different power ups that would influence the flow of the game. One power up can heal you back to full health, another one will give you rockets you can fire that will do twice the damage of a normal tank shell, and the last power up will give your tank twice the speed.

## Technical Overview

For this project, we were given skeleton code that we had to fill out the missing pieces until it became a fully functioning game. it was very helpful to use abstractions for creating the necessary classes to build this project. It was recommended to create an abstract class “GameObject” that would be extended by the “Tank”, “Bullet” classes, “Power Up” classes, and the “Wall” classes because many of these classes share the same methods, albeit some minor changes to fit their intended purpose.

By making all of these have the base of “GameObject” I was able to create an ArrayList of GameObjects to make it easier to keep track of everything that would be created and drawn in the main program. The only exception is that the Tank classes have their own ArrayList of GameObjects for the bullets that they shoot. This made it easier for me to keep track which tank was shooting which bullet. To check for tank, and bullet collision, I created a new class called “CollisionDetection”. This class has three functions that require parameters of both tanks, and the ArrayList of GameObjects. One function will check for tank colliding with another tank, tank colliding with walls, and tanks colliding with power ups. The other two functions keep track of the bullets shoot by each tank.

## Summary of Work Completed

|  |  |
| --- | --- |
| Requirement | Fulfillment |
| Starting screen |  |
| Ending screen |  |
| Two players |  |
| Tanks can move forward and backward |  |
| Tanks can rotate |  |
| Split screen |  |
| Mini map |  |
| Health bars for tanks |  |
| Number of lives for each tank |  |
| 3 different power ups |  |
| Unbreakable walls |  |
| Breakable walls |  |
| Bullets collides with walls |  |
| Bullets collides with tanks |  |
| JAR file in the JAR folder |  |
| README.md filled out |  |

# Development Environment

IntelliJ Ultimate Edition

Openjdk 16: version 16.0.1

# How to Build/Import your Project

To import the project, “open project” and pick the folder “csc413-tankgame-swu465”. To build the project through IntelliJ, right click the Launcher.java and pick the “Build Module ‘csc413-tankgame-swu465’” option.

To build the JAR file in IntelliJ, go to “Project Structures, under “Project settings” go to “Artifacts”. Add a JAR file with the option “from Module Dependencies”

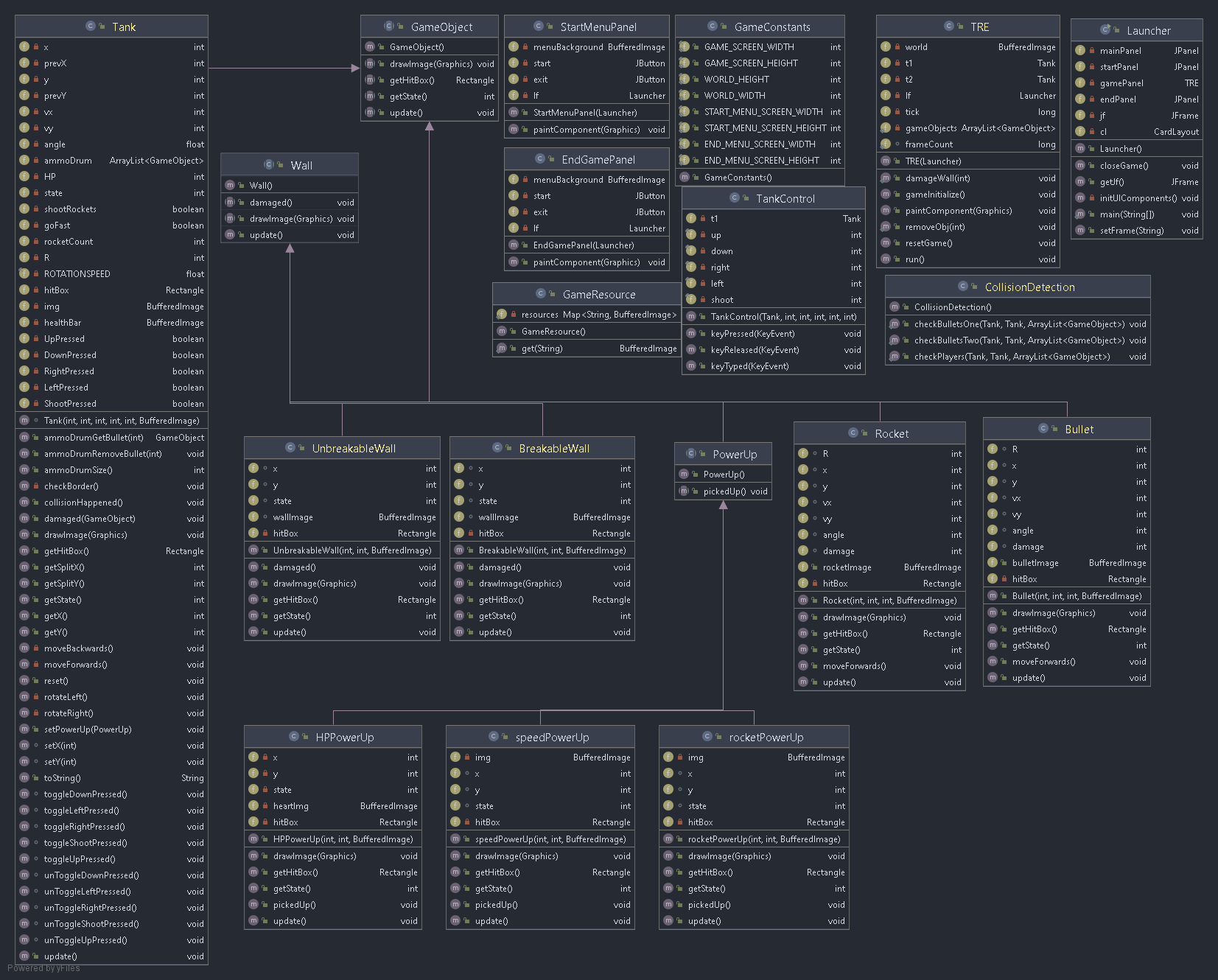
# How to Run your Project

# Assumption Made

The players know the controls to move and shoot.

# Implementation Discussion

## Class Diagram



# Project Reflection

This project had me literally pulling out my hair trying to figure out bugs. Some of them were very silly and I needed another pair of eyes to look at my code, only to find out it was something simple that I forgot. This project really makes me appreciate the polish that I see in good games because of how much work can go into something as simple as a tank war game. But at the same time, I feel like I will be more critical of some of the other games that I play which can have silly bugs. Like how someone fishing in a certain spot in the world can crash the entire server for an MMORPG.

# Project Conclusion/Results

The game will run and play correctly. There are some unexpected exception errors that will pop up when playing, but the one that is left does not interrupt the gameplay from the testing that I have done. Overall, I am satisfied that I got this in working condition that fulfil the necessary requirements for this project. This project also made me very interested in how old curtain fire games are structured, and how does it compare to what I have done for this project.