

CS5004 Object-Oriented Design

# CS5004 Final Project Artillery Game

Yaozheng Wang, Guang Hu

April 27, 2022



## 1 Overview

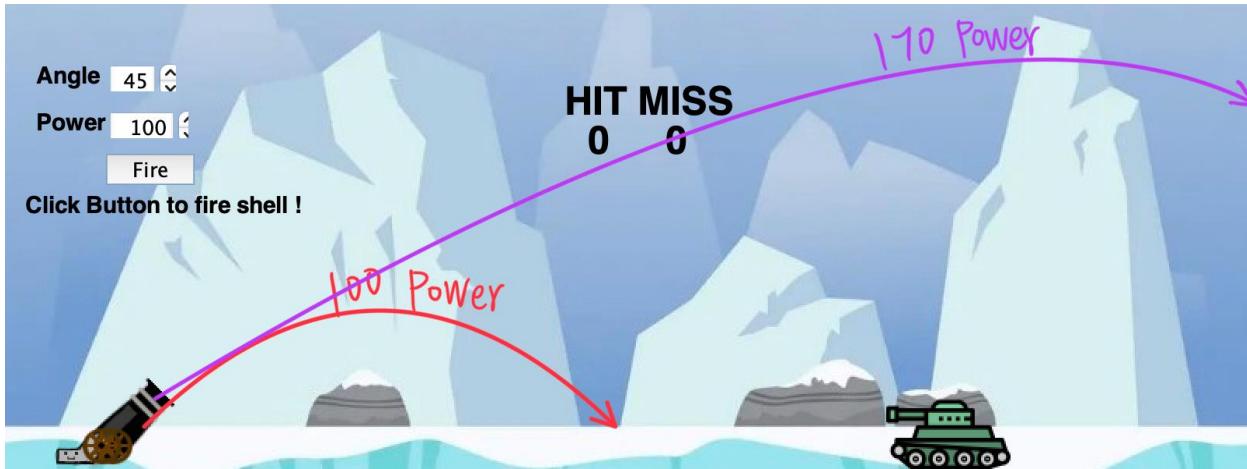
For the cs5004 final project, we selected the second project – artillery game. It has a menu page, a help page, a select map page and a game page. The shell will move through the air in the desired direction and angle.

## 2 How to Play

The game can be run by clicking **CS5004FinalProject.jar**, and the source code can be obtained by unpacking the JAR file. Then you can click the play button to select a map that you like or click the help button to see the tutorial or click the exit button to quit the game.



The game has one cannon and one target tank. Players can adjust the cannon's angle from 0 to 90,



and change the power from 100 to 170.



Each time the player clicks the Fire button, the cannon will fire a shell.

If they miss the target tank, the Miss value will increase by 1; if the shell hits the tank, the Hit value will increase by 1. Because these two parameters influence each other, it increases the complexity and difficulty of the game.

Players need to adjust the angle and power to make the shells hit the tank. Once the shell is fired, it is out of control, so the player must set these two parameters before firing the shell.

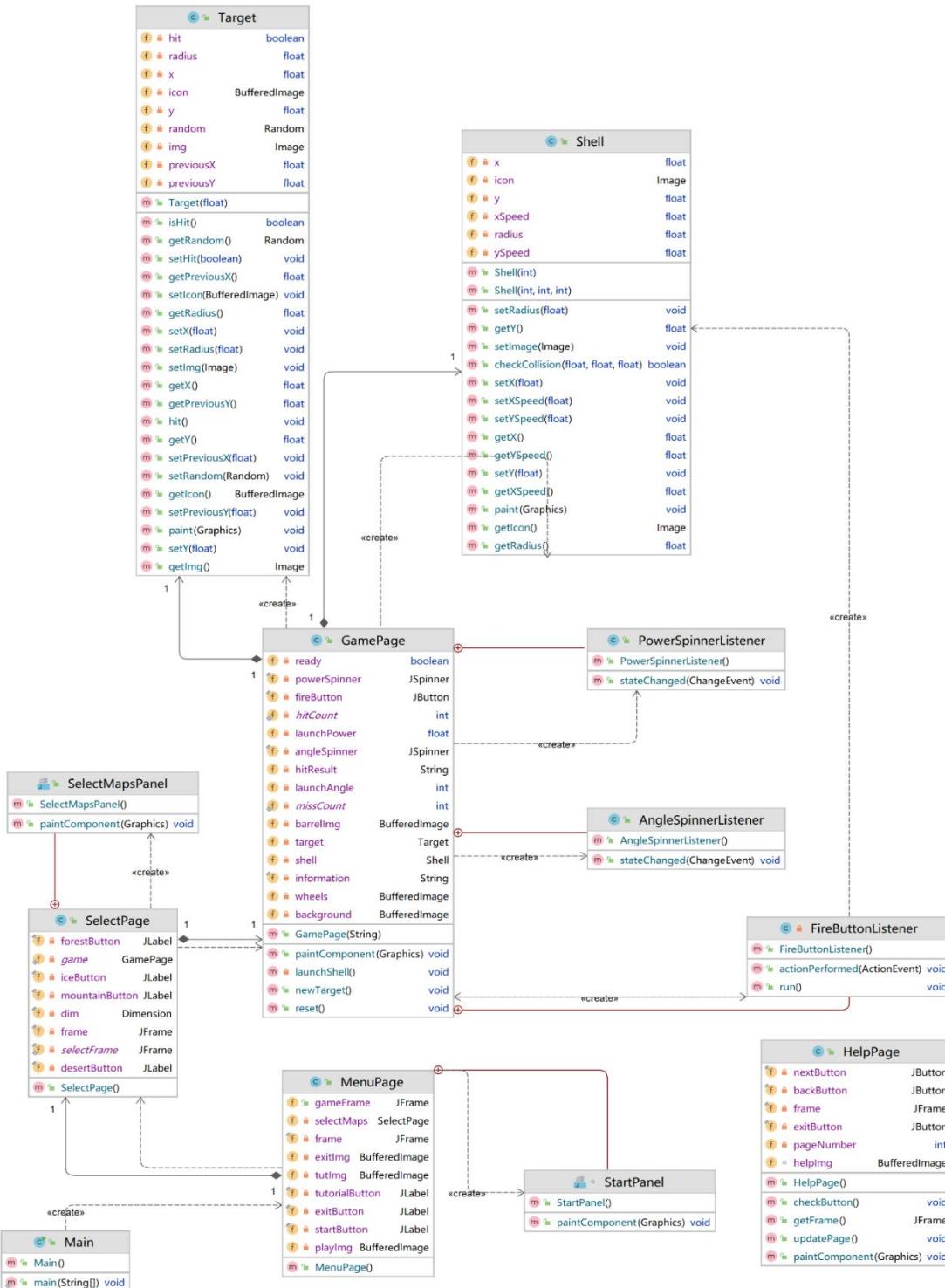
After many rounds of games, players have some experience and can imagine the trajectory of the shells, which increases their chances of hitting.

At the next level, the tank will move to the next random position, giving the player an updated challenge.



### 3 Design

The Class with properties, constructors and methods



## 4. Implementation

### 4.1 Shell Class

The shell class has position and speed and radius attributes. The constructor creates new shell objects at different positions. There are two methods: checkCollision and paint. CheckCollision method gets target position and radius, then computes  $\sqrt{(x1 - x2)^2 + (y1 - y2)^2}$  to compare  $(radius1 + radius2)$ . If the first result isn't less than the second result, the shell hits the target.

### 4.2 Target Class

The target class has position, radius and random. The constructor will create a new target. The random class will create different positions. The hit method will change the target hit condition. if the target be hit, the boolean hit will become true.

### 4.3 MenuPage Class

The menu page class has a game frame and three buttons to control the shell's power, angle and fire. Besides, the frame adds mouse listeners to check mouse action. If the mouse clicks the first button "Play", it will create a new selectPage class to show maps. The "Help" button will show the artillery game tutorial. The last one is the "Exit" button is the way to quit the game.

### 4.4 SelectPage Class

The select page class extends JPanel class, it's a panel type. There are 4 labels containing four different maps: ice, forest, mountain and desert. After the player selects a map, it will create a new GamePage to show the map. So, the selectPage class has mouse listeners to detect which map the player has chosen.

### 4.5 HelpPage Class

The help page class extends JPanel class, it's a component of the frame. It has three buttons: "next", "back" and "exit" to control the tutorial contents. The player can click buttons to control it.

### 4.6 GamePage Class

The game page class is the most important class. Because it implements the whole game. It extends the JPanel class. It contains the two game objects: shell and target. The hit result



information will be shown in the game panel. JSpinners and JButtons add listeners to check the change actions. Two JSpinners are to adjust the shell's power and angle. Fire button will call the launch method to update the shell position.

The launchShell method will use parabolic formulas to compute the shell's new position information, XSpeed, YSpeed. And it will recall the checkCollision method to check if the shell hit the target. But the launchShell method randomly generates the target location by calling the reset method and newTarget method regardless of whether the player hits the target or not.

## 5. Conclusion and Evaluation

In conclusion, our artillery game meets the requirements of the final project. It also implements some interesting things, such as allowing players to choose their favorite game background, to browse the game tutorial and so on.

If time allowed, the game could make players control the cannon positions, they can use their keyboard to move the cannon. And the target can be an AI to shoot players' cannon, which makes the game more competitive and interesting,



## 6. Appendix

Here are the game screenshots.

