MULTIMEDIA UNIVERSITY®

TCP 2201

Object Oriented Analysis and Design

Lecturer Name: Mr. Lee Kian Chin

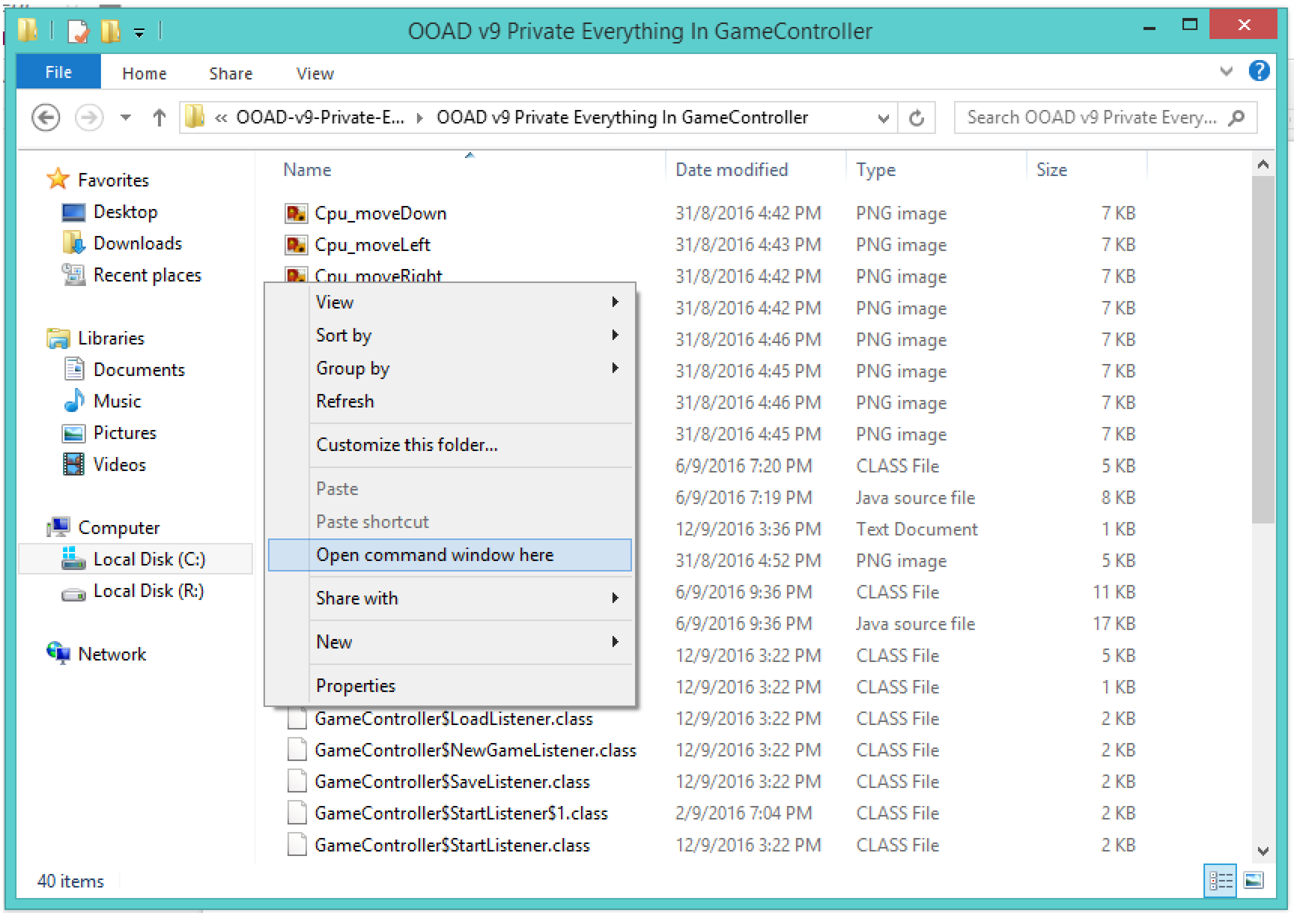
Lecture Session: TC02

Group Members:

Name	ID	Phone	Email
Aaron Patrick Nathaniel	1151301880	0164735933	aaron.nathaniel95@gmail.com
Abdul Muiz bin Abadan	1141327833	0135803195	1141327833@student.mmu.edu.my
Gan Qi Tze	1142700525	0102527511	ganqitze@gmail.com
Lim Yue Hng	1141337945	0166585334	Nlneillim@gmail.com
Tiew Kee Hui (Group Leader)	1151303086	0123625985	huiblabla@hotmail.com

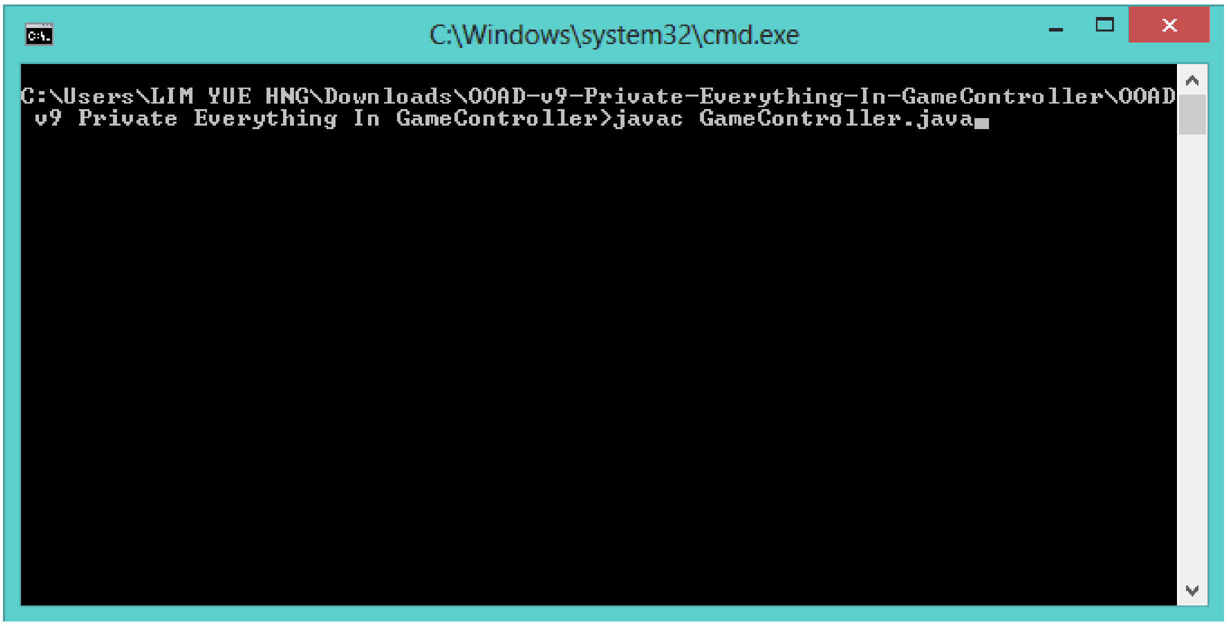
Instructions and User Documentation

1. Extract the .zip file and Shift + Right-click on the folder. Click "Open command window here"



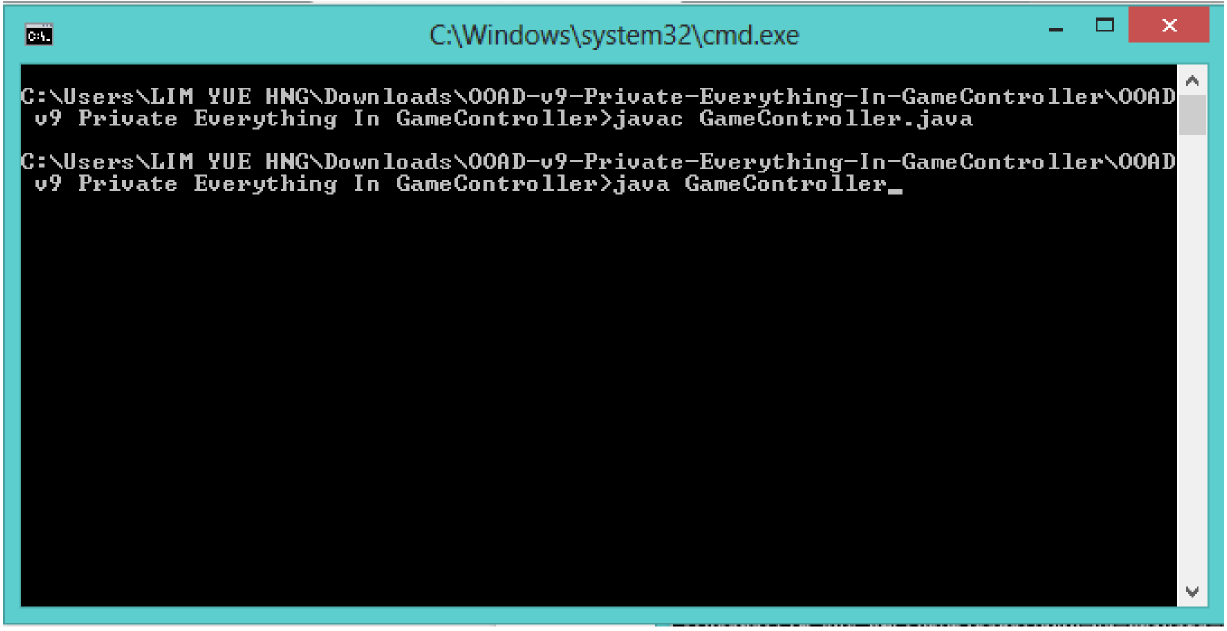
Name	Date modified	Type	Size
Cpu_moveDown	31/8/2016 4:42 PM	PNG image	7 KB
Cpu_moveLeft	31/8/2016 4:43 PM	PNG image	7 KB
Cpu_moveRight	31/8/2016 4:42 PM	PNG image	7 KB
	31/8/2016 4:42 PM	PNG image	7 KB
	31/8/2016 4:46 PM	PNG image	7 KB
	31/8/2016 4:45 PM	PNG image	7 KB
	31/8/2016 4:46 PM	PNG image	7 KB
	31/8/2016 4:45 PM	PNG image	7 KB
	6/9/2016 7:20 PM	CLASS File	5 KB
	6/9/2016 7:19 PM	Java source file	8 KB
	12/9/2016 3:36 PM	Text Document	1 KB
	31/8/2016 4:52 PM	PNG image	5 KB
	6/9/2016 9:36 PM	CLASS File	11 KB
	6/9/2016 9:36 PM	Java source file	17 KB
	12/9/2016 3:22 PM	CLASS File	5 KB
	12/9/2016 3:22 PM	CLASS File	1 KB
GameController$LoadListener.class	12/9/2016 3:22 PM	CLASS File	2 KB
GameController$NewGameListener.class	12/9/2016 3:22 PM	CLASS File	2 KB
GameController$SaveListener.class	12/9/2016 3:22 PM	CLASS File	2 KB
GameController$StartListener$1.class	2/9/2016 7:04 PM	CLASS File	2 KB
GameController$StartListener.class	12/9/2016 3:22 PM	CLASS File	2 KB

2. Enter “javac GameController.java” and wait for it to compile



```
C:\Users\LIM YUE HNG\Downloads\OOAD-v9-Private-Everything-In-GameController\OOAD
 v9 Private Everything In GameController>javac GameController.java
```

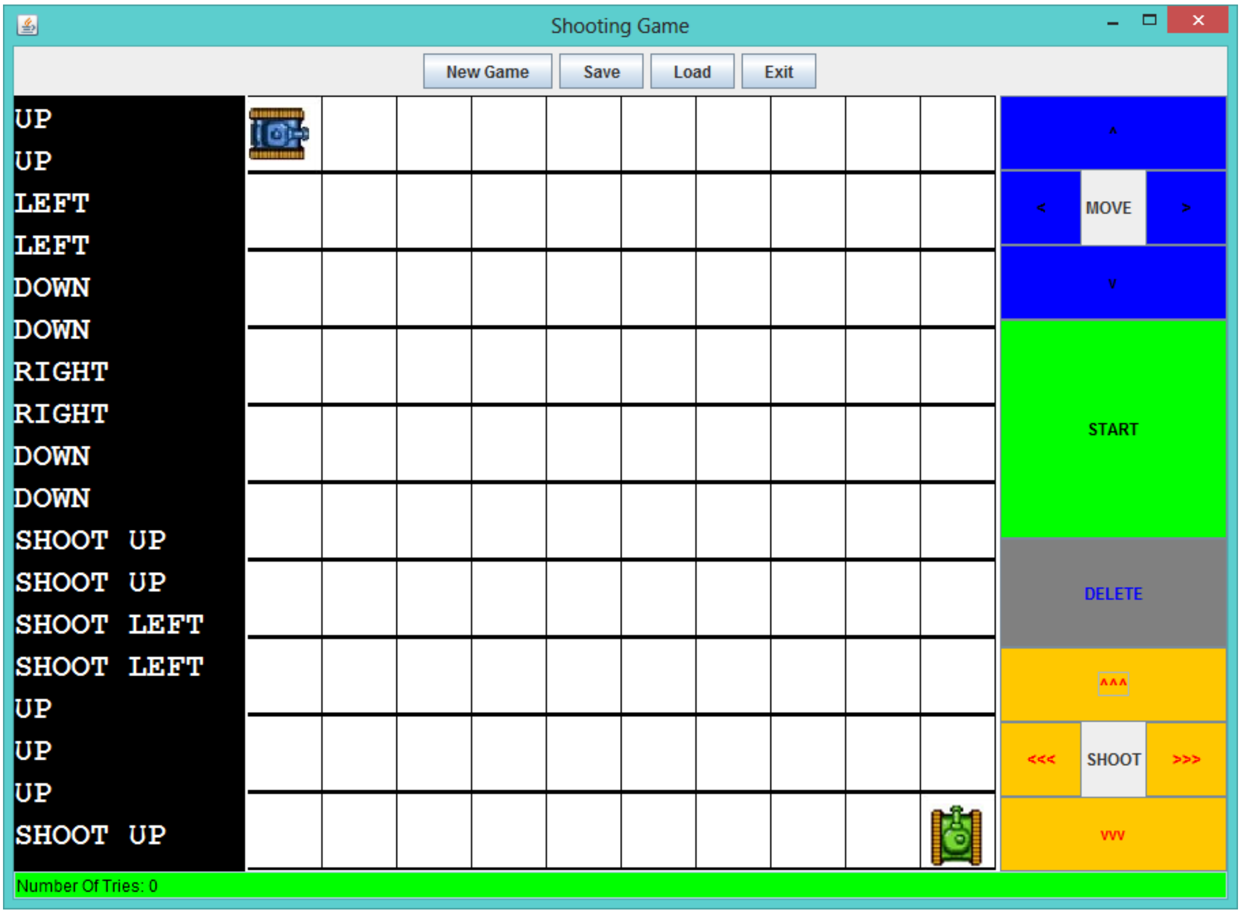
3. Enter “java GameController” and wait for the game screen to load completely



```
C:\Users\LIM YUE HNG\Downloads\OOAD-v9-Private-Everything-In-GameController\OOAD
 v9 Private Everything In GameController>javac GameController.java

C:\Users\LIM YUE HNG\Downloads\OOAD-v9-Private-Everything-In-GameController\OOAD
 v9 Private Everything In GameController>java GameController
```

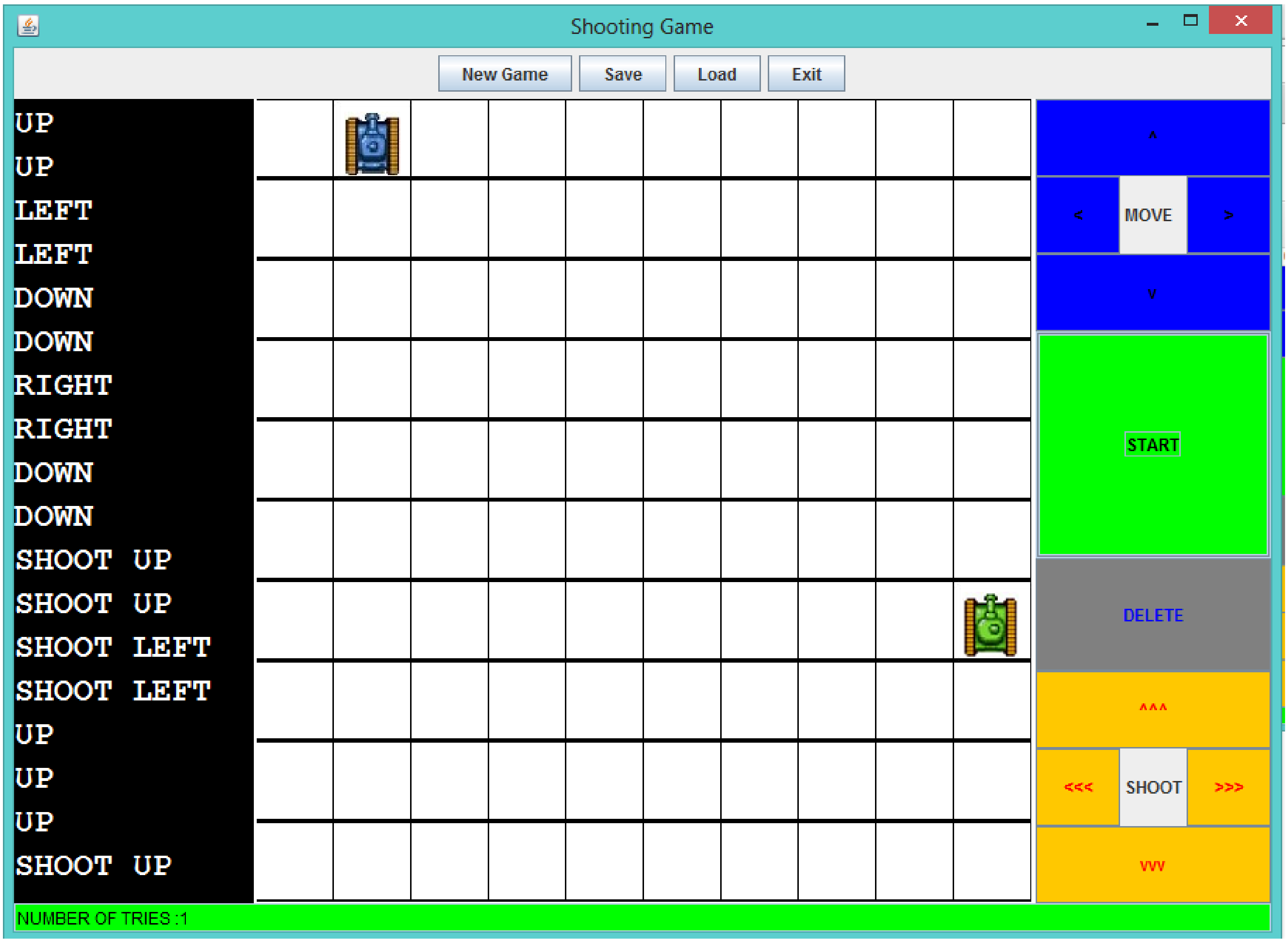
5. Plan your movements (Blue) and shooting (Orange), then press the corresponding arrows (18 moves). The green robot at the bottom right is under your control. You may undo a move by pressing the delete (Grey) button.



```
UP
UP
LEFT
LEFT
DOWN
DOWN
RIGHT
RIGHT
DOWN
DOWN
SHOOT UP
SHOOT UP
SHOOT LEFT
SHOOT LEFT
UP
UP
UP
SHOOT UP
```

Number Of Tries: 0

6. When you are ready, press the green “START” button and watch the events unfold.

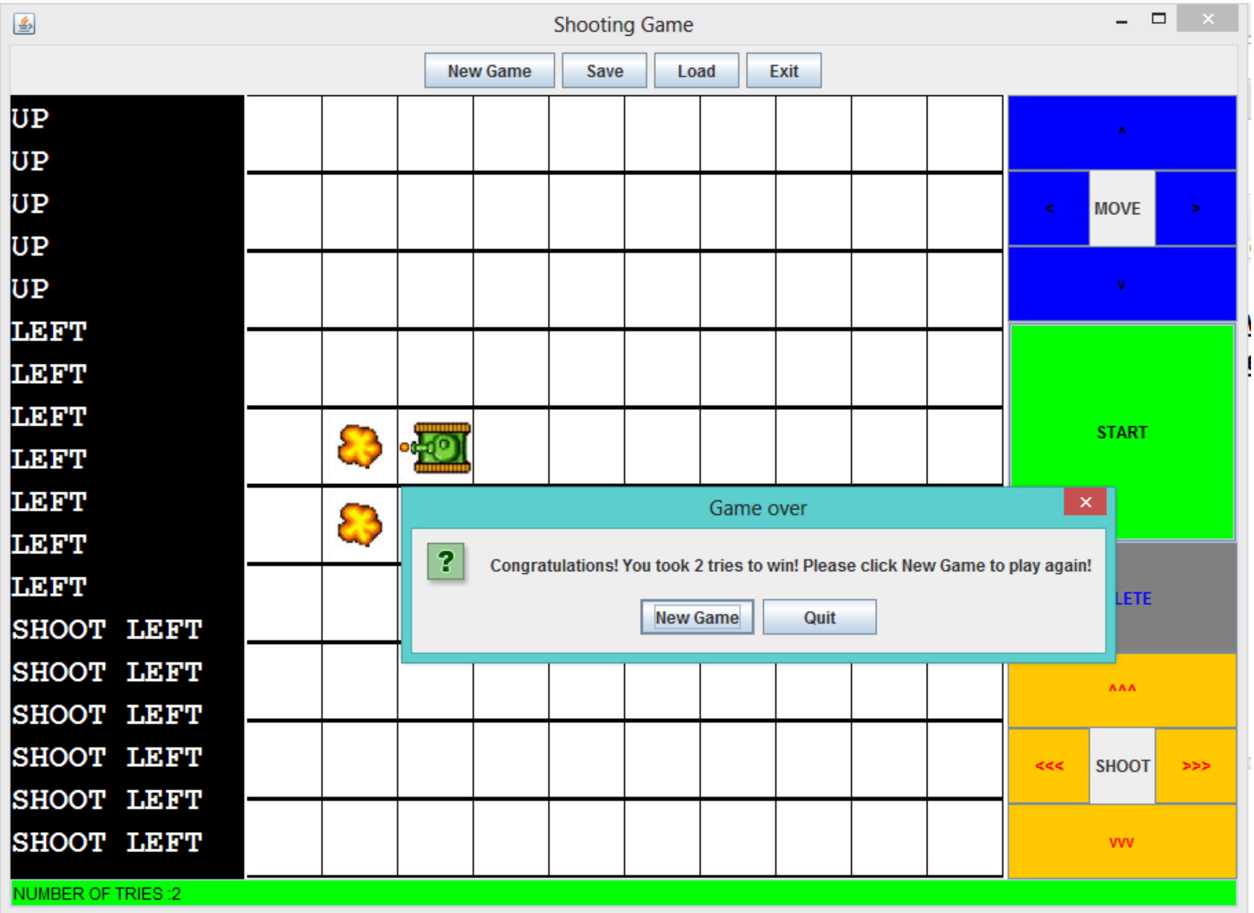


```
UP
UP
LEFT
LEFT
DOWN
DOWN
RIGHT
RIGHT
DOWN
DOWN
SHOOT  UP
SHOOT  UP
SHOOT  LEFT
SHOOT  LEFT
UP
UP
UP
SHOOT  UP
```

NUMBER OF TRIES :1

7. If you have not eliminated the enemy, the game will reset and you are given the chance to try again. See how many tries it takes you to win (bottom left of the game screen).

7.1 If you manage to shoot the enemy robot, you win! A choice will be given to either start a new game or to quit the program.

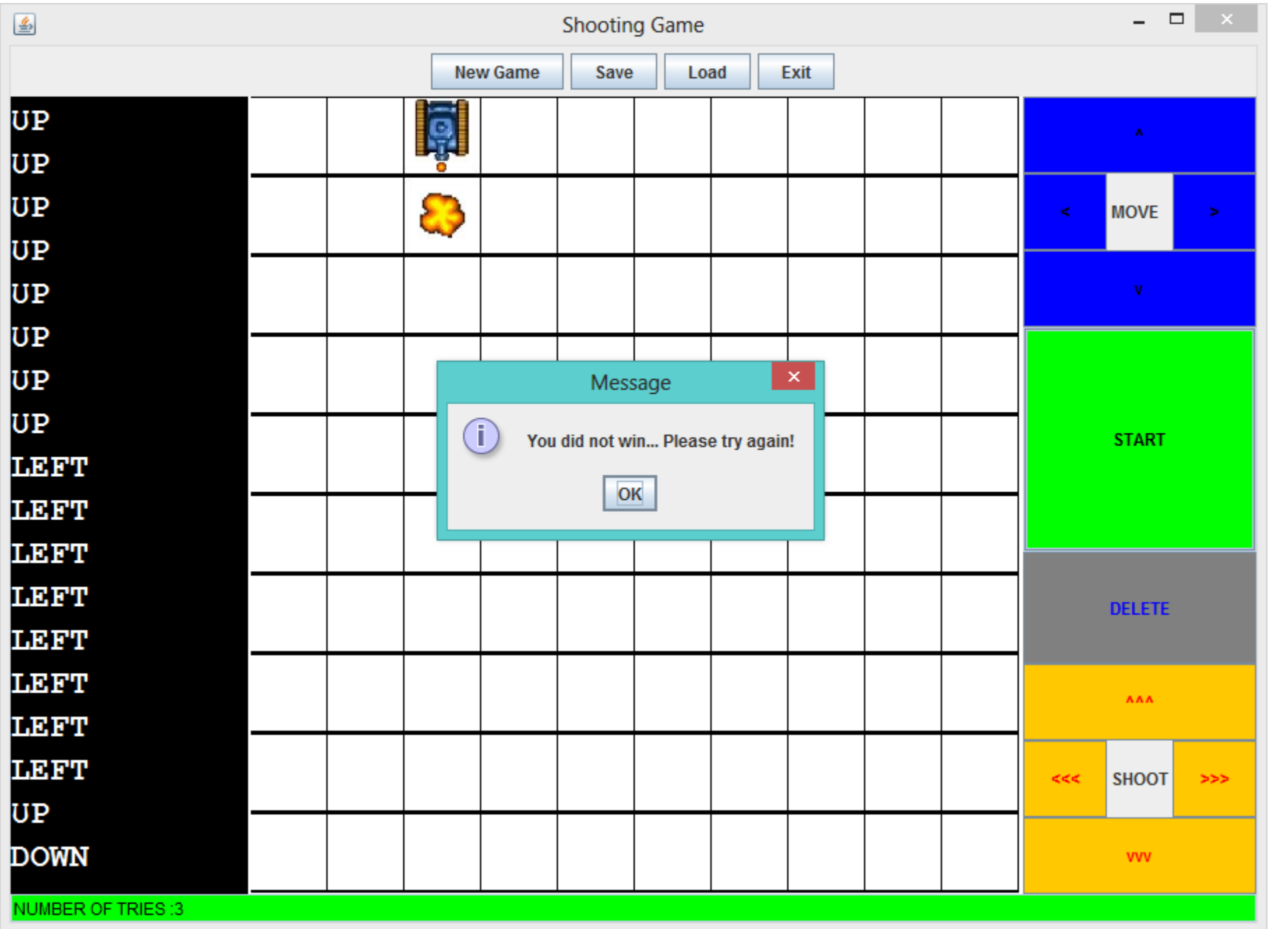


```
UP
UP
UP
UP
UP
LEFT
LEFT
LEFT
LEFT
LEFT
LEFT
LEFT
SHOOT  LEFT
SHOOT  LEFT
SHOOT  LEFT
SHOOT  LEFT
SHOOT  LEFT
SHOOT  LEFT
```

Congratulations! You took 2 tries to win! Please click New Game to play again!

NUMBER OF TRIES :2

7.2 If you are shot or crash into the enemy robot, you lose! You may then try again.



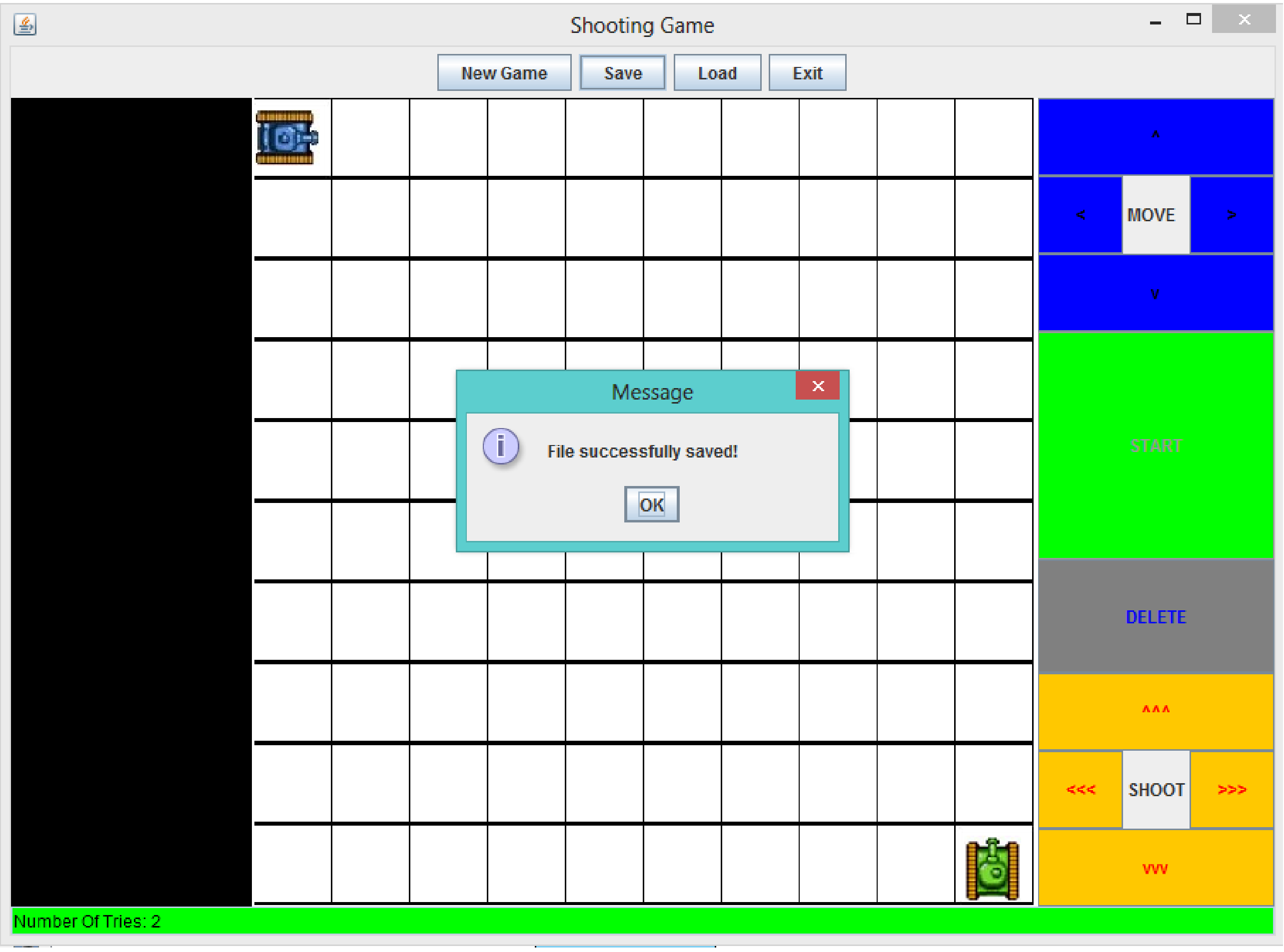
```
UP
UP
UP
UP
UP
UP
UP
UP
LEFT
LEFT
LEFT
LEFT
LEFT
LEFT
LEFT
LEFT
UP
DOWN
```

You did not win... Please try again!

NUMBER OF TRIES :3

8. You may choose to press “New Game” on the top toolbar to reset the game and the number of tries recorded. However, a new sequence of moves will be generated for the computer robot and you have to figure out how to beat it again.

9. The option to “Save” at the top toolbar will save your moves that you have entered and the amount of tries made. (Only saves completed rounds of moving and shooting)



Shooting Game

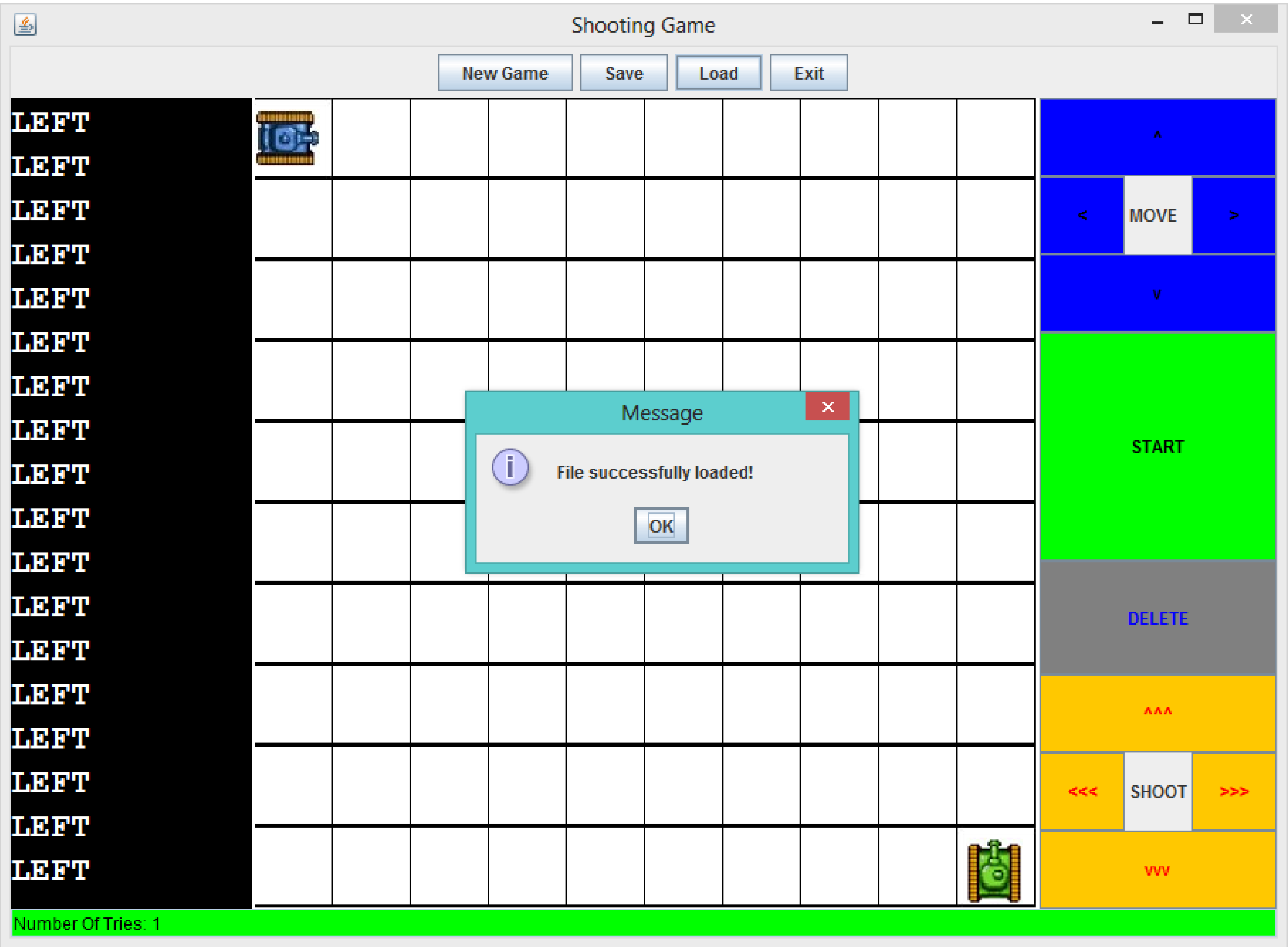
Message

File successfully saved!

OK

Number Of Tries: 2

10. You may click the “Load” button to load the move set of a previously saved round.

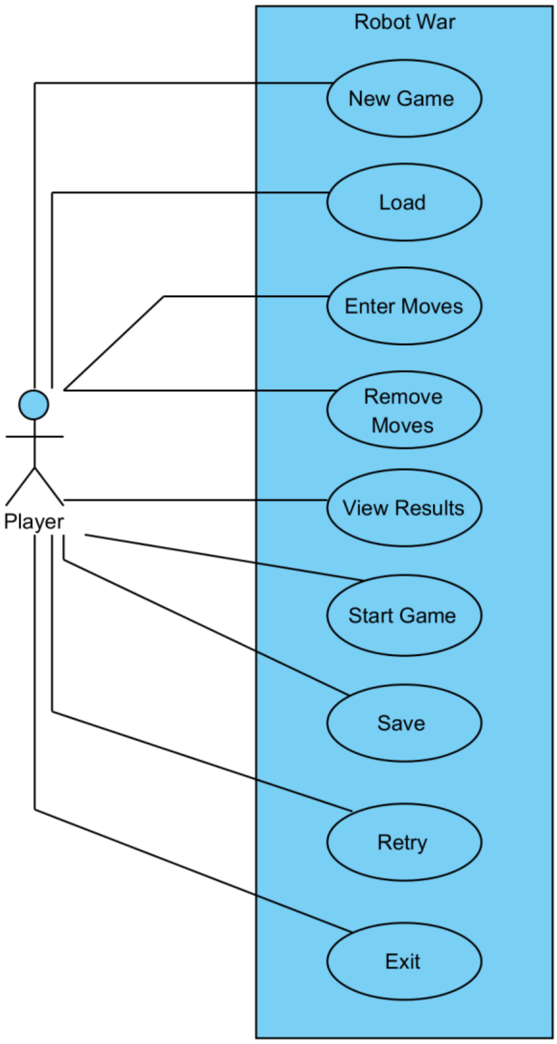


File successfully loaded!

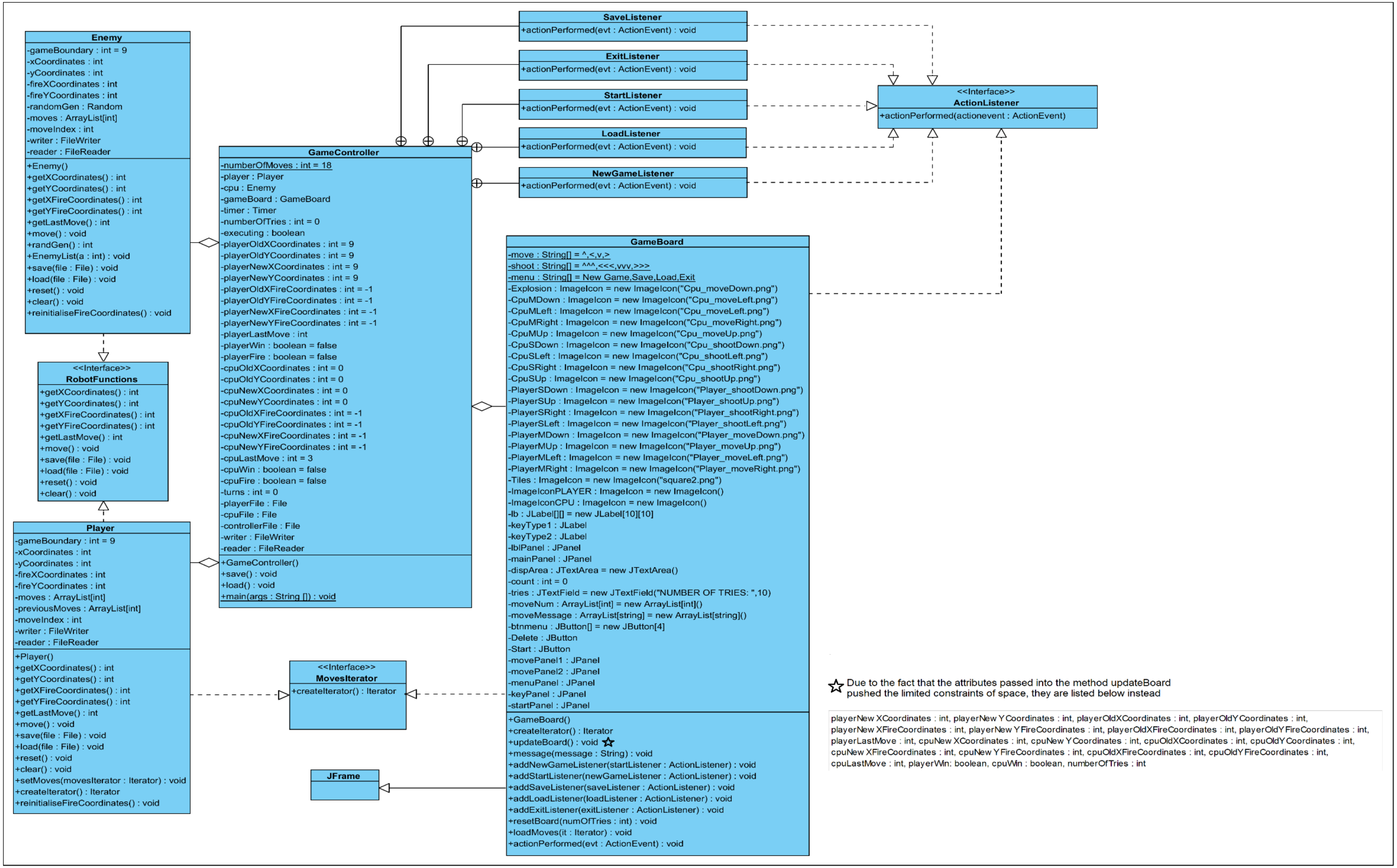
Number Of Tries: 1

11. Press the “Exit” button at the top toolbar when you have played enough. Hope you had fun!

Use Case Diagram



Class Diagram (Please view the ClassDiagram.png image included to zoom in)


Enemy
-gameBoundary : int = 9
-xCoordinates : int
-yCoordinates : int
-fireXCoordinates : int
-fireYCoordinates : int
-randomGen : Random
-moves : ArrayList[int]
-moveIndex : int
-writer : FileWriter
-reader : FileReader
+Enemy()
+getXCoordinates() : int
+getYCoordinates() : int
+getXFireCoordinates() : int
+getYFireCoordinates() : int
+getLastMove() : int
+move() : void
+randGen() : int
+EnemyList(a : int) : void
+save(file : File) : void
+load(file : File) : void
+reset() : void
+clear() : void
+reinitialiseFireCoordinates() : void
<<Interface>>
RobotFunctions
+getXCoordinates() : int
+getYCoordinates() : int
+getXFireCoordinates() : int
+getYFireCoordinates() : int
+getLastMove() : int
+move() : void
+save(file : File) : void
+load(file : File) : void
+reset() : void
+clear() : void
Player
-gameBoundary : int = 9
-xCoordinates : int
-yCoordinates : int
-fireXCoordinates : int
-fireYCoordinates : int
-moves : ArrayList[int]
-previousMoves : ArrayList[int]
-moveIndex : int
-writer : FileWriter
-reader : FileReader
+Player()
+getXCoordinates() : int
+getYCoordinates() : int
+getXFireCoordinates() : int
+getYFireCoordinates() : int
+getLastMove() : int
+move() : void
+save(file : File) : void
+load(file : File) : void
+reset() : void
+clear() : void
+setMoves(movesIterator : Iterator) : void
+createIterator() : Iterator
+reinitialiseFireCoordinates() : void
GameController
-numberOfMoves : int = 18
-player : Player
-cpu : Enemy
-gameBoard : GameBoard
-timer : Timer
-numberOfTries : int = 0
-executing : boolean
-playerOldXCoordinates : int = 9
-playerOldYCoordinates : int = 9
-playerNewXCoordinates : int = 9
-playerNewYCoordinates : int = 9
-playerOldXFireCoordinates : int = -1
-playerOldYFireCoordinates : int = -1
-playerNewXFireCoordinates : int = -1
-playerNewYFireCoordinates : int = -1
-playerLastMove : int
-playerWin : boolean = false
-playerFire : boolean = false
-cpuOldXCoordinates : int = 0
-cpuOldYCoordinates : int = 0
-cpuNewXCoordinates : int = 0
-cpuNewYCoordinates : int = 0
-cpuOldXFireCoordinates : int = -1
-cpuOldYFireCoordinates : int = -1
-cpuNewXFireCoordinates : int = -1
-cpuNewYFireCoordinates : int = -1
-cpuLastMove : int = 3
-cpuWin : boolean = false
-cpuFire : boolean = false
-turns : int = 0
-playerFile : File
-cpuFile : File
-controllerFile : File
-writer : FileWriter
-reader : FileReader
+GameController()
+save() : void
+load() : void
+main(args : String []) : void
<<Interface>>
MovesIterator
+createIterator() : Iterator
JFrame
SaveListener
+actionPerformed(evt : ActionEvent) : void
ExitListener
+actionPerformed(evt : ActionEvent) : void
StartListener
+actionPerformed(evt : ActionEvent) : void
LoadListener
+actionPerformed(evt : ActionEvent) : void
NewGameListener
+actionPerformed(evt : ActionEvent) : void
<<Interface>>
ActionListener
+actionPerformed(actionevent : ActionEvent)
GameBoard
-move : String[] = ^,<,v,>
-shoot : String[] = ^^^,<<<,vvv,>>>
-menu : String[] = New Game,Save,Load,Exit
-Explosion : ImageIcon = new ImageIcon("Cpu_moveDown.png")
-CpuMDown : ImageIcon = new ImageIcon("Cpu_moveLeft.png")
-CpuMLeft : ImageIcon = new ImageIcon("Cpu_moveLeft.png")
-CpuMRight : ImageIcon = new ImageIcon("Cpu_moveRight.png")
-CpuMUp : ImageIcon = new ImageIcon("Cpu_moveUp.png")
-CpuSDown : ImageIcon = new ImageIcon("Cpu_shootDown.png")
-CpuSLeft : ImageIcon = new ImageIcon("Cpu_shootLeft.png")
-CpuSRight : ImageIcon = new ImageIcon("Cpu_shootRight.png")
-CpuSUp : ImageIcon = new ImageIcon("Cpu_shootUp.png")
-PlayerSDown : ImageIcon = new ImageIcon("Player_shootDown.png")
-PlayerSUp : ImageIcon = new ImageIcon("Player_shootUp.png")
-PlayerSRight : ImageIcon = new ImageIcon("Player_shootRight.png")
-PlayerSLeft : ImageIcon = new ImageIcon("Player_shootLeft.png")
-PlayerMDown : ImageIcon = new ImageIcon("Player_moveDown.png")
-PlayerMUp : ImageIcon = new ImageIcon("Player_moveUp.png")
-PlayerMLeft : ImageIcon = new ImageIcon("Player_moveLeft.png")
-PlayerMRight : ImageIcon = new ImageIcon("Player_moveRight.png")
-Tiles : ImageIcon = new ImageIcon("square2.png")
-ImageIconPLAYER : ImageIcon = new ImageIcon()
-ImageIconCPU : ImageIcon = new ImageIcon()
-lb : JLabel[][] = new JLabel[10][10]
-keyType1 : JLabel
-keyType2 : JLabel
-lblPanel : JPanel
-mainPanel : JPanel
-dispArea : JTextArea = new JTextArea()
-count : int = 0
-tries : JTextField = new JTextField("NUMBER OF TRIES: ",10)
-moveNum : ArrayList[int] = new ArrayList[int]()
-moveMessage : ArrayList[string] = new ArrayList[string]()
-btnmenu : JButton[] = new JButton[4]
-Delete : JButton
-Start : JButton
-movePanel1 : JPanel
-movePanel2 : JPanel
-menuPanel : JPanel
-keyPanel : JPanel
-startPanel : JPanel
+GameBoard()
+createIterator() : Iterator
+updateBoard() : void ☆
+message(message : String) : void
+addNewGameListener(startListener : ActionListener) : void
+addStartListener(newGameListener : ActionListener) : void
+addSaveListener(saveListener : ActionListener) : void
+addLoadListener(loadListener : ActionListener) : void
+addExitListener(exitListener : ActionListener) : void
+resetBoard(numOfTries : int) : void
+loadMoves(it : Iterator) : void
+actionPerformed(evt : ActionEvent) : void

☆ Due to the fact that the attributes passed into the method updateBoard pushed the limited constraints of space, they are listed below instead

playerNew XCoordinates : int, playerNew Y Coordinates : int, playerOldXCoordinates : int, playerOldY Coordinates : int, playerNew XFireCoordinates : int, playerNew Y FireCoordinates : int, playerOldXFireCoordinates : int, playerOldY FireCoordinates : int, playerLastMove : int, cpuNew XCoordinates : int, cpuNew Y Coordinates : int, cpuOldXCoordinates : int, cpuOldY Coordinates : int, cpuNew XFireCoordinates : int, cpuNew Y FireCoordinates : int, cpuOldXFireCoordinates : int, cpuOldY FireCoordinates : int, cpuLastMove : int, playerWin: boolean, cpuWin : boolean, numberOfTries : int

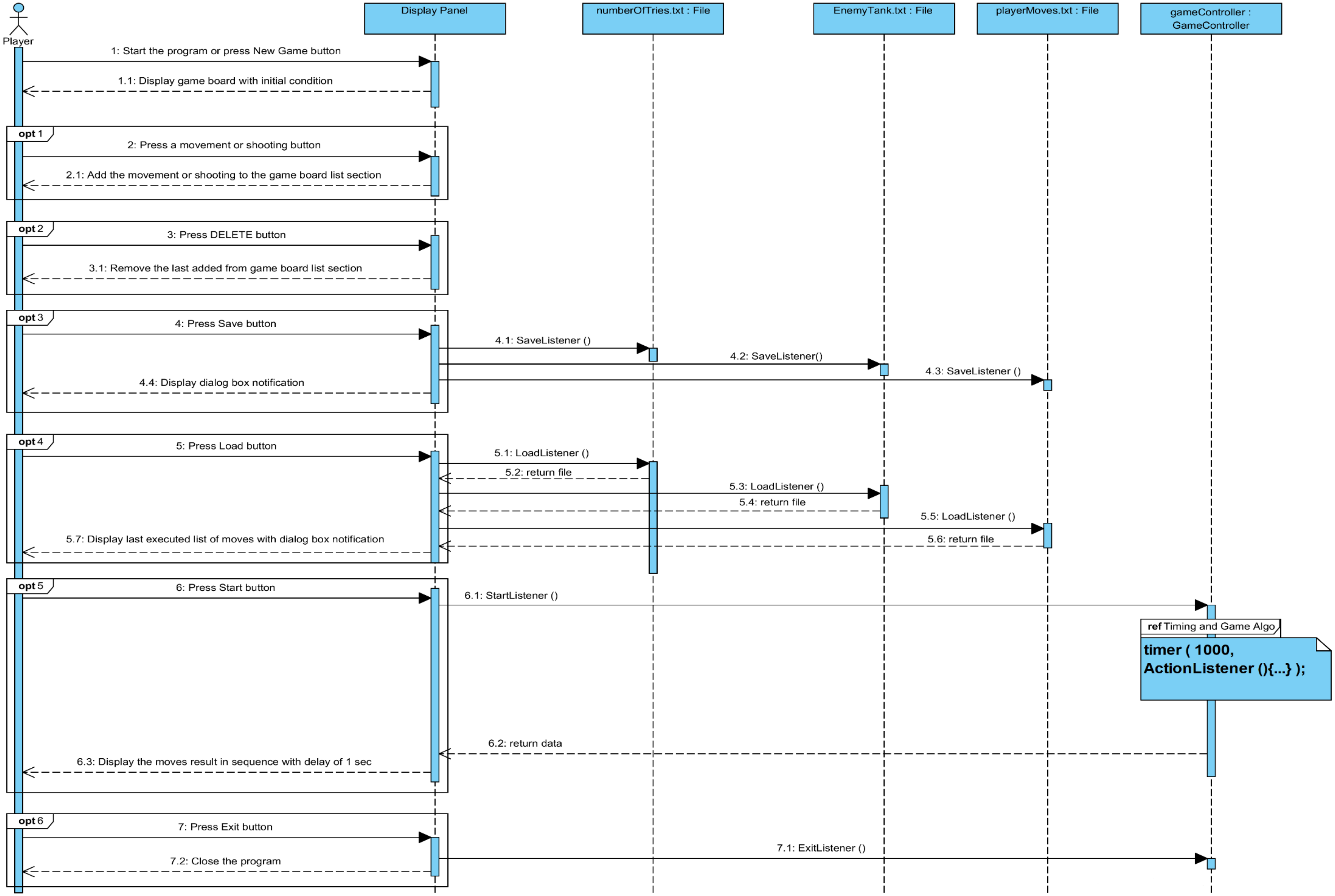
Explanation On Classes Involved In Design Pattern

The design pattern we chose to use is the Iterator Design Pattern. The interface which is used to implement it is the MovesIterator class. The Player class and GameBoard class both realize the MovesIterator class.

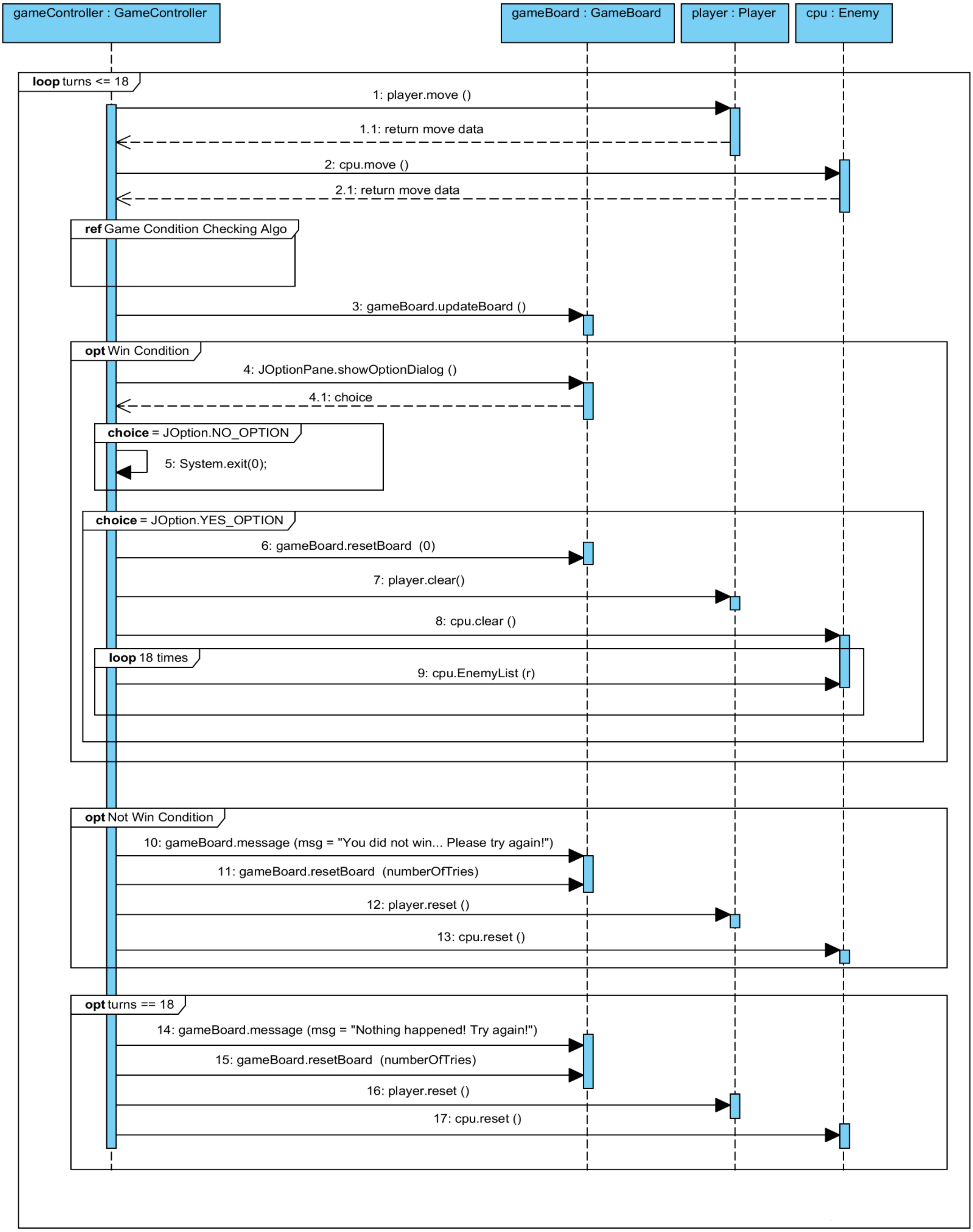
The GameBoard class uses the Iterator Design Pattern when it wants to pass the 18 moves of the user to the Player class via the GameController.

The Player class uses the Iterator Design Pattern when the user loads a save file. It has to load the 18 moves in the save file into its own class (Player class) and also pass the moves to the GameBoard class so that the GameBoard class can display the 18 moves loaded from the save file.

Sequence Diagram 1 – Main functions of the game



Sequence Diagram 2 – Description of the timer in gameController


gameController : GameController
gameBoard : GameBoard
player : Player
cpu : Enemy
loop turns <= 18
1: player.move ()
1.1: return move data
2: cpu.move ()
2.1: return move data
ref Game Condition Checking Algo
3: gameBoard.updateBoard ()
opt Win Condition
4: JOptionPane.showOptionDialog ()
4.1: choice
choice = JOption.NO_OPTION
5: System.exit(0);
choice = JOption.YES_OPTION
6: gameBoard.resetBoard (0)
7: player.clear()
8: cpu.clear ()
loop 18 times
9: cpu.EnemyList (r)
opt Not Win Condition
10: gameBoard.message (msg = "You did not win... Please try again!")
11: gameBoard.resetBoard (numberOfTries)
12: player.reset ()
13: cpu.reset ()
opt turns == 18
14: gameBoard.message (msg = "Nothing happened! Try again!")
15: gameBoard.resetBoard (numberOfTries)
16: player.reset ()
17: cpu.reset ()