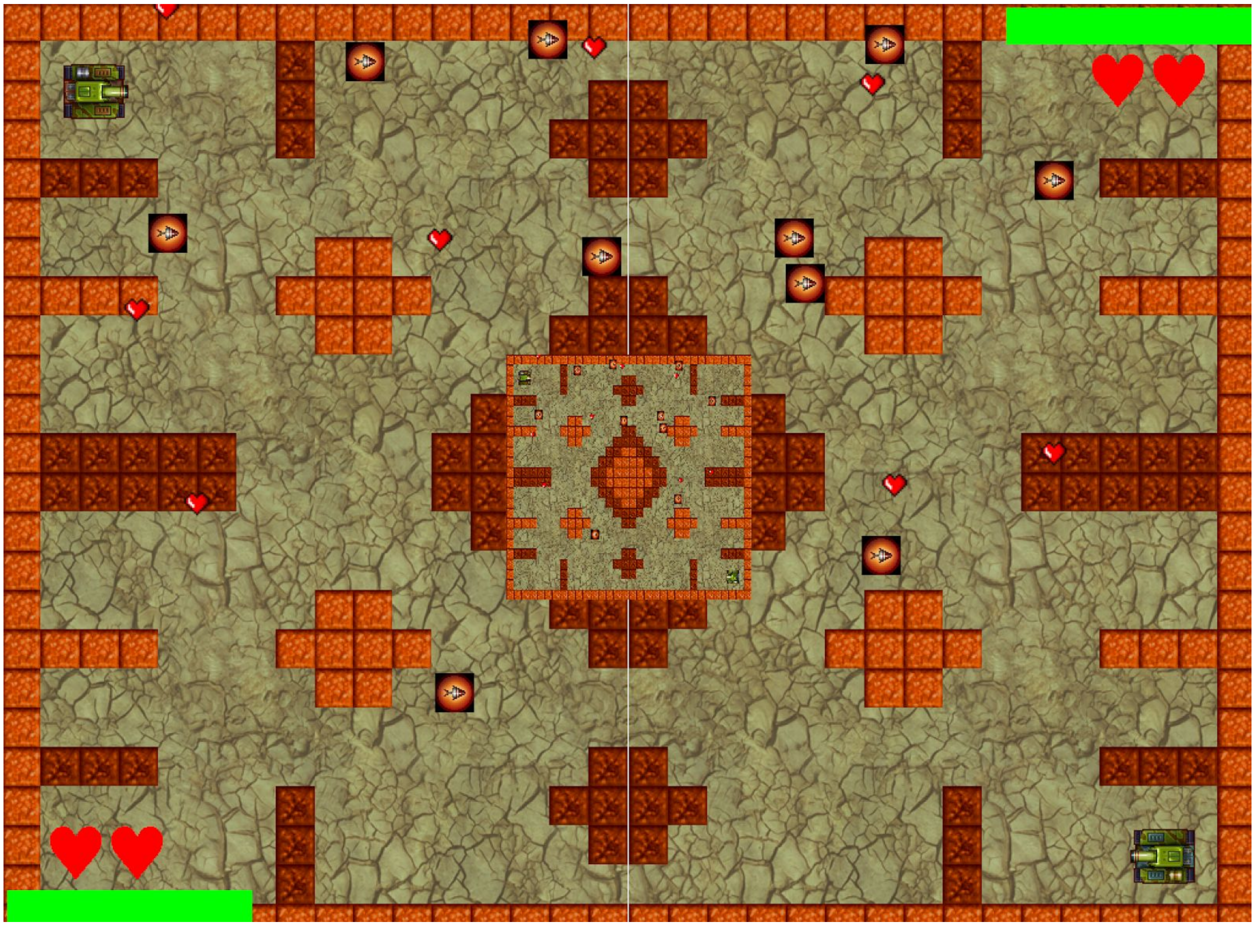
Term Project: Tank Game

CSC 413 Summer 2019
Student name: Tsun Ming Lee
ID: 918541473

Github Link:
https://github.com/csc413-02-summer2019/csc413-tankgame-hkmatthew711

1. Introduction

1.1 Project Overview



1.2 Introduction of the tank game

The purpose of this program is implementing java tank game with 2 players. In this program, there have one package and 10 java classes. The tank game should be satisfied with the following requirements: Tank Game must have 2 Players who can move forwards, backwards, and rotate in all directions. The background should have split screen and mini- map so that we can track tanks. At the same time, each tank will carry health bars and live counts. There are two kinds of walls in the tank game which are unbreakable walls and breakable walls. The most important thing is that tanks should need to item to health and attack.

2. Development Environment

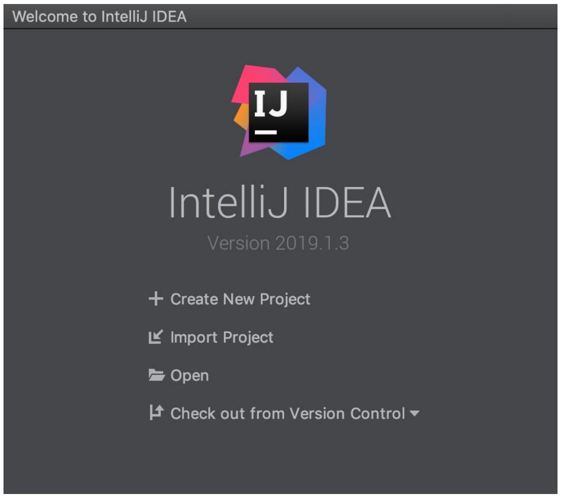
- For this project, I use IntelliJ IDEA CE to compile it, java version "2019.1.3".

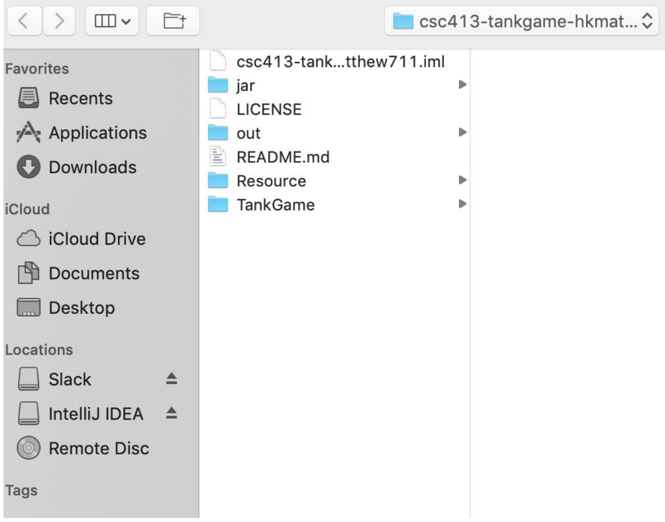
- The special resources: map.txt

3. How to Build/Import your Project

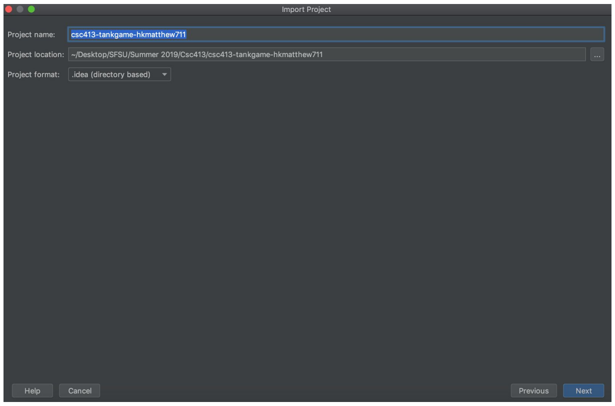
3.1 Import file

The environment for the project is IntelliJ. First, open IntelliJ. Then click the Import Project on the welcome screen. Find the project location and click open.





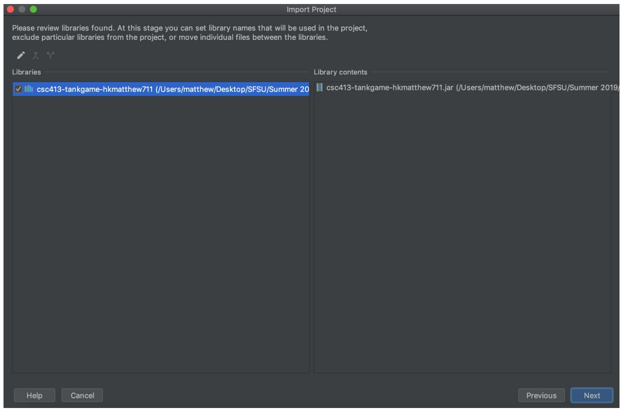
Then following the screen and keep click on Next, until Finish.



Project name: csc413-tankgame-hkmatthew711

Project location: ~/Desktop/SFSU/Summer 2019/Csc413/csc413-tankgame-hkmatthew711

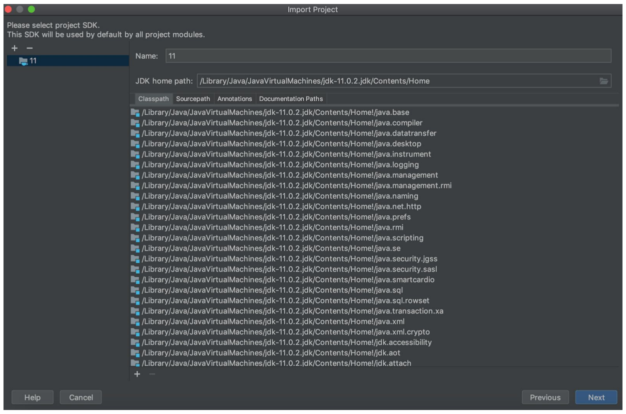
Project format: .idea (directory based)



Please review libraries found. 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.

☑ csc413-tankgame-hkmatthew711 (/Users/matthew/Desktop/SFSU/Summer 20

csc413-tankgame-hkmatthew711.jar (/Users/matthew/Desktop/SFSU/Summer 2019/



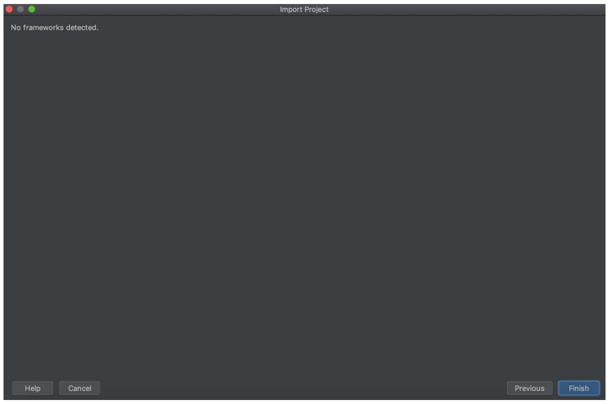
Import Project

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

Name: 11

JDK home path: /Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home

```
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.base
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.compiler
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.datatransfer
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.desktop
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.instrument
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.logging
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.management
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.management.rmi
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.naming
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.net.http
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.prefs
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.rmi
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.scripting
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.se
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.security.jgss
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.security.sasl
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.smartcardio
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.sql
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.sql.rowset
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.transaction.xa
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.xml
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/java.xml.crypto
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/jdk.accessibility
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/jdk.aot
/Library/Java/JavaVirtualMachines/jdk-11.0.2.jdk/Contents/Home!/jdk.attach
```

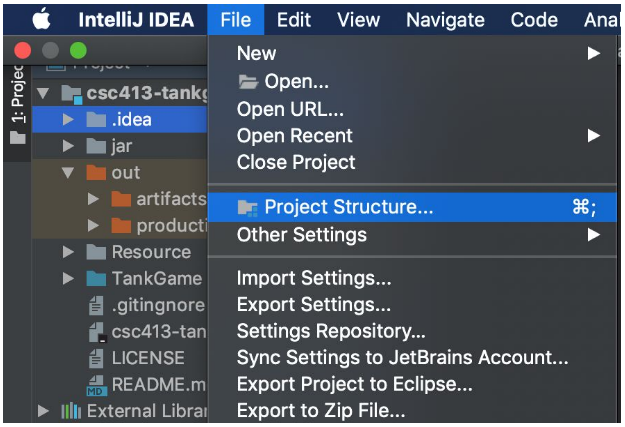


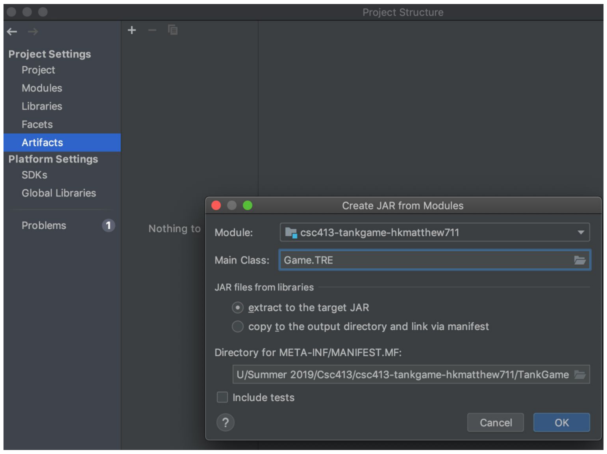
Import Project

No frameworks detected.

3.2 Build the jar file

Click File → Project Structure → Artifacts → Main Class select Game.TRE → OK → OK





Project Structure

Create JAR from Modules

Module: csc413-tankgame-hkmatthew711

Main Class: Game.TRE

JAR files from libraries

● extract to the target JAR

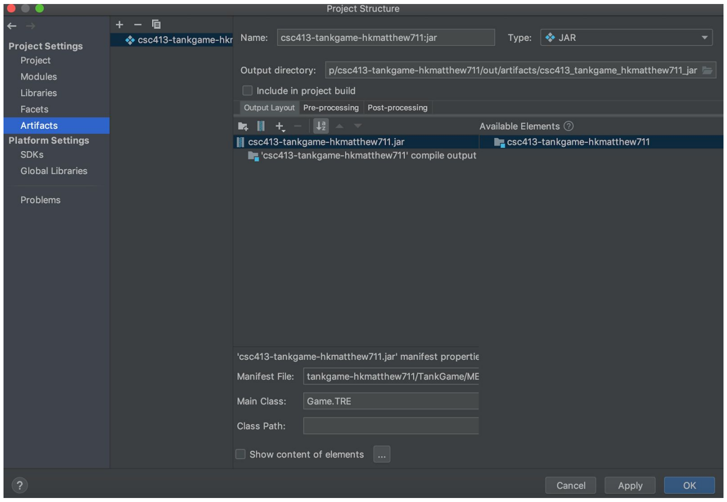
○ copy to the output directory and link via manifest

Directory for META-INF/MANIFEST.MF:

U/Summer 2019/Csc413/csc413-tankgame-hkmatthew711/TankGame

□ Include tests

Nothing to



Project Structure

Name: csc413-tankgame-hkmatthew711:jar

Type: JAR

Output directory: p/csc413-tankgame-hkmatthew711/out/artifacts/csc413_tankgame_hkmatthew711_jar

□ Include in project build

csc413-tankgame-hkmatthew711.jar
'csc413-tankgame-hkmatthew711' compile output

Available Elements
csc413-tankgame-hkmatthew711

'csc413-tankgame-hkmatthew711.jar' manifest propertie

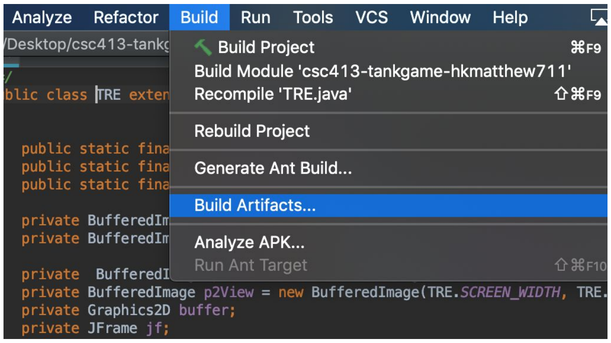
Manifest File: tankgame-hkmatthew711/TankGame/ME

Main Class: Game.TRE

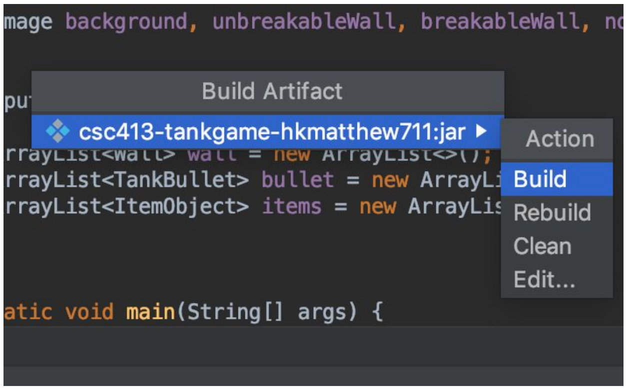
Class Path:

□ Show content of elements

Click Build → Build Artifacts → Build

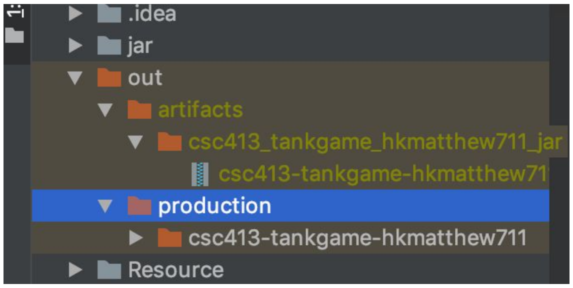


```
blic class TRE exten

    public static fina
    public static fina
    public static fina

    private BufferedIm
    private BufferedIm

    private  BufferedI
    private BufferedImage p2View = new BufferedImage(TRE.SCREEN_WIDTH, TRE.
    private Graphics2D buffer;
    private JFrame jf;
```

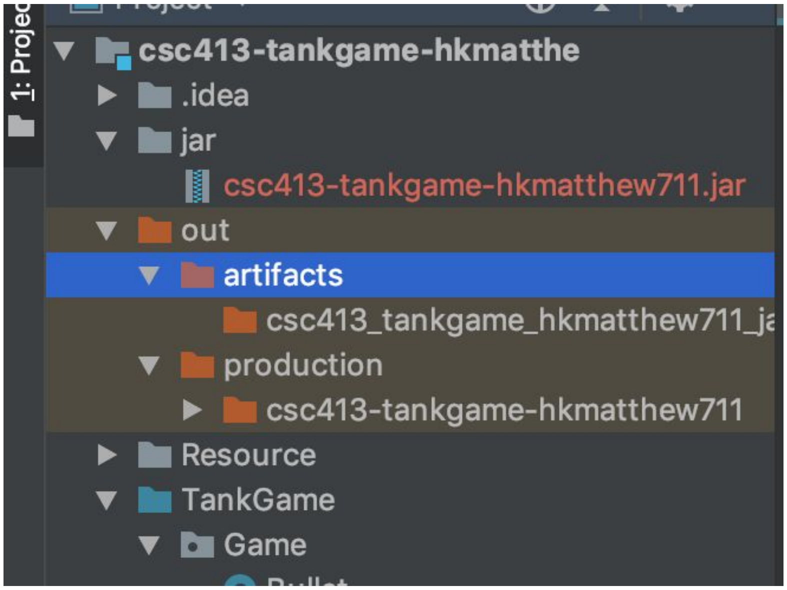


```
mage background, unbreakableWall, breakableWall, no
pu
rrayList<Wall> wall = new ArrayList<>();
rrayList<TankBullet> bullet = new ArrayLi
rrayList<ItemObject> items = new ArrayLis

atic void main(String[] args) {
```

After build the file, we can see the “csc413-tankgame-hkmatthew711.jar” in artifacts file

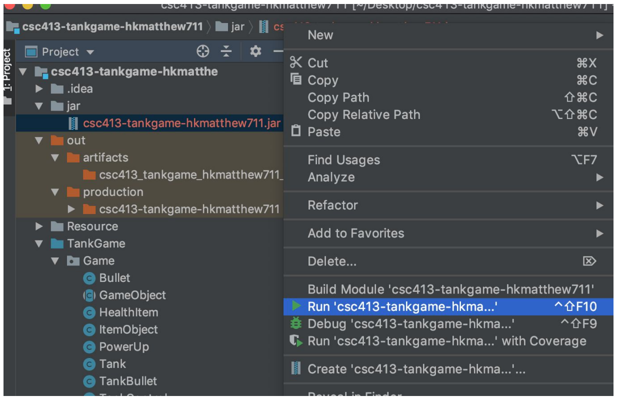

1:
.idea
jar
out
artifacts
csc413_tankgame_hkmatthew711_jar
csc413-tankgame-hkmatthew71
production
csc413-tankgame-hkmatthew711
Resource

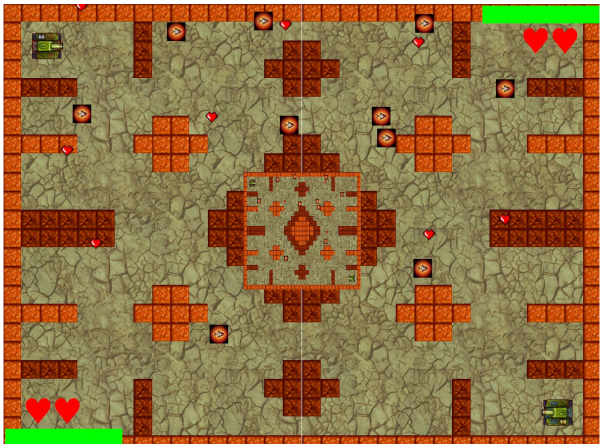
Then move the “csc413-tankgame-hkmatthew711.jar” to jar file.


1: Projec
csc413-tankgame-hkmatthe
.idea
jar
csc413-tankgame-hkmatthew711.jar
out
artifacts
csc413_tankgame_hkmatthew711_ja
production
csc413-tankgame-hkmatthew711
Resource
TankGame
Game

3.3 Run the built jar

Right click the “csc413-tankgame-hkmatthew711.jar” and click the Run “csc413-tankgame-hkmatthew711.jar”, then the game will start.

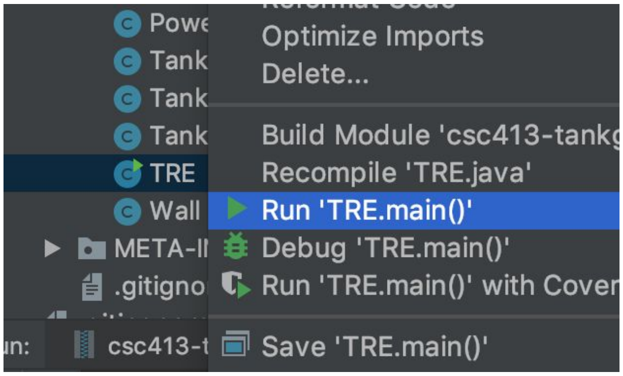

csc413-tankgame-hkmatthew711
jar
Project
csc413-tankgame-hkmatthe
.idea
jar
csc413-tankgame-hkmatthew711.jar
out
artifacts
csc413_tankgame_hkmatthew711_
production
csc413-tankgame-hkmatthew711
Resource
TankGame
Game
Bullet
GameObject
HealthItem
ItemObject
PowerUp
Tank
TankBullet
New
Cut ⌘X
Copy ⌘C
Copy Path ⇧⌘C
Copy Relative Path ⌥⇧⌘C
Paste ⌘V
Find Usages ⌥F7
Analyze
Refactor
Add to Favorites
Delete...
Build Module 'csc413-tankgame-hkmatthew711'
Run 'csc413-tankgame-hkma...' ^⇧F10
Debug 'csc413-tankgame-hkma...' ^⇧F9
Run 'csc413-tankgame-hkma...' with Coverage
Create 'csc413-tankgame-hkma...'...

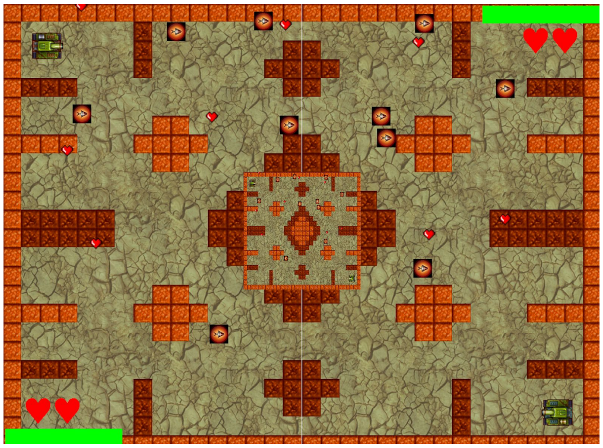


4. How to Run your Project

4.1 run the game

Right click the TRE → Run ‘TRE.main()’ → the game will show up again



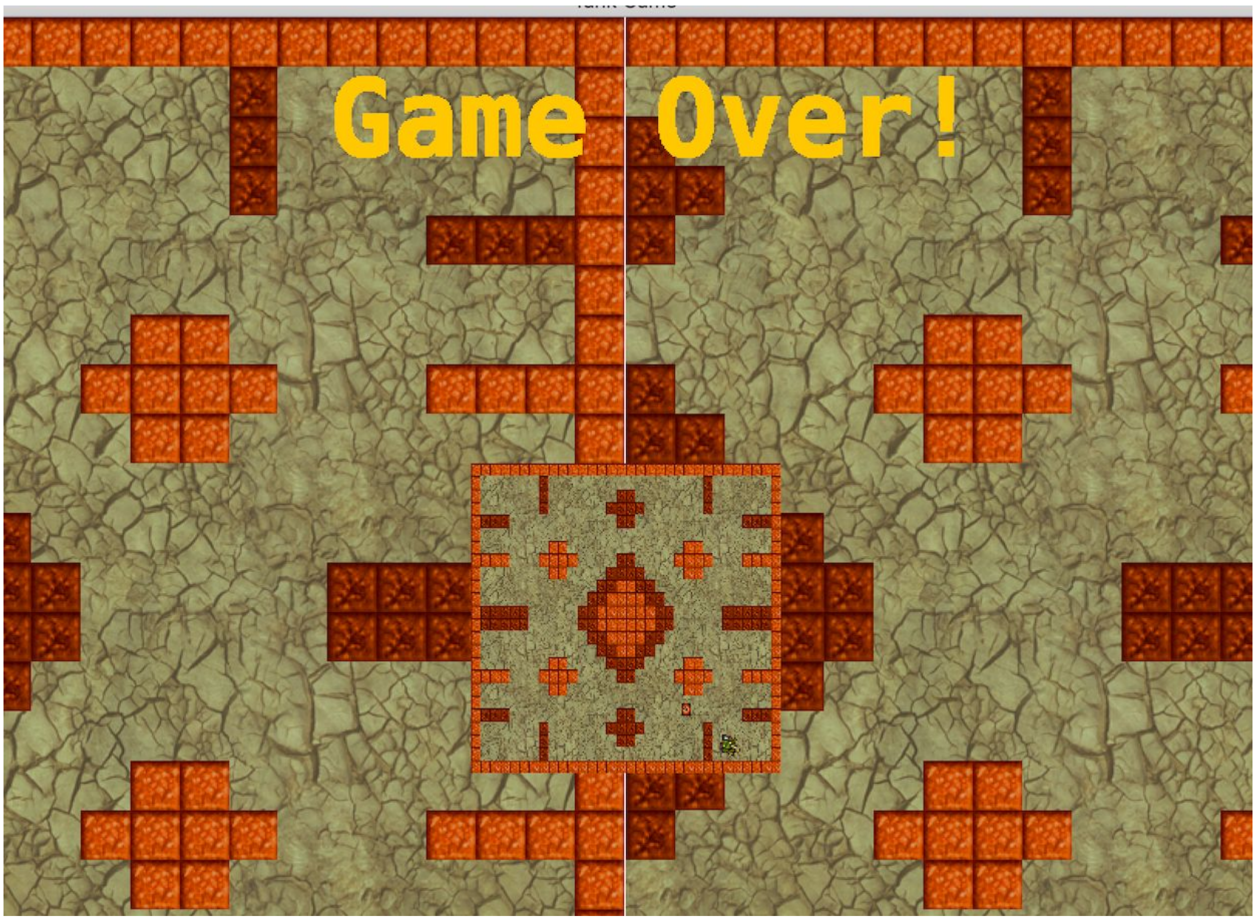


4.2 how to control the tank and the rules of the game

- Each tank has 3 lives and 200 health points.
- Each bullet hit cost 50 health points.
- If a tank's health point is below zero, the tank will respawn at its original position.
- If a breakable wall is destroyed, it will respawn again after 10 seconds.
- In every 7 seconds, there would be a special item and a health item generated at a random spot on the map.
- If a tank picks up a special item, the opponent tank would instantly lose 50 health points.

- If a tank picks up a health item, it would heal itself with 50 health points.
- The program will exit automatically after the game ended.

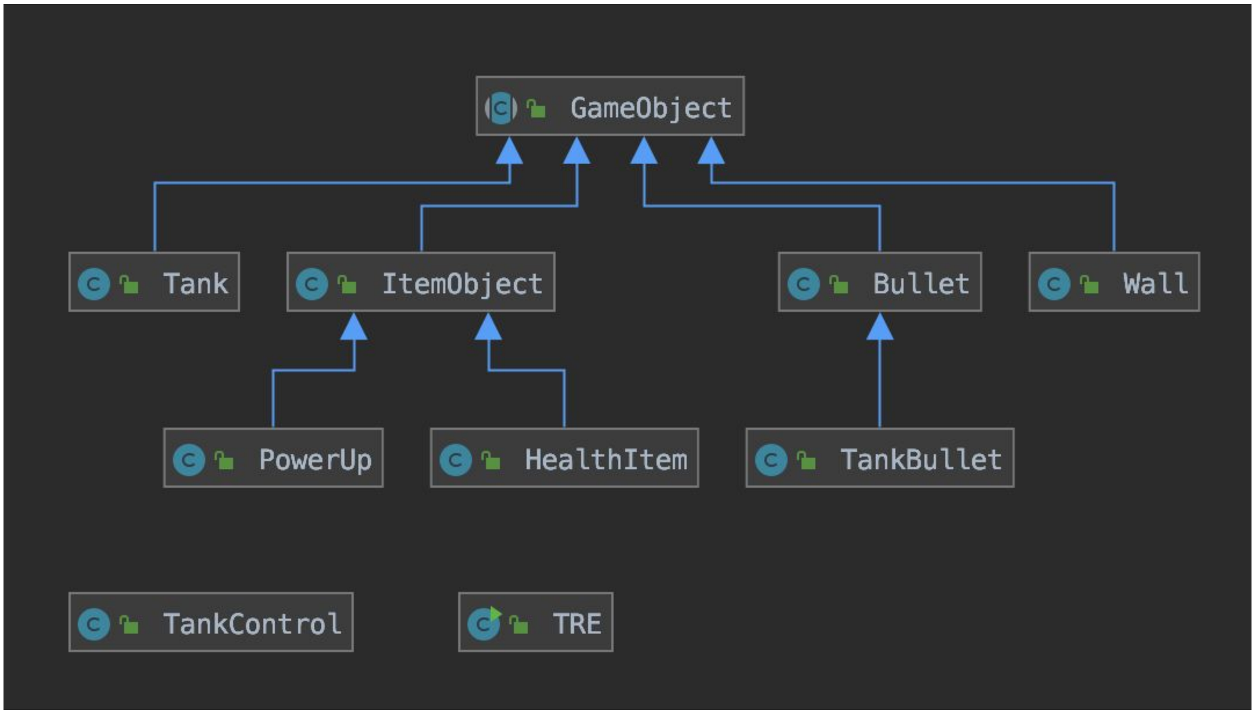
Tank 1	Tank 2
UP: forward	W: forward
DOWN: backward	S: backward
Left: left	A: left
Down: right	D: right
Enter: fire	Space: fire



5. Assumptions Made

When I started designing the game, I assumed the mathematics used, java language to develop java games. Moreover, I use the online material which are provided by professor to do the game designation. When I finished the game, I used my own map.txt and generated the .jar file.

6. Tank Game Diagram



7. Class Descriptions of classes specific to Tank Game

Bullet.java:

public Bullet(Image img, int x, int y, int speed);
public boolean isVisible();
public void draw(ImageObserver obs, Graphics2D g);

GameObject.java:

public GameObject (Image img, int x, int y, int speed);
public Image getImg() {return this,img};
public int getX() {return this.x};
public int getY() {return this.y};
public int getWidth() {return width};
public int getSpeed() {return speed};
public int getHeight() { return height};
public boolean collision(int x, int y, int w, int h);
public void update(Observable obj, Object arg);
public void draw(ImageObserver obs, Graphics2D g);

HealthItem.java:

```
public HealthItem(Image img, int x, int y, int speed);
public void update();
```

ItemObject.java:

```
public ItemObject(Image img, int x, int y, int speed);
public boolean isVisible() {return show};
public void update();
public void daw(ImageObserver obs, Graphics2D g);
```

PowerUp.java:

```
public PowerUp(Image img, int x, int y, int speed);
public void update();
```

Tank.java:

```
public Tank(int x, int y, int vx, int vy, int angle, BufferedImage img);
void toggleUpPressed() {this.UpPressed = true};
void toggleDownPressed() {this.DownPressed = true};
void toggleRightPressed() {this.RightPressed = true};
void toggleLeftPressed() {this.LeftPressed = true};
void unToggleUpPressed() {this.UpPressed = false};
void unToggleDownPressed() {this.DownPressed = false};
void unToggleRightPressed() {this.RightPressed = false};
void unToggleLeftPressed() {this.LeftPressed = false};
public void update();
private void rotateLeft() {this.angle -= this.ROTATIONSPEED};
private void rotateRight() {this.angle += this.ROTATIONSPEED};
private void moveBackwards();
private void moveForwards();
private void checkBorder();
public int getAngle() {return angle};
public int getTankCenterX() {return x + img.getWidth(null) / 2};
public int getTankCenterY() {return y + img.getHeight(null) / 2};
public void ShootBullet(Tank a);
public void paintHealth(Graphics g, int x, int y);
public void paintLives(Graphics g, int x, int y);
public int getLives(){return lives};
public void hitTank() {this.health -= 50};
public void addHealth();
public boolean endGame() {return end};
public String toString() {return "x=" + x + ", y=" + y + ", angle=" + angle};
void drawImage(Graphics g);
```

TankBullet.java:

```
public TankBullet(Image img, int speed, Tank tank, int dmg);
Public void update();
public void draw(ImageObserver obs, Graphics2D g);
```

TankControl.java:

```
public TankControl(Tank t1, int up, int down, int left, int right, int shoot) ;
public void keyTyped(KeyEvent ke);
public void keyPressed(KeyEvent ke);
public void keyReleased(KeyEvent ke);
```

TRE.java:

```
public static void main(String[] args);
private void init();
public ArrayList<TankBullet> getBullet() {return bullet};
public Image getBulletImage() {return normalBullet};
public ArrayList<Wall> getWall() {return wall};
public static TRE getTankGame() {return trex};
public static Tank getTank(int j);
public void drawDetail();
public int tankViewX(Tank tank);
public int tankViewY(Tank tank);
public void drawBackGround();
public void LoadMap();
public void paintComponent(Graphics g);
```

Wall.java:

```
public Wall(Image img, int x, int y, boolean breakableWall);
public boolean isBreakable() {return breakable};
public boolean isRespawning() {return coolDown == 0};
public void breakWall() {coolDown = 700};
public Rectangle getWallRectangle() {return wallRect};
public void draw(ImageObserver obs, Graphics2D g);
public void update();
```

8. Self-reflection on Development process during the term project

For this project, I think it is a big challenge for me to complete the project. Because I haven’t had any game experiences before. Thus, when I started the game, I spent a lot of time studying how to develop a java game. In addition, we didn’t have teammates to work together, so I need to do everything. While I write the coding assignments, I tried to use the online tools which likes to google, GitHub to help me finish the project. Also, I have met a lot of issues when I did the project. For example, the tanks could go through the walls, hid the over-screen, and the tanks’ lives could not update, so I searched online and used online methods to figure out. Of course, I also gained a lot of experiences from the term project. I think the most useful experience are that clear variable name, work your project early because you need a lot of time to debug, and write class diagram before coding. Overall, the term project helps me have a good review of java languages.

9. Project Conclusion

Doing the term project has taught me how to use some logic to design the structure of the software and use abstract class. Create couple classes and connect all of the coding pieces by pieces. There is no doubt that it is a big challenge for me. However, although I have met a lot of challenges, I am willing to spend a lot of time to figure out because I know I will be so happy after the whole project complete. Also, the tank game needs to set the location and size, so the class can be used for the game objects. In addition, I learned a lot as me to develop the software structure. After finishing the term project, it gives me more confidence about doing the java game. In the future, I think I will keep the interests with the java game.