CSC 413 Term Project Documentation

Summer 2023

Ruxue Jin

923092817

Class Section:01

GitHub repository Link:

csc413-SFSU-Souza/csc413-tankgame-RuxueJ: csc413-tankgame-RuxueJ created by GitHub Classroom

Table of Contents

1 Introduction

1.1 Project Overview

This term project is Tank Game with two players fight with each other. Each tank has three lives, each life has 100 points blood and 100 bullet. The environment of world map includes two types of breakable walls, unbreakable wall, grass, grass, and river. There are four types of powerups. One is bullet supply, adding 100 bullets to the tank. One is blood supply, adding 20 points of blood. One is transformed bullet; bullets can be shot quickly and powerful. The last one is invisible. Any tank who hits an invisible powerup will be invisible for 15 seconds.

Any tank who loses all three lives will lose the game.

After one tank loses, user can choose whether exit the game or restart.

1.2 Introduction of the Tank game

Menus package includes an abstract class Screen. StartMenuPanel, EndMenuPanel, and PauseGamePanel extend Screen.

Game package includes classes of game objects, such as Tank, Bullet, 4 powerups, and Walls, Grass, and River. It contains Animations and Sound class as well.

Resources package includes ResourceManager that loads all images, sound, animation resources.

The Launcher class launches the game. It sets main panel to its jframe and set a cardLayout to switch subpanel such as startPanel, gamePanel, and endPanel.

gamePanel is an object of GameWorld class. It has a list of Game Objects. And it runs in another thread to keep the game running smoothly. Wall, River, Grass, Tank, Bullet, BulletSupply, BloodSupply, Invisible, Transform extend GameObject class. Four powerups (BulletSupply, BloodSupply, Invisible, and Transform) implements Powerup interface, and implement applyPowerUp method.

GameConstants contains all constains in the Game.

2 Development Environment

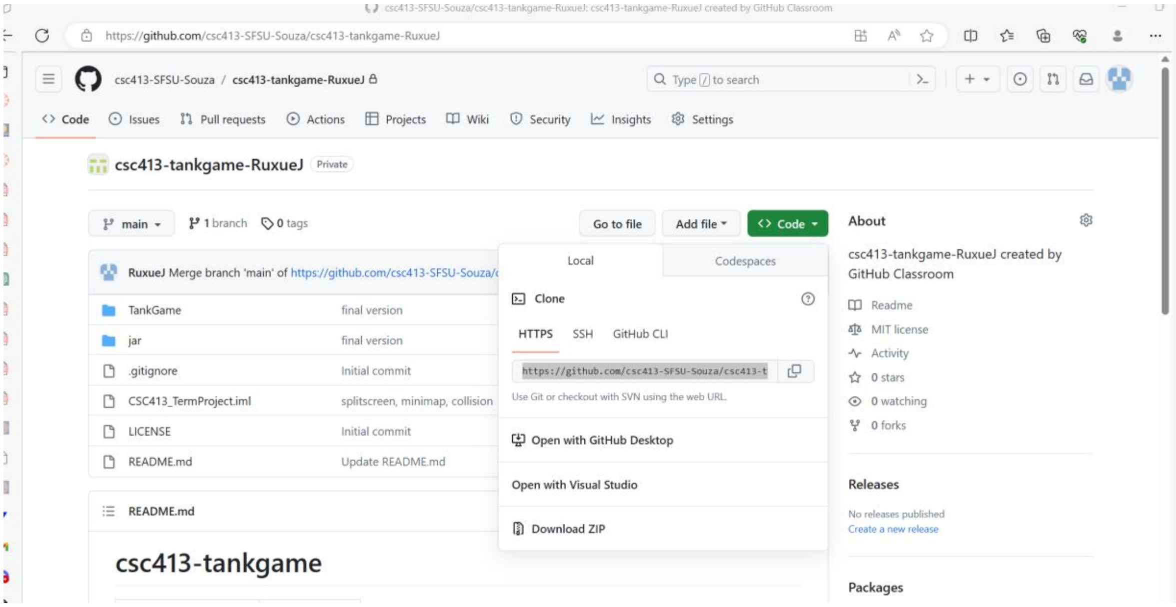
a. Version of Java version: 17.0.6
b. IDE Used: IntelliJ IDEA 2022.3.2(Ultimate Edition)
c. Any special libraries used or special resources and where I got them from.
   I got animations, sound, and some images from Canvas. Other images such as grass and river are from MidJourney.

3 How to Build/Import your Project

a. How to build project in IntelliJ

csc413-SFSU-Souza/csc413-tankgame-RuxueJ: csc413-tankgame-RuxueJ created by GitHub Classroom

Ctrl + click to follow the link.



RuxueJ Merge branch 'main' of https://github.com/csc413-SFSU-Souza/c

TankGame	final version
jar	final version
.gitignore	Initial commit
CSC413_TermProject.iml	splitscreen, minimap, collision
LICENSE	Initial commit
README.md	Update README.md

https://github.com/csc413-SFSU-Souza/csc413-t

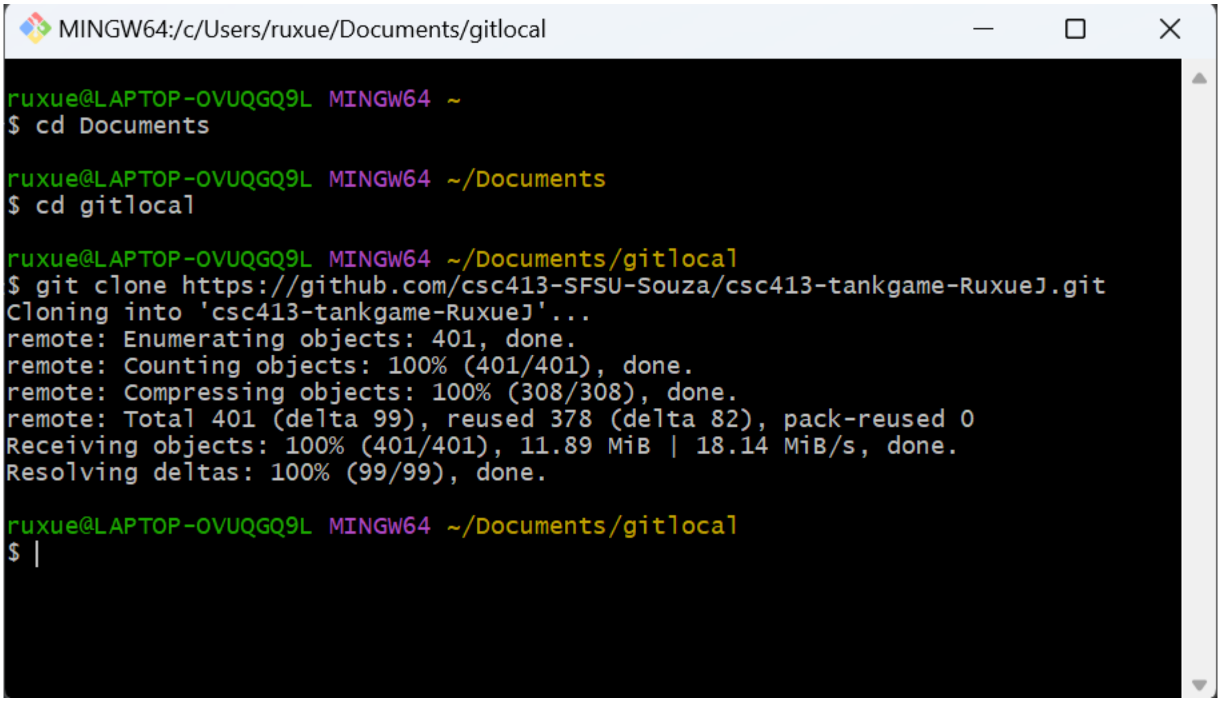
Use Git or checkout with SVN using the web URL.

csc413-tankgame-RuxueJ created by GitHub Classroom

Click the green “Code” button on my repo’s home page. Then copy HTTPS.

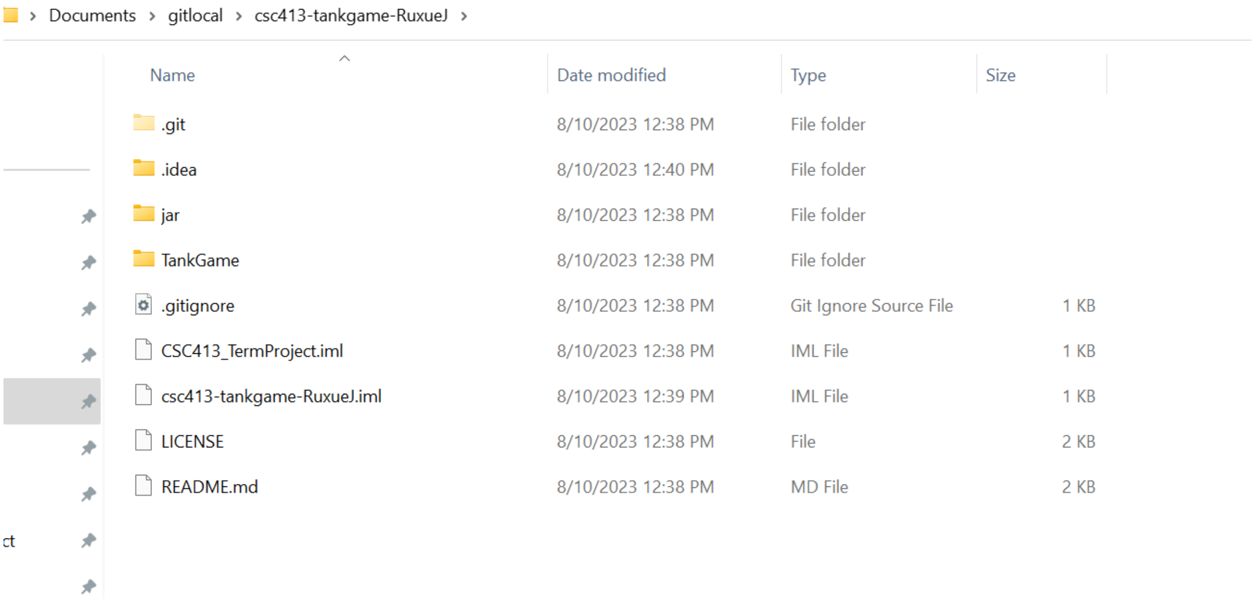
In the terminal, cd to the folder you want to store the project.

Then type: git clone repo_url_you_copied.



```
ruxue@LAPTOP-OVUQGQ9L MINGW64 ~
$ cd Documents

ruxue@LAPTOP-OVUQGQ9L MINGW64 ~/Documents
$ cd gitlocal

ruxue@LAPTOP-OVUQGQ9L MINGW64 ~/Documents/gitlocal
$ git clone https://github.com/csc413-SFSU-Souza/csc413-tankgame-RuxueJ.git
Cloning into 'csc413-tankgame-RuxueJ'...
remote: Enumerating objects: 401, done.
remote: Counting objects: 100% (401/401), done.
remote: Compressing objects: 100% (308/308), done.
remote: Total 401 (delta 99), reused 378 (delta 82), pack-reused 0
Receiving objects: 100% (401/401), 11.89 MiB | 18.14 MiB/s, done.
Resolving deltas: 100% (99/99), done.

ruxue@LAPTOP-OVUQGQ9L MINGW64 ~/Documents/gitlocal
$ |
```

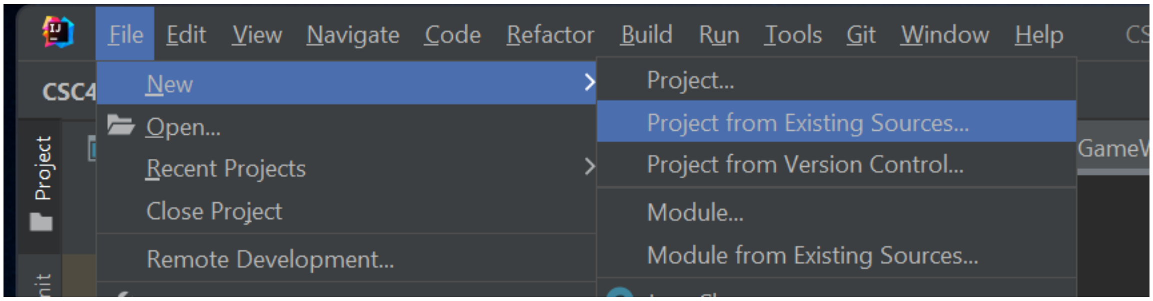
Then I should have local repo.



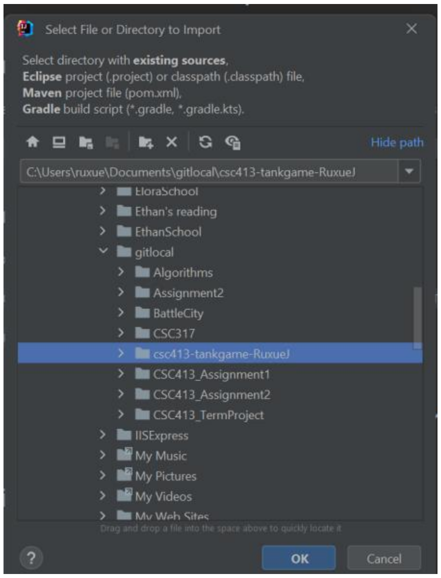
Documents › gitlocal › csc413-tankgame-RuxueJ ›

Name	Date modified	Type	Size
.git	8/10/2023 12:38 PM	File folder	
.idea	8/10/2023 12:40 PM	File folder	
jar	8/10/2023 12:38 PM	File folder	
TankGame	8/10/2023 12:38 PM	File folder	
.gitignore	8/10/2023 12:38 PM	Git Ignore Source File	1 KB
CSC413_TermProject.iml	8/10/2023 12:38 PM	IML File	1 KB
csc413-tankgame-RuxueJ.iml	8/10/2023 12:39 PM	IML File	1 KB
LICENSE	8/10/2023 12:38 PM	File	2 KB
README.md	8/10/2023 12:38 PM	MD File	2 KB

Open IntelliJ, click File -> New-> Project from Existing Sources…



Select the TankGame folder in the csc413-tankgame-RuxueJ in Test folder, click OK.



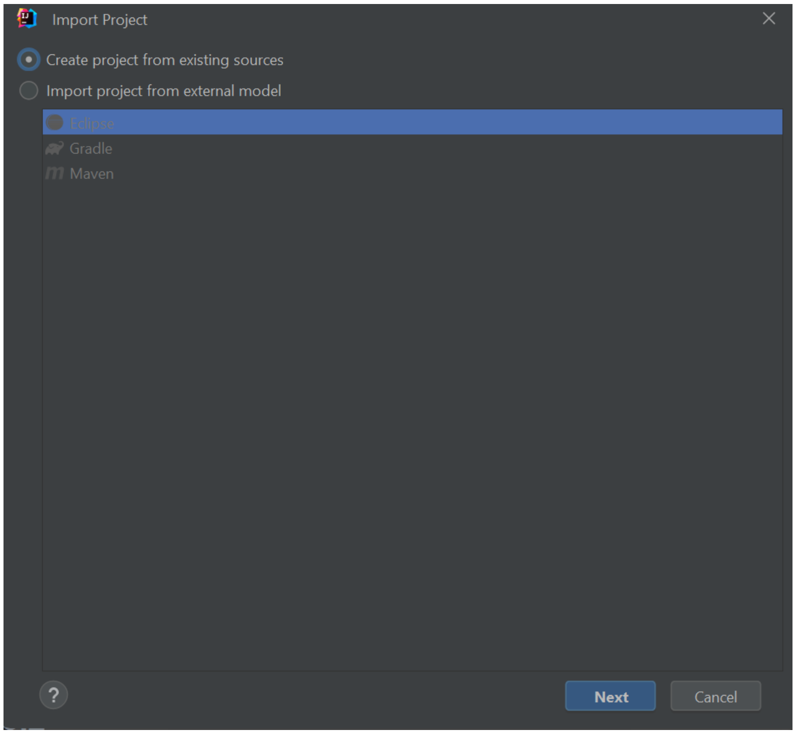
Select File or Directory to Import

Select directory with **existing sources**,
Eclipse project (.project) or classpath (.classpath) file,
Maven project file (pom.xml),
Gradle build script (*.gradle, *.gradle.kts).

C:\Users\ruxue\Documents\gitlocal\csc413-tankgame-RuxueJ

Drag and drop a file into the space above to quickly locate it

Keep the “Create project from existing resources” radio button selected, click Next. All default fields can be left alone here.

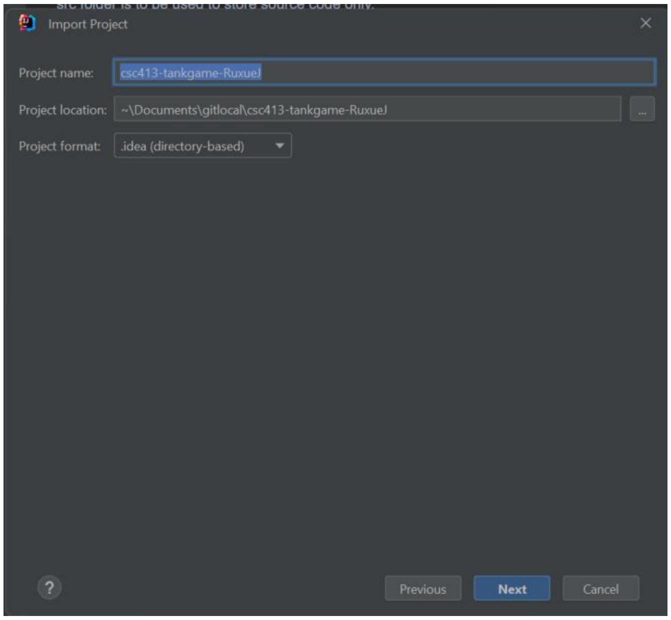


Import Project

● Create project from existing sources

○ Import project from external model

Click Next



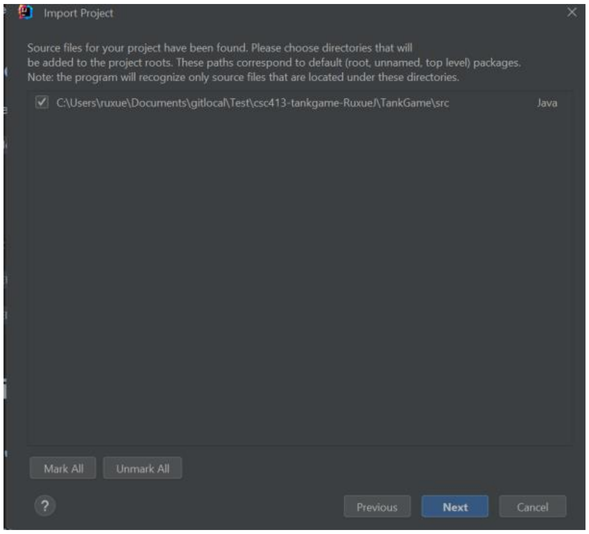
Import Project

Project name: csc413-tankgame-RuxueJ

Project location: ~\Documents\gitlocal\csc413-tankgame-RuxueJ

Project format: .idea (directory-based)

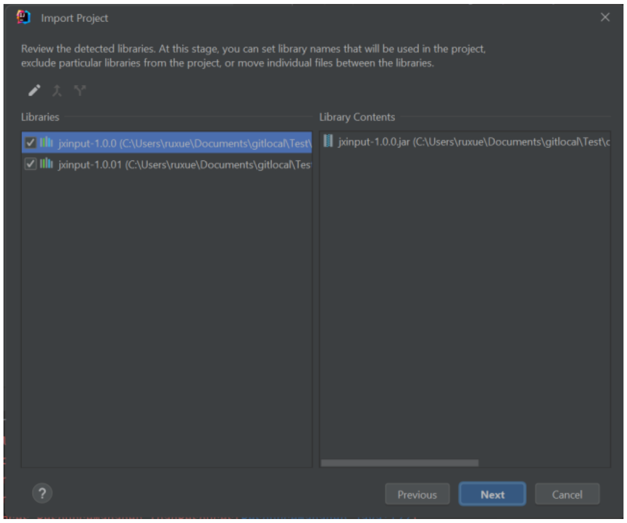
Click Next



Source files for your project have been found. Please choose directories that will be added to the project roots. These paths correspond to default (root, unnamed, top level) packages. Note: the program will recognize only source files that are located under these directories.

☑ C:\Users\ruxue\Documents\gitlocal\Test\csc413-tankgame-Ruxue\TankGame\src Java

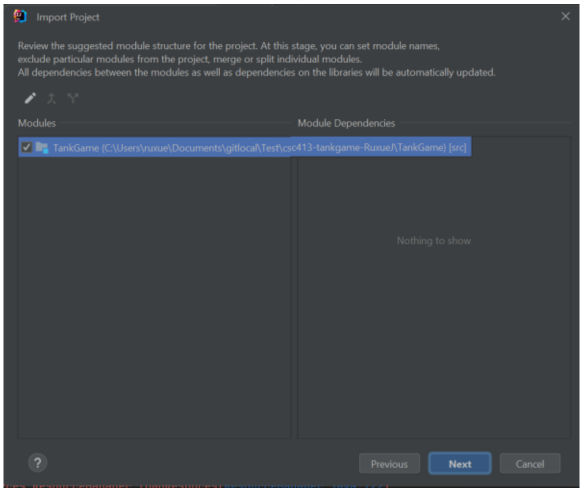
Click Next



Review the detected libraries. At this stage, you can set library names that will be used in the project, exclude particular libraries from the project, or move individual files between the libraries.

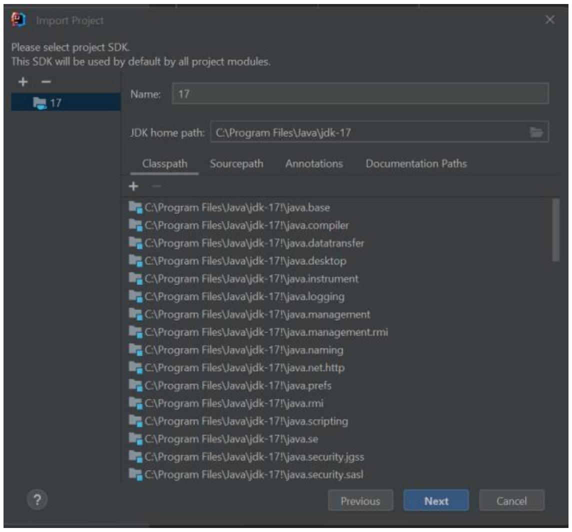
Libraries	Library Contents
☑ jxinput-1.0.0 (C:\Users\ruxue\Documents\gitlocal\Test\	jxinput-1.0.0.jar (C:\Users\ruxue\Documents\gitlocal\Test\c
☑ jxinput-1.0.01 (C:\Users\ruxue\Documents\gitlocal\Tes	

Click Next



Review the suggested module structure for the project. At this stage, you can set module names, exclude particular modules from the project, merge or split individual modules. All dependencies between the modules as well as dependencies on the libraries will be automatically updated.

Modules	Module Dependencies
☑ TankGame (C:\Users\ruxue\Documents\gitlocal\Test\csc413-tankgame-Ruxue\TankGame) [src]	Nothing to show



Please select project SDK.
This SDK will be used by default by all project modules.

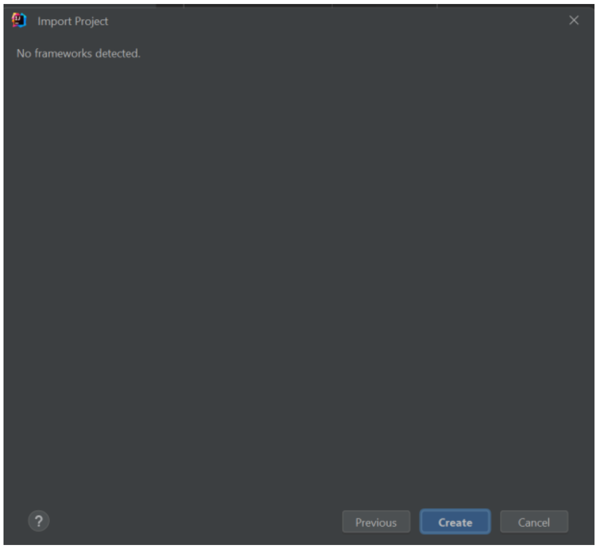
17

Name: 17

JDK home path: C:\Program Files\Java\jdk-17

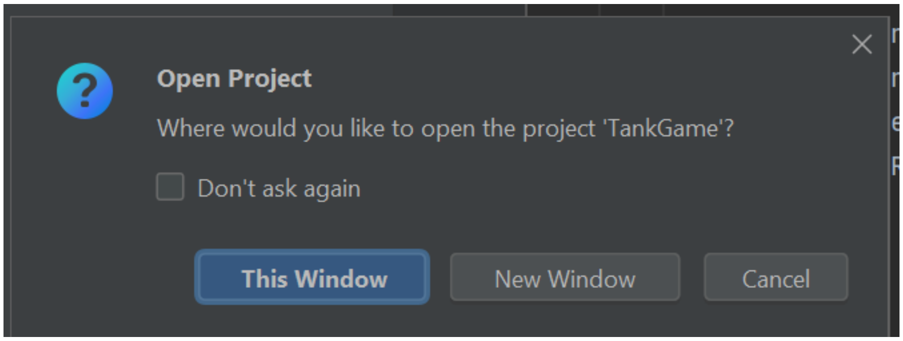
```
C:\Program Files\Java\jdk-17!\java.base
C:\Program Files\Java\jdk-17!\java.compiler
C:\Program Files\Java\jdk-17!\java.datatransfer
C:\Program Files\Java\jdk-17!\java.desktop
C:\Program Files\Java\jdk-17!\java.instrument
C:\Program Files\Java\jdk-17!\java.logging
C:\Program Files\Java\jdk-17!\java.management
C:\Program Files\Java\jdk-17!\java.management.rmi
C:\Program Files\Java\jdk-17!\java.naming
C:\Program Files\Java\jdk-17!\java.net.http
C:\Program Files\Java\jdk-17!\java.prefs
C:\Program Files\Java\jdk-17!\java.rmi
C:\Program Files\Java\jdk-17!\java.scripting
C:\Program Files\Java\jdk-17!\java.se
C:\Program Files\Java\jdk-17!\java.security.jgss
C:\Program Files\Java\jdk-17!\java.security.sasl
```

Click Next



No frameworks detected.

Click Create

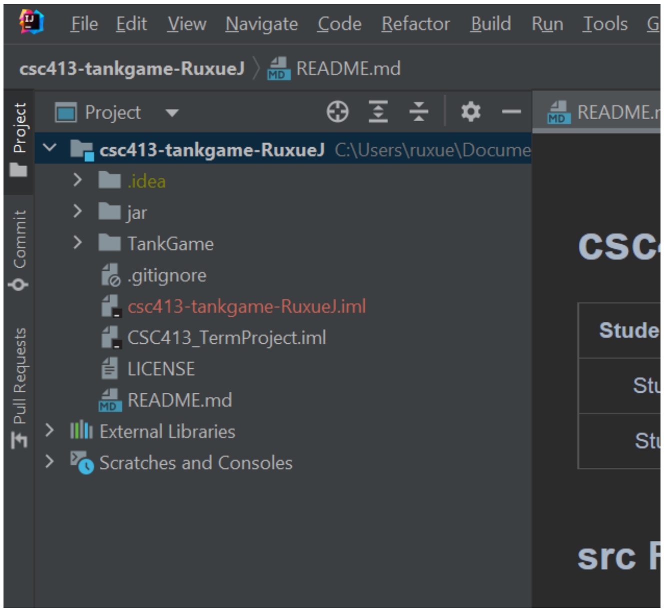


Open Project

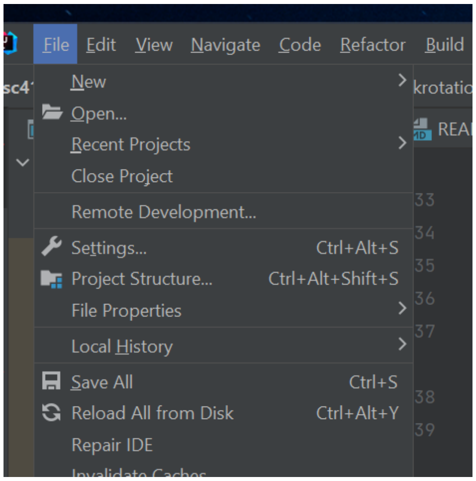
Where would you like to open the project 'TankGame'?

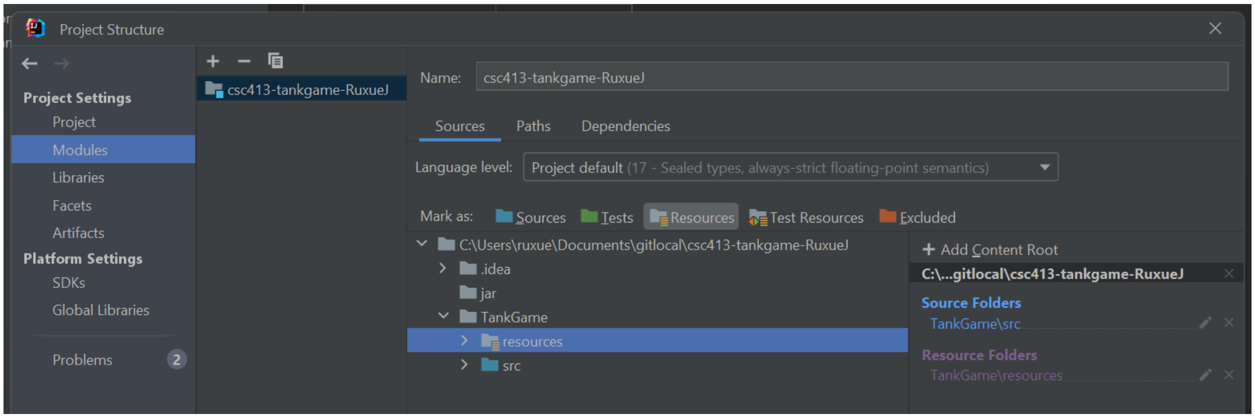
□ Don't ask again

Click New Window, and you will have this project.


File Edit View Navigate Code Refactor Build Run Tools G
csc413-tankgame-RuxueJ 〉 README.md
Project
README.r
csc413-tankgame-RuxueJ C:\Users\ruxue\Docume
.idea
jar
TankGame
.gitignore
csc413-tankgame-RuxueJ.iml
CSC413_TermProject.iml
LICENSE
README.md
External Libraries
Scratches and Consoles
Commit
Pull Requests
csc
Stude
Stu
Stu
src F

Go to File and click Project Structure, click Modules, click resources folder in TankGame, mark it as resource root, click Apply and OK.





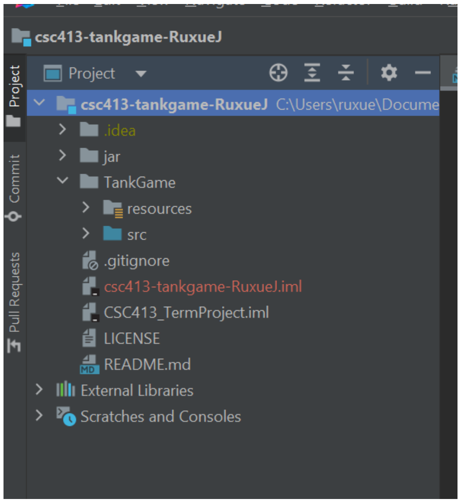
Name: csc413-tankgame-RuxueJ

Language level: Project default (17 - Sealed types, always-strict floating-point semantics)

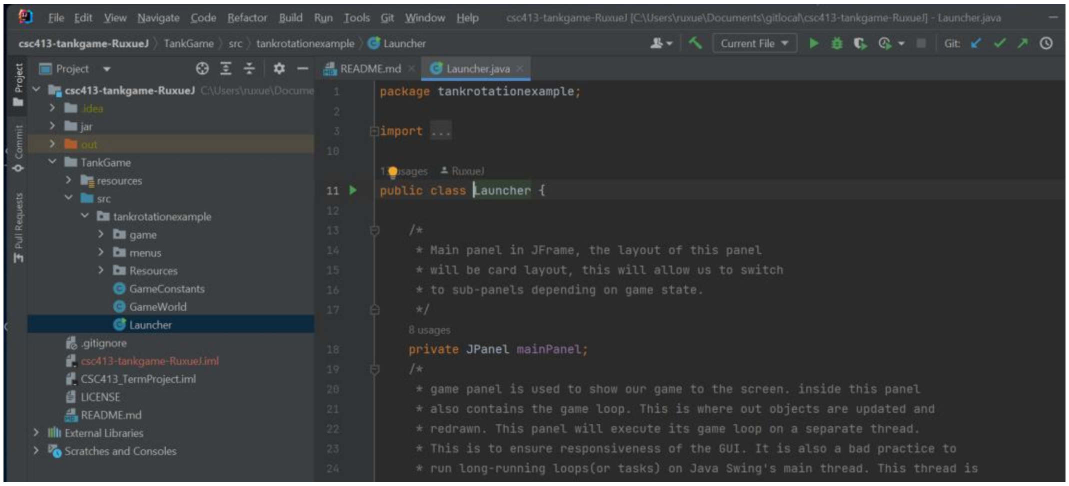
Source Folders
TankGame\src

Resource Folders
TankGame\resources

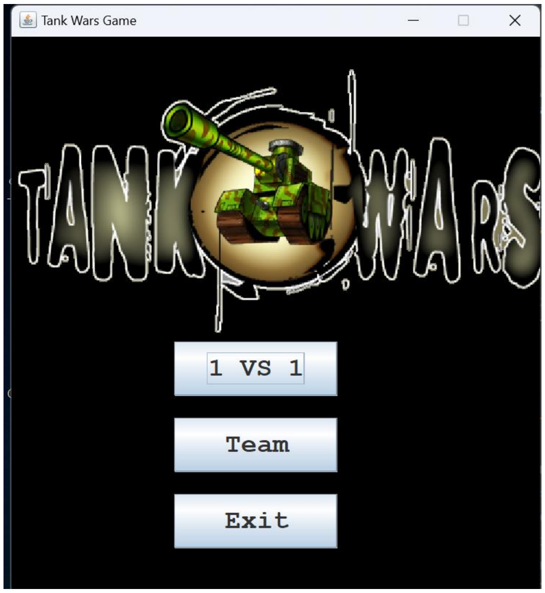
Then the project structure will look like this.



Then find Launcher class, click green triangle button, run.

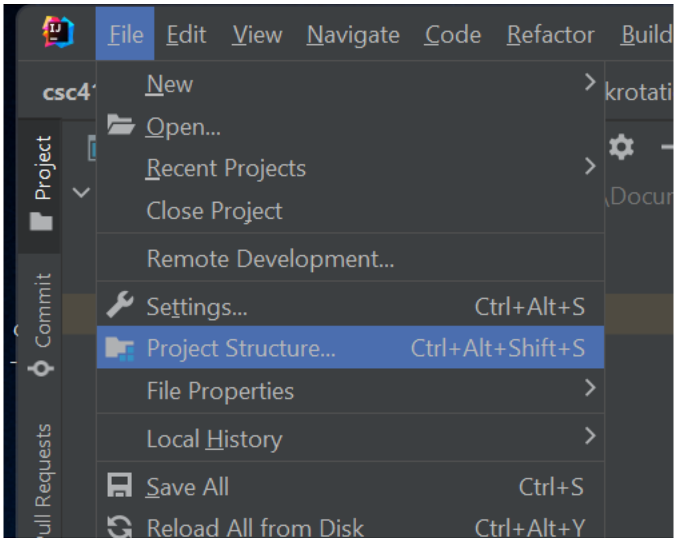

File Edit View Navigate Code Refactor Build Run Tools Git Window Help
csc413-tankgame-RuxueJ [C:\Users\ruxue\Documents\gitlocal\csc413-tankgame-RuxueJ] - Launcher.java
csc413-tankgame-RuxueJ 〉 TankGame 〉 src 〉 tankrotationexample 〉 Launcher
Current File
Git:
Project
README.md
Launcher.java
csc413-tankgame-RuxueJ C:\Users\ruxue\Docume
.idea
jar
out
TankGame
resources
src
tankrotationexample
game
menus
Resources
GameConstants
GameWorld
Launcher
.gitignore
csc413-tankgame-RuxueJ.iml
CSC413_TermProject.iml
LICENSE
README.md
External Libraries
Scratches and Consoles
1 usages
RuxueJ
8 usages

```
package tankrotationexample;

import ...

public class Launcher {

    /*
     * Main panel in JFrame, the layout of this panel
     * will be card layout, this will allow us to switch
     * to sub-panels depending on game state.
     */
    private JPanel mainPanel;
    /*
     * game panel is used to show our game to the screen. inside this panel
     * also contains the game loop. This is where out objects are updated and
     * redrawn. This panel will execute its game loop on a separate thread.
     * This is to ensure responsiveness of the GUI. It is also a bad practice to
     * run long-running loops(or tasks) on Java Swing's main thread. This thread is
```

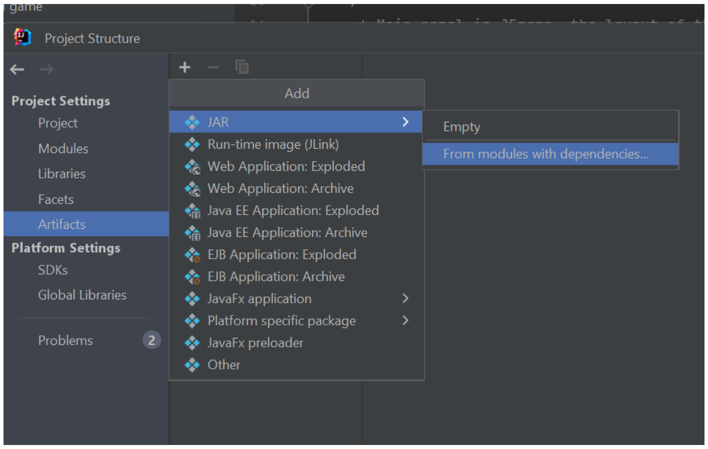


Then you can run the project.

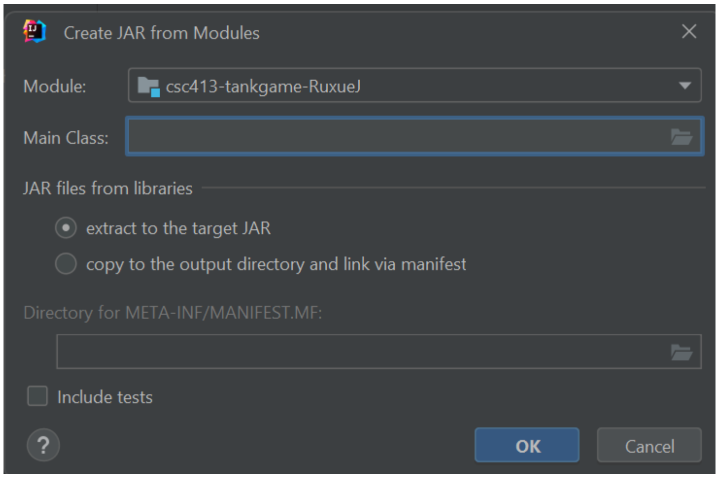
b. How to build jar
In IntelliJ, click File, click Project Structure



Click Artifacts, click +, click JAR, click From modules with dependencies.



In the Main Class, choose Launcher, click OK.


Create JAR from Modules

Module: csc413-tankgame-RuxueJ

Main Class:

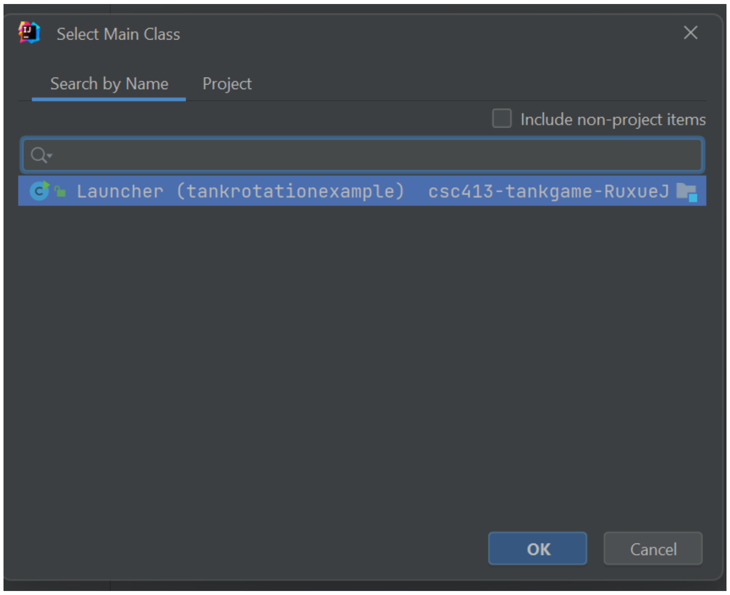
JAR files from libraries

● extract to the target JAR

○ copy to the output directory and link via manifest

Directory for META-INF/MANIFEST.MF:

□ Include tests



Select Main Class

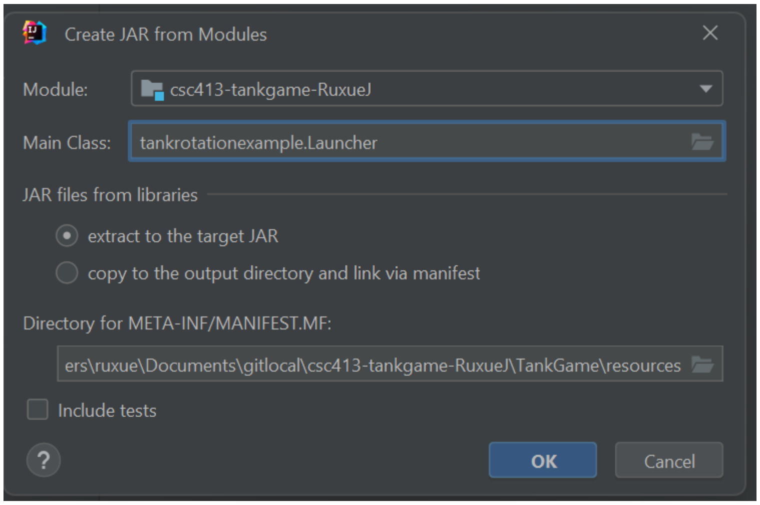
Search by Name | Project

□ Include non-project items

Launcher (tankrotationexample) csc413-tankgame-RuxueJ

OK | Cancel

Click OK



Create JAR from Modules

Module: csc413-tankgame-RuxueJ

Main Class: tankrotationexample.Launcher

JAR files from libraries

● extract to the target JAR

○ copy to the output directory and link via manifest

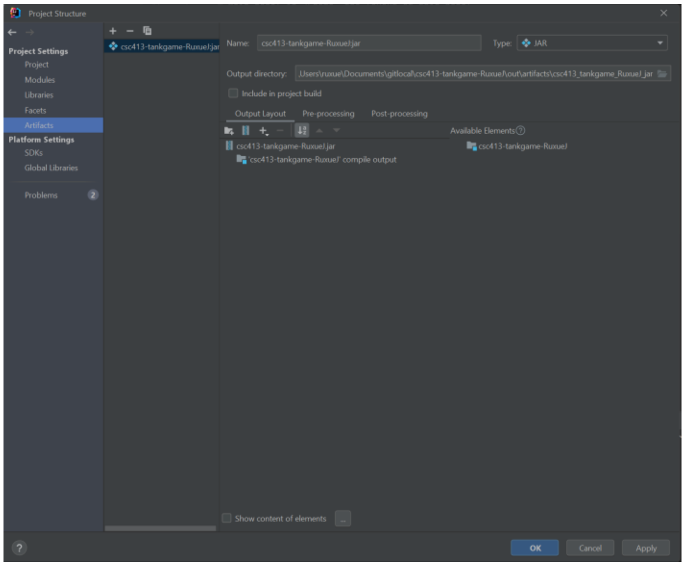
Directory for META-INF/MANIFEST.MF:

ers\ruxue\Documents\gitlocal\csc413-tankgame-RuxueJ\TankGame\resources

□ Include tests

OK | Cancel

Click Apply and click OK.



Project Structure

Project Settings: Project, Modules, Libraries, Facets, Artifacts

Platform Settings: SDKs, Global Libraries

Problems 2

csc413-tankgame-RuxueJ:jar

Name: csc413-tankgame-RuxueJ:jar

Type: JAR

Output directory: \Users\ruxue\Documents\gitlocal\csc413-tankgame-RuxueJ\out\artifacts\csc413_tankgame_RuxueJ_jar

□ Include in project build

Output Layout | Pre-processing | Post-processing

csc413-tankgame-RuxueJ.jar

'csc413-tankgame-RuxueJ' compile output

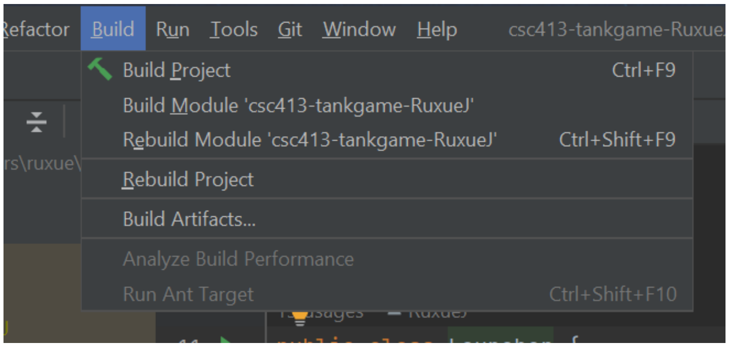
Available Elements

csc413-tankgame-RuxueJ

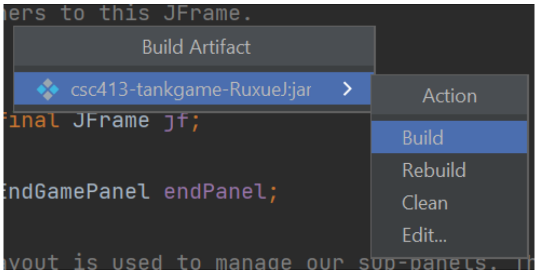
□ Show content of elements

OK | Cancel | Apply

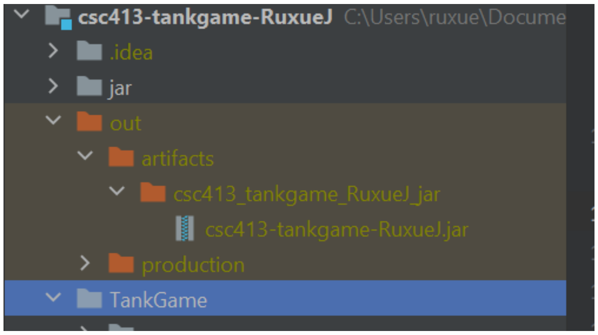
Click Build and click Build Artifacts.


Refactor Build Run Tools Git Window Help csc413-tankgame-Ruxue
Build Project Ctrl+F9
Build Module 'csc413-tankgame-RuxueJ'
Rebuild Module 'csc413-tankgame-RuxueJ' Ctrl+Shift+F9
Rebuild Project
Build Artifacts...
Analyze Build Performance
Run Ant Target Ctrl+Shift+F10

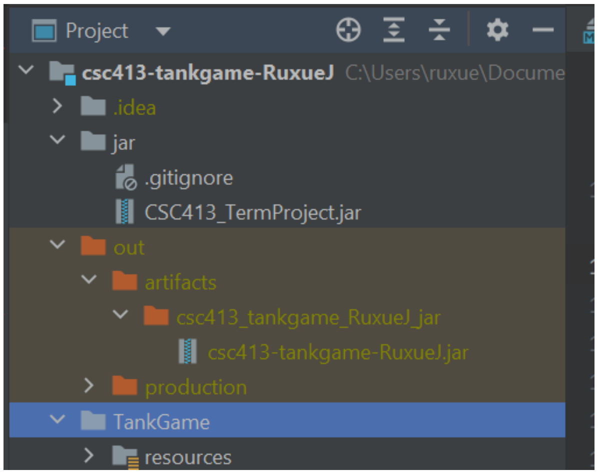
Click Build.



Move the csc413-tankgame-RuxueH.jar to jar folder.

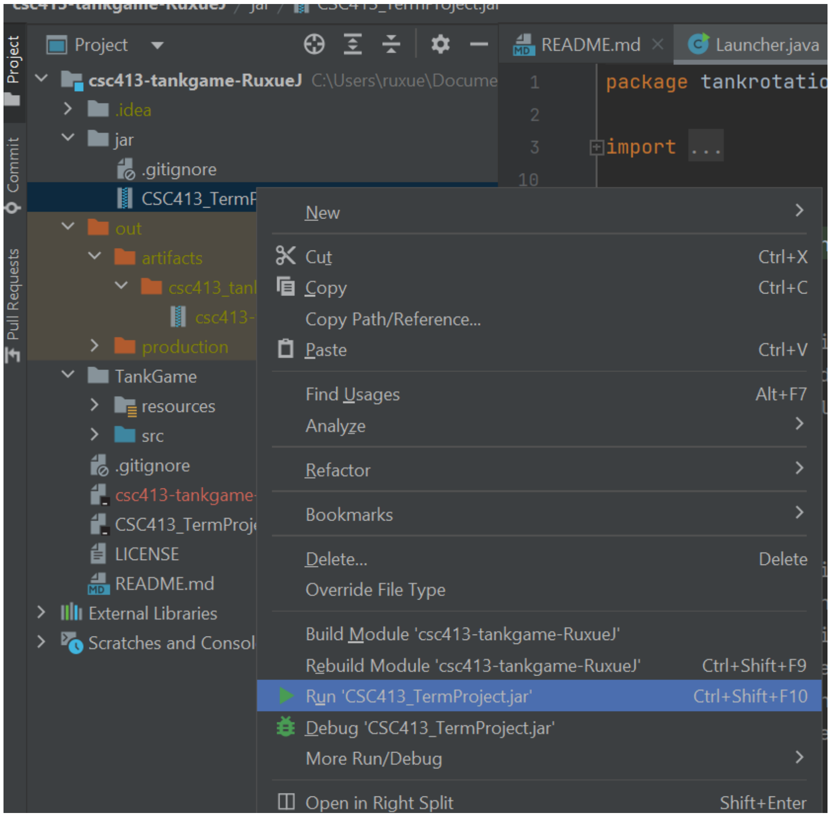

csc413-tankgame-RuxueJ C:\Users\ruxue\Docume
.idea
jar
out
artifacts
csc413_tankgame_RuxueJ_jar
csc413-tankgame-RuxueJ.jar
production
TankGame

Then the jar for this tank game program has been built successfully.


Project
csc413-tankgame-RuxueJ C:\Users\ruxue\Docume
.idea
jar
.gitignore
CSC413_TermProject.jar
out
artifacts
csc413_tankgame_RuxueJ_jar
csc413-tankgame-RuxueJ.jar
production
TankGame
resources

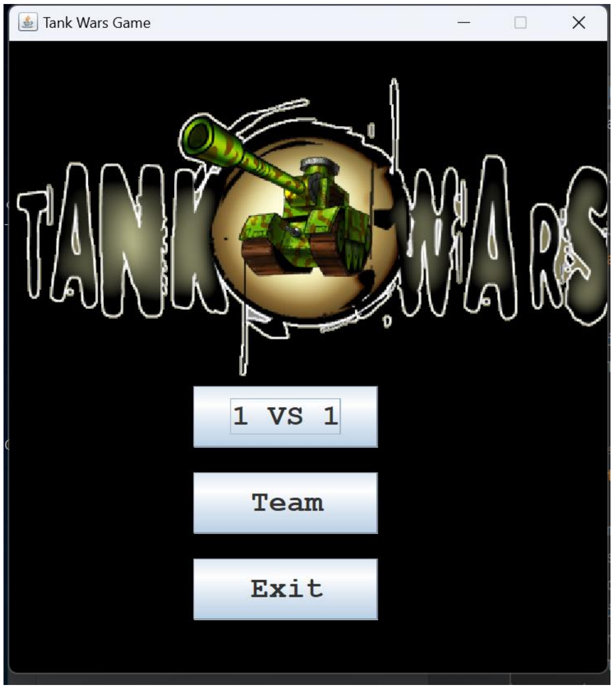
c. Command to run the built jar.

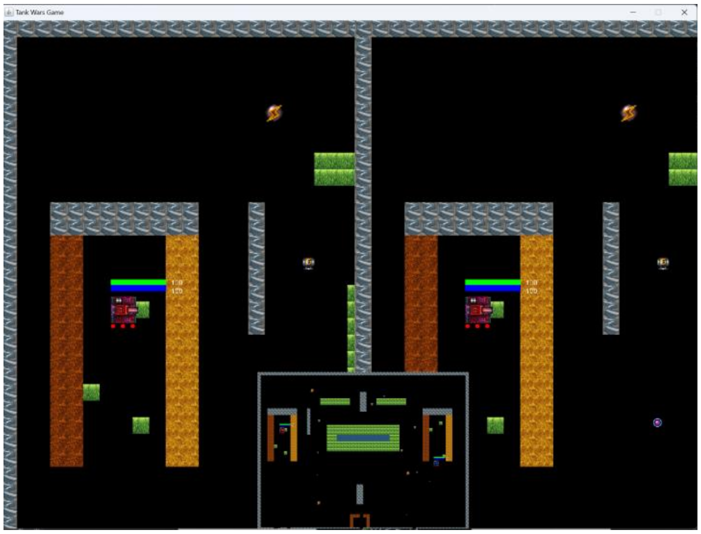
Right click the CSC413_TermProject.jar and click Run.



```
package tankrotatio
import ...
```

Then you can run the jar.

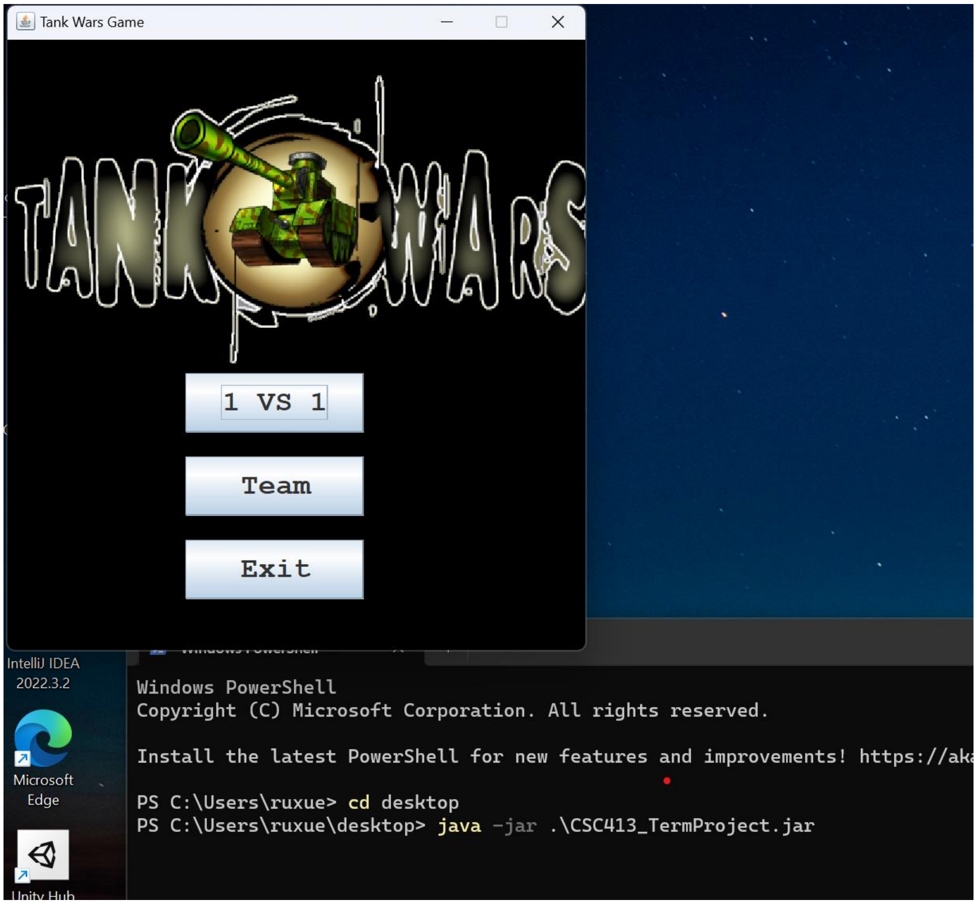




You can also move the jar to desktop, and double click the jar, and open it.

Also, in PowerShell, cd to the directory where you jar is.

Java -jar .\CSC413_TermProject.jar and open the jar.

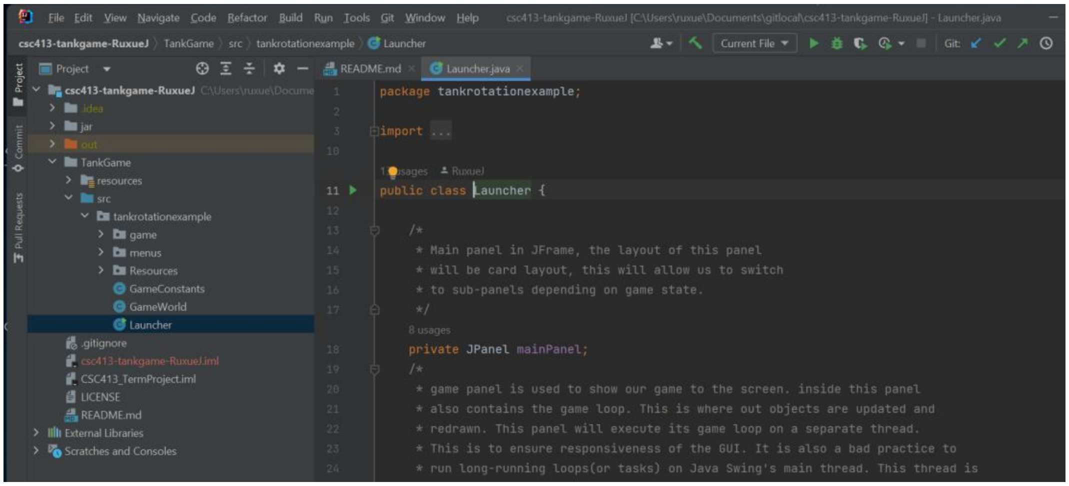


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka

PS C:\Users\ruxue> cd desktop
PS C:\Users\ruxue\desktop> java -jar .\CSC413_TermProject.jar
```

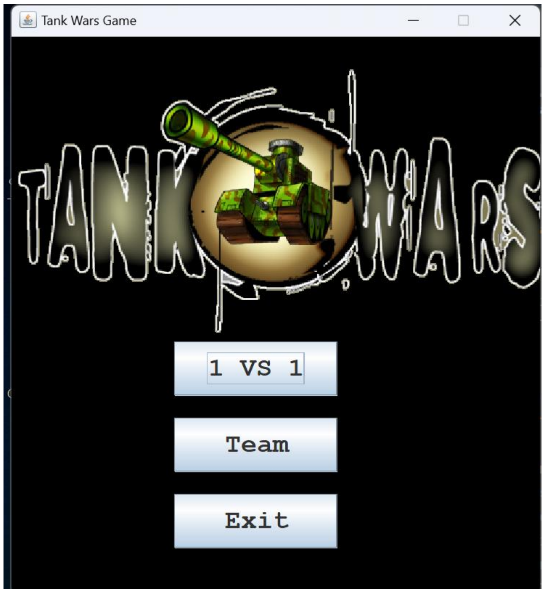
4 How to Run your Project

Make sure you set the resource path in the step 3.

Then find Launcher class, click green triangle button, run.


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
csc413-tankgame-RuxueJ [C:\Users\ruxue\Documents\gitlocal\csc413-tankgame-RuxueJ] - Launcher.java
csc413-tankgame-RuxueJ 〉 TankGame 〉 src 〉 tankrotationexample 〉 Launcher
Current File
Git:
Project
Commit
Pull Requests
csc413-tankgame-RuxueJ C:\Users\ruxue\Docume
.idea
jar
out
TankGame
resources
src
tankrotationexample
game
menus
Resources
GameConstants
GameWorld
Launcher
.gitignore
csc413-tankgame-RuxueJ.iml
CSC413_TermProject.iml
LICENSE
README.md
External Libraries
Scratches and Consoles
README.md
Launcher.java

```
package tankrotationexample;

import ...

1 usages  RuxueJ
public class Launcher {

    /*
     * Main panel in JFrame, the layout of this panel
     * will be card layout, this will allow us to switch
     * to sub-panels depending on game state.
     */
    8 usages
    private JPanel mainPanel;
    /*
     * game panel is used to show our game to the screen. inside this panel
     * also contains the game loop. This is where out objects are updated and
     * redrawn. This panel will execute its game loop on a separate thread.
     * This is to ensure responsiveness of the GUI. It is also a bad practice to
     * run long-running loops(or tasks) on Java Swing's main thread. This thread is
```



Or double click the jar file.

Or use command line java -jar .\CSC413_TermProject.jar to open the jar.

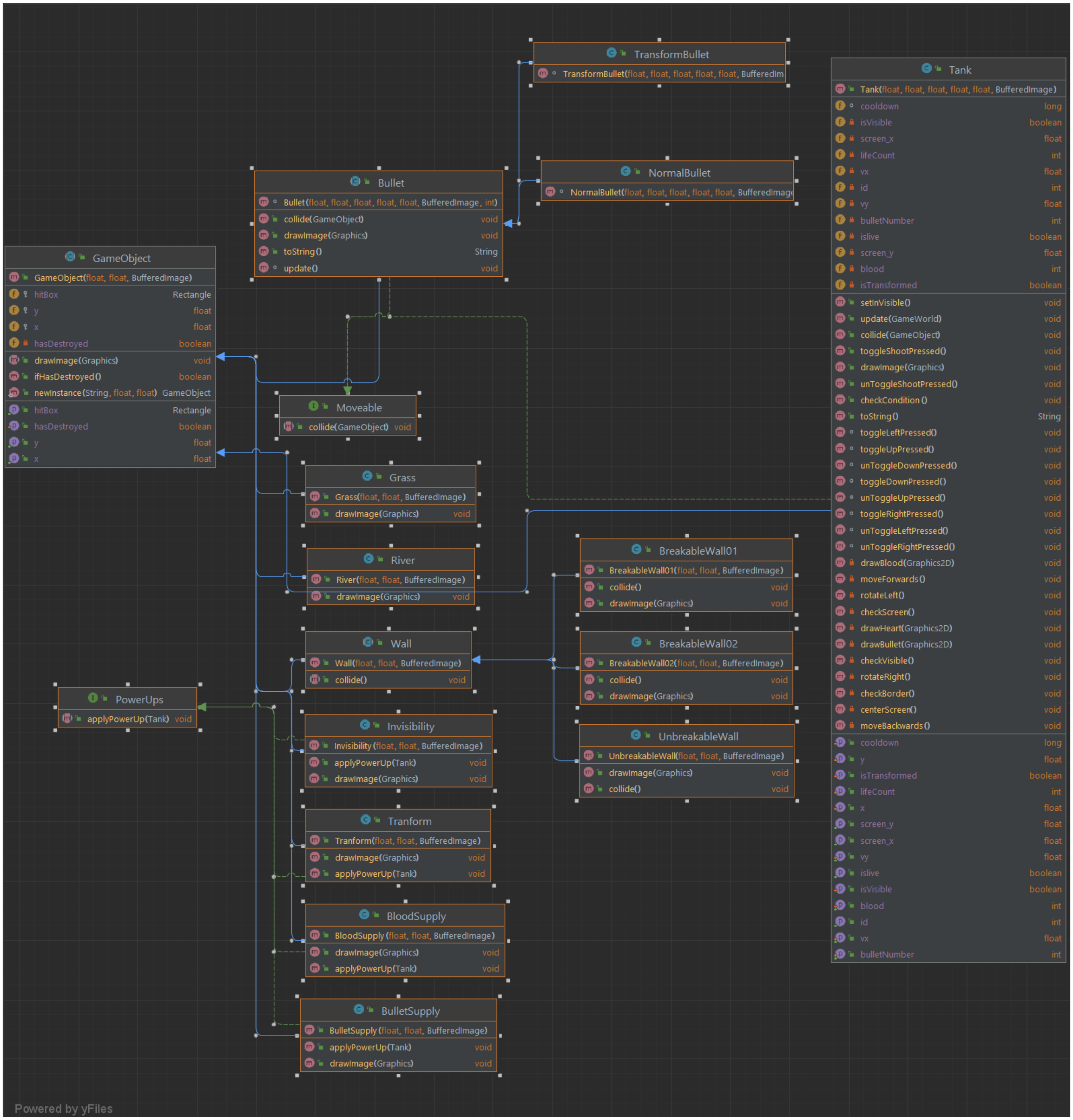
For the first player, W for forward, A for left rotate, S for backward, D for right rotation, WhiteSpace for shoot.

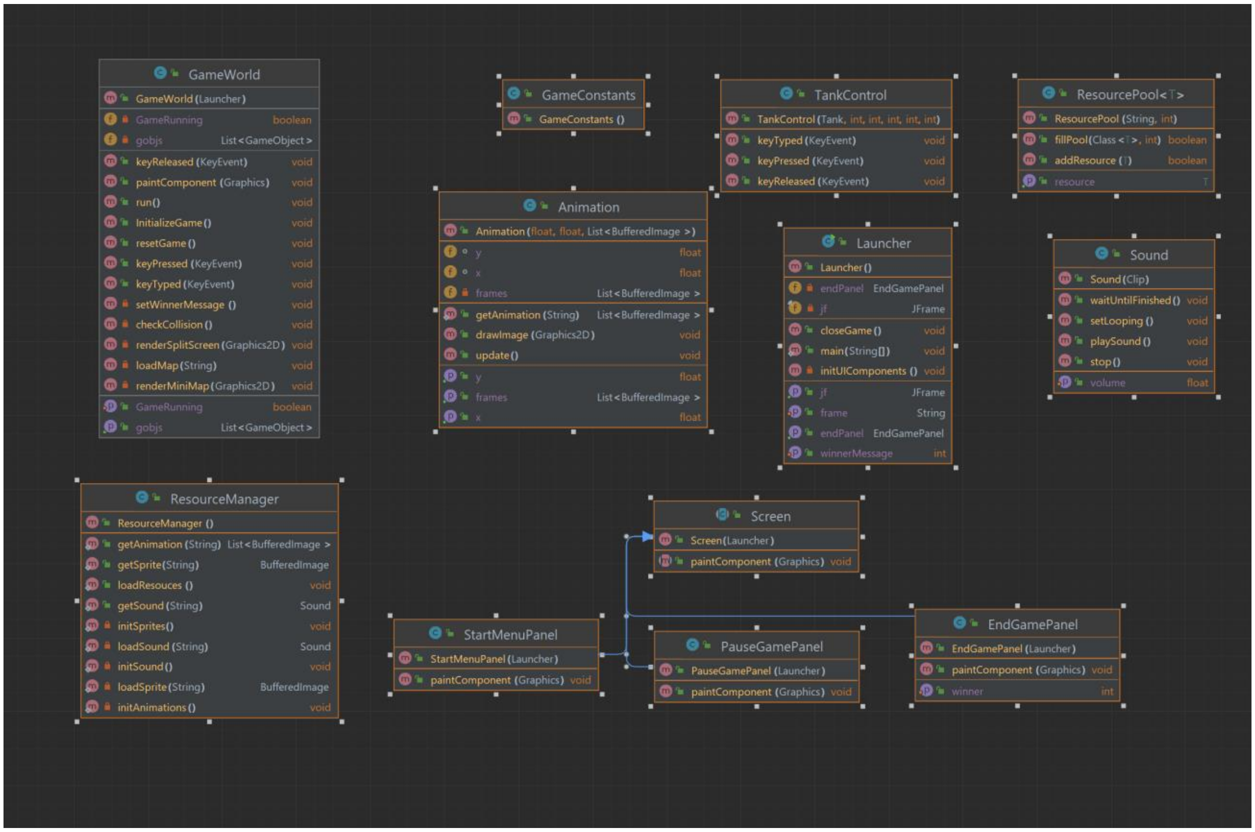
For the second player, UP is for forward, Left is for left rotate, Down is for backward, Right is for right rotation, Ctrl is for shoot.

5 Assumption Made

There are 2 players. One vs one.

6 Tank Game Class Diagram Tank





7 Class Descriptions of classes implemented in the Tank Game

Menus package includes an abstract class Screen. StartMenuPanel, EndMenuPanel, and PauseGamePanel extend Screen.

Game package includes classes of game objects, such as Tank, Bullet, 4 powerups, and Walls, Grass, and River. It contains Animations and Sound class as well.

Resources package includes ResourceManager that loads all images, sound, animation resources.

The Launcher class launches the game. It sets main panel to its jframe and set a cardLayout to switch subpanel such as startPanel, gamePanel, and endPanel.

gamePanel is an object of GameWorld class. It has a list of Game Objects. And it runs in another thread to keep the game running smoothly. Wall, River, Grass, Tank, Bullet, BulletSupply, BloodSupply, Invisible, Transform extend GameObject class. Four powerups (BulletSupply, BloodSupply, Invisible, and Transform) implements Powerup interface, and implement applyPowerUp method.

GameConstants contains all constains in the Game.

8 Project Reflection

I have played City Battle when I was a kid, so I was ambitious when I started the game. I want to display almost the same program. But then I am stuck with enemy tanks, so I switched to finish the 1 VS 1 mode.

I have gained deeper understanding about the association in OOP. I spend mush time figuring about what will happen when tank shoot bullet. If a bullet is created, I need to add it to the list of game object to update and draw it. So, I need to pass Gameworld object to tank class. I can pass it through constructor or through the parameter in update function in Tank class. I think I still need to review the course content about the relationships between classes.

I also gained experience about object collides, switch panels in card layout, and how to load resources such as images, animations, and sounds.

At last, I learned there is no best design, but a better design. Programmers need to understand codes deeply, and compare pros and cons, and make decisions. I am often struggling in making decisions and choices, but this experience teaches me completion is better than perfect.

9 Project Conclusion/Results

The project meets the requirements for this course, but I still want to implement more functions to it.

First, I want to add a pause panel. Whenever user press F9, the gameworld thread pause, and display a pause panel.

Second, I want to add map selection function in the menu.

Third, I want to add teamwork mode in the start game panel. And that requires another Game world class.

Forth, I want to hide tank underneath the grass.

Fifth, I want to hide powerups underneath the walls.

In a nutshell, this is a great experience for me. I will continue work on this program.