TCP2201
Object-Oriented Analysis and Design

TC02
GROUP ASSIGNMENT

Lecturer:
Lee Kian Chin

Project Name:
World War 4.0

Group Members:

Nicholas Tan Yu Zhe	1142701655	nicholas.290696@gmail.com	0167768182
Choo Jia Sheng	1142700814	jason952002@gmail.com	01110705798
Wong Zi Xiang	1142701102	wongzx96@gmail.com	01116906368
Nicolas Raj a/l Paul Durai	1142700982	me@nicolasraj.com	0107669780
Muhammad Hisyam bin Bujang	1142701234	flundut@gmail.com	0128869204

Instructions

1.Compile the file by using the following code:

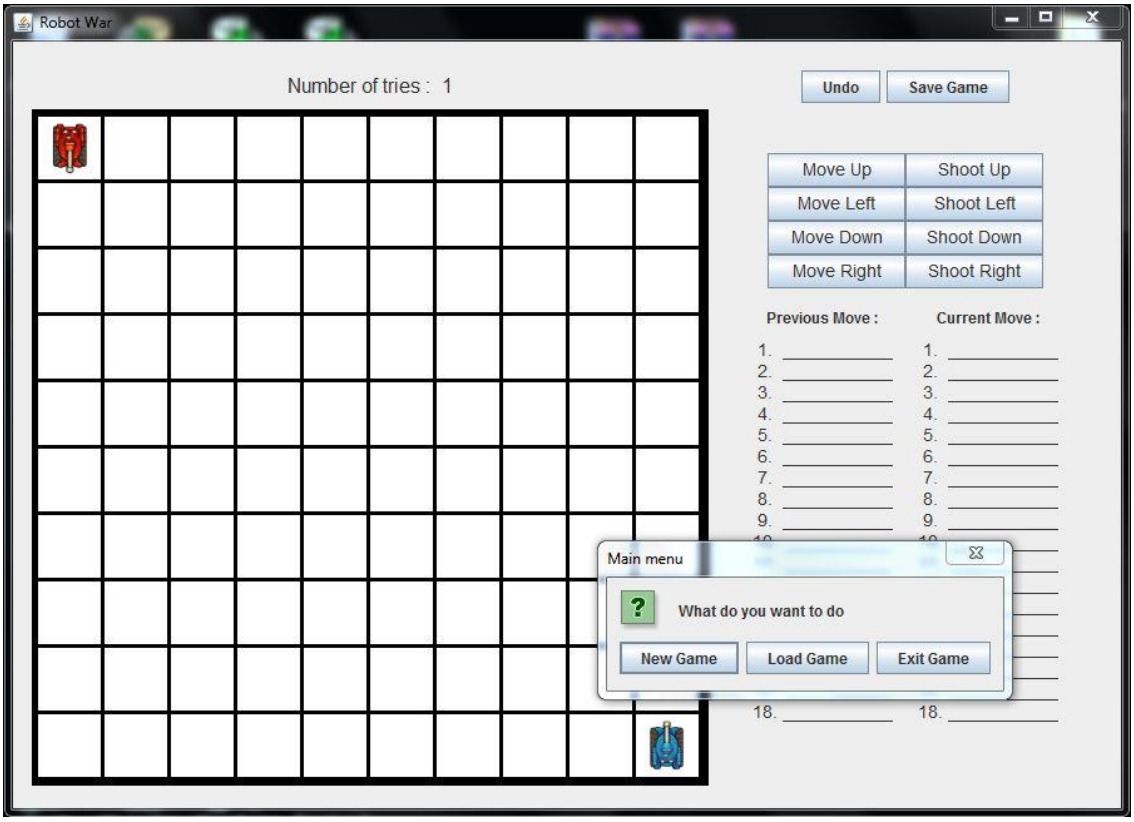
```
javac MVCGame.java
```

2.Run the game by using the following code:

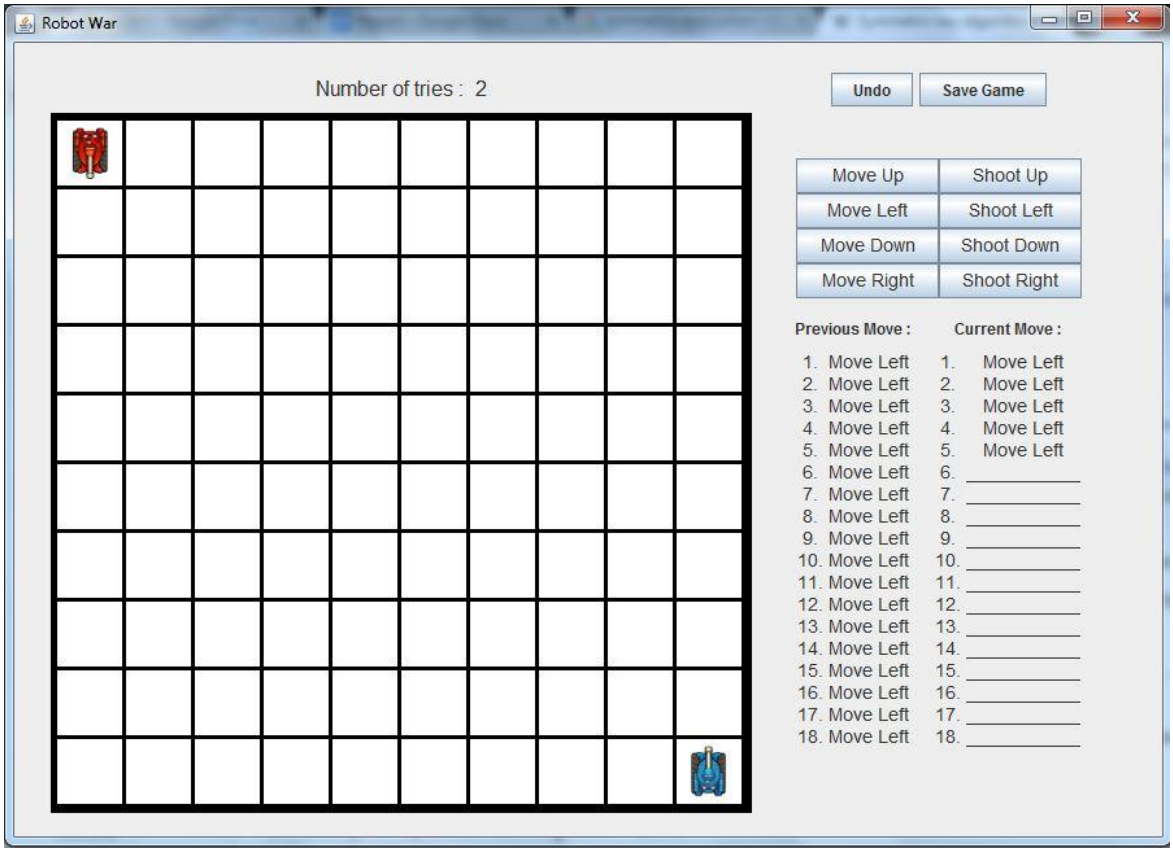
```
java MVCGame
```

User Documentation

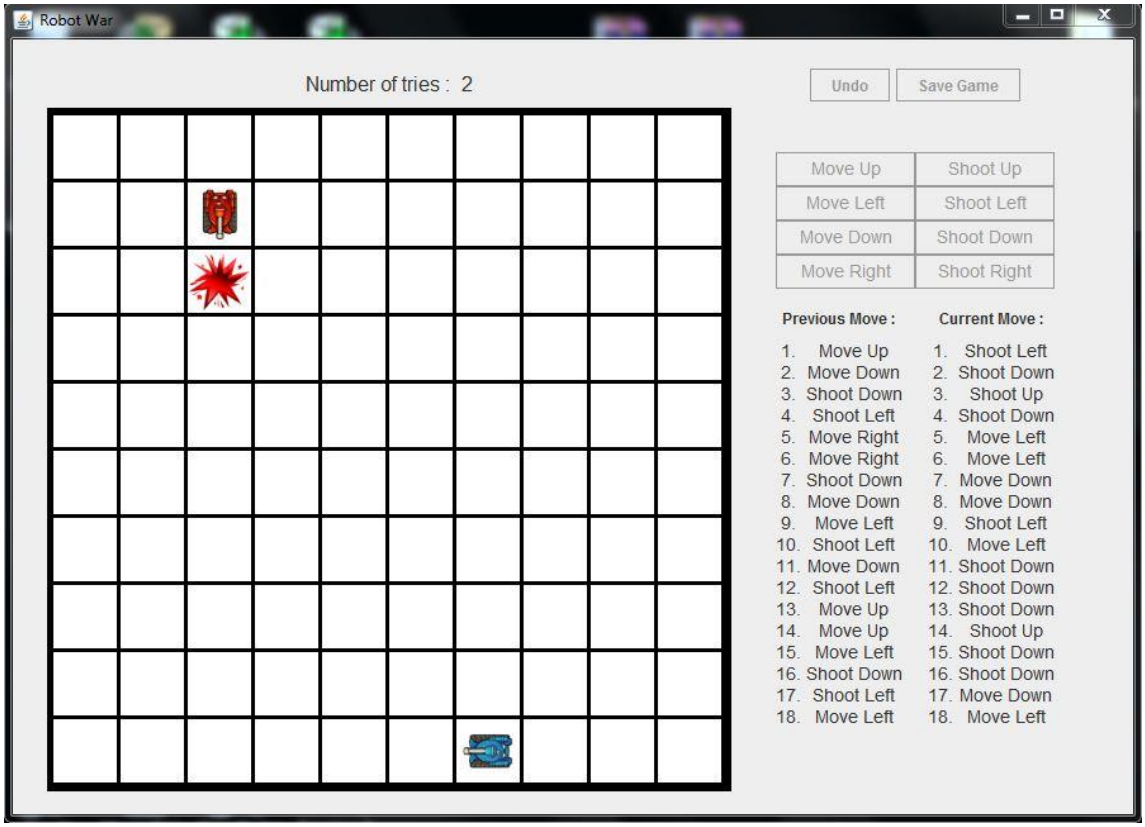
1. Select between New Game, Load Game and Exit Game.



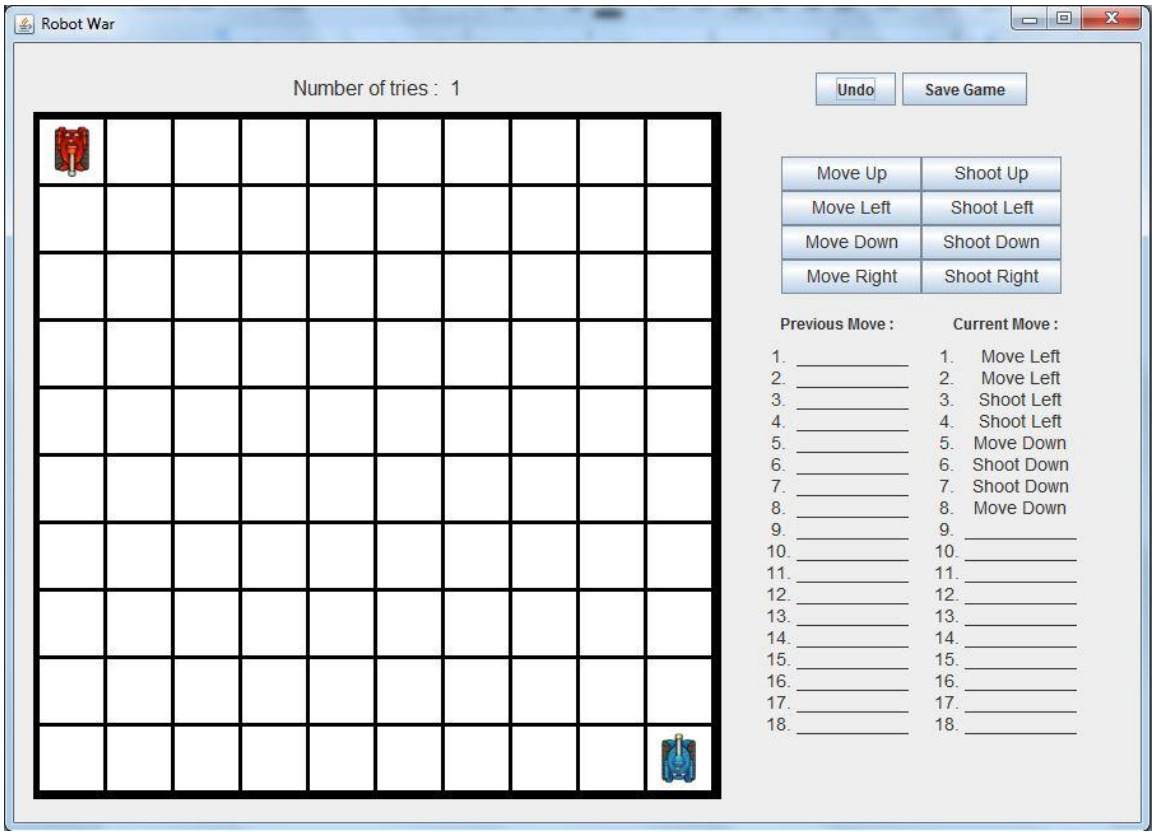
2. If Load Game is selected the program continues from the load file.



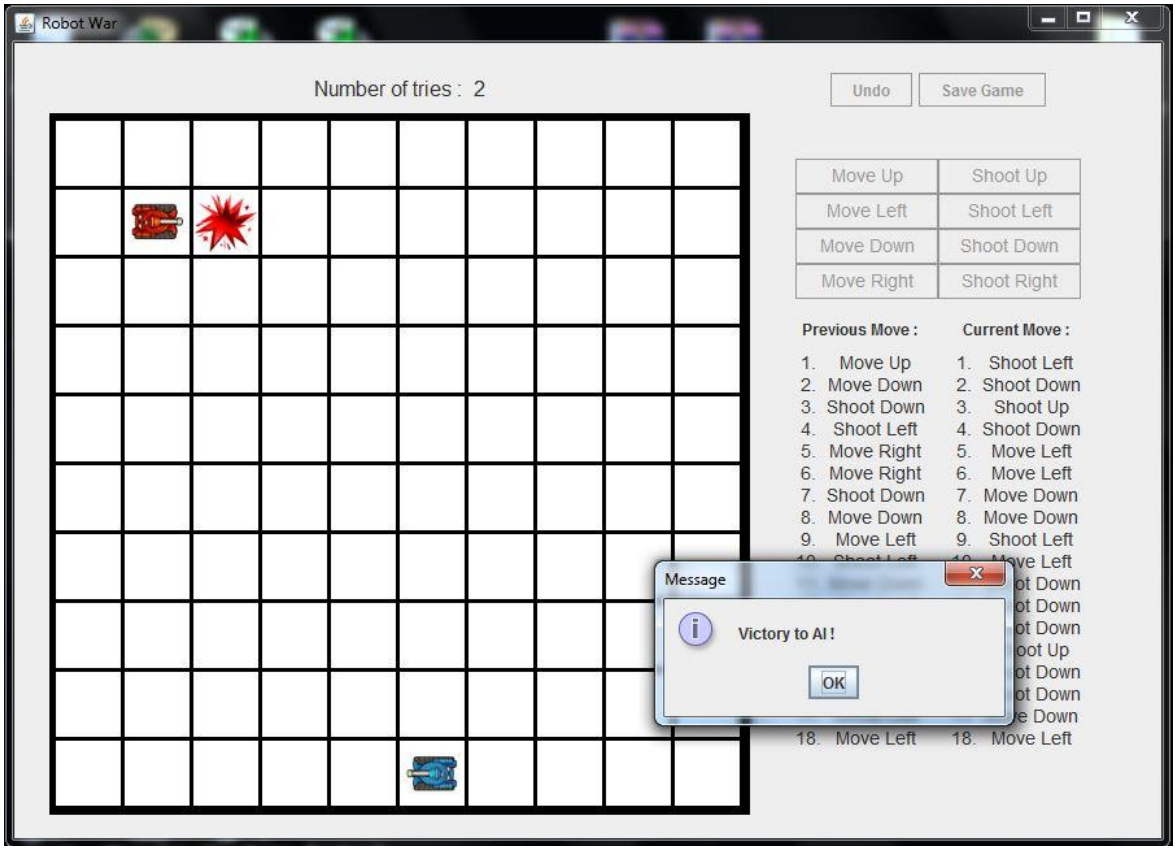
3. If New Game is selected you must choose 18 sequences that you wish to be executed by your tank to destroy the AI tank.



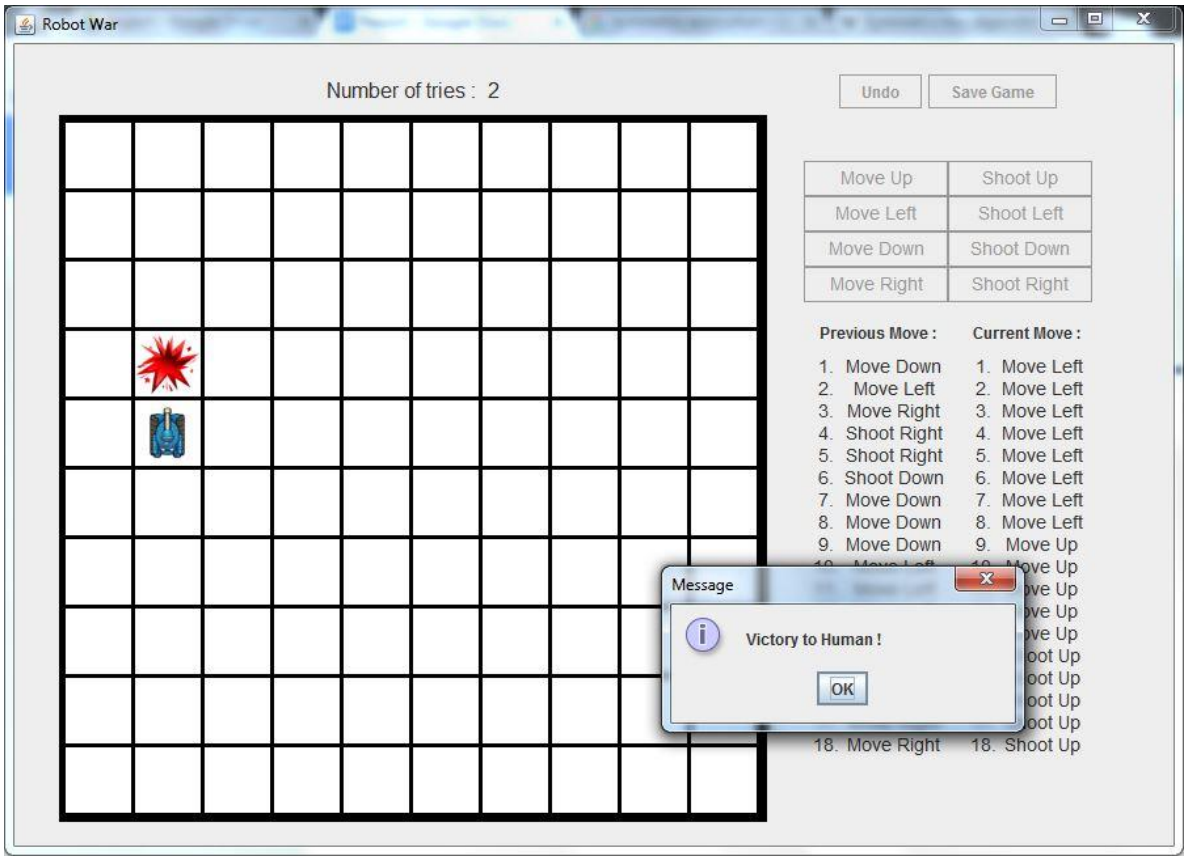
4. You can undo your moves as many times as you want.



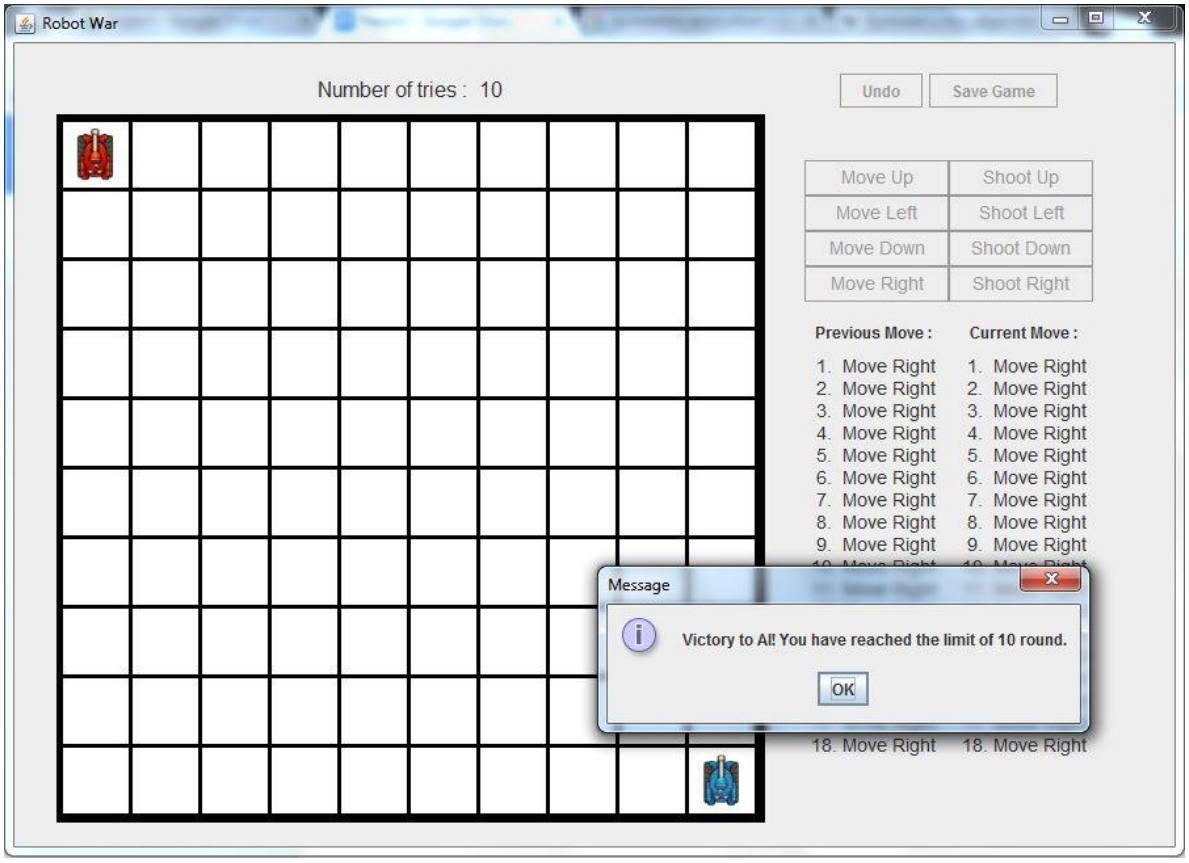
5. If you fail to to destroy the tank, you can try again for 9 times.



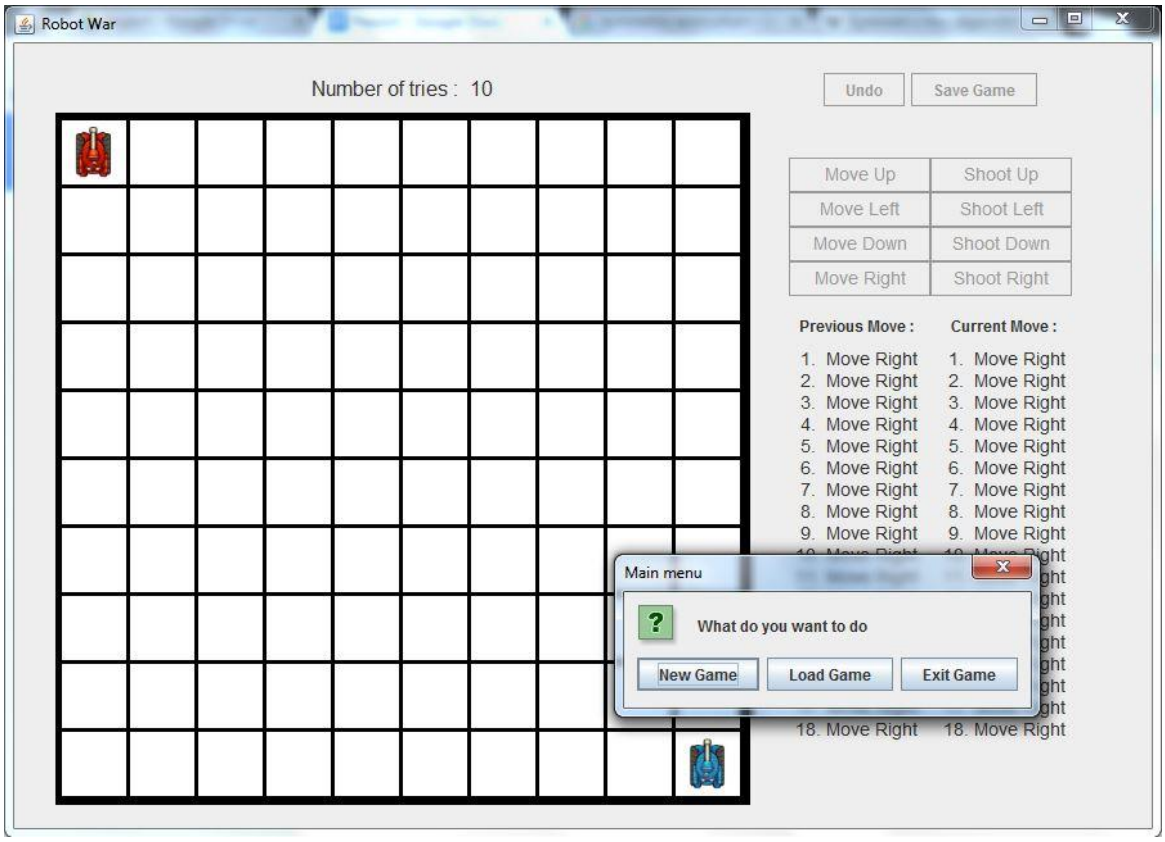
6. If you manage to destroy the tank a victory message is displayed.



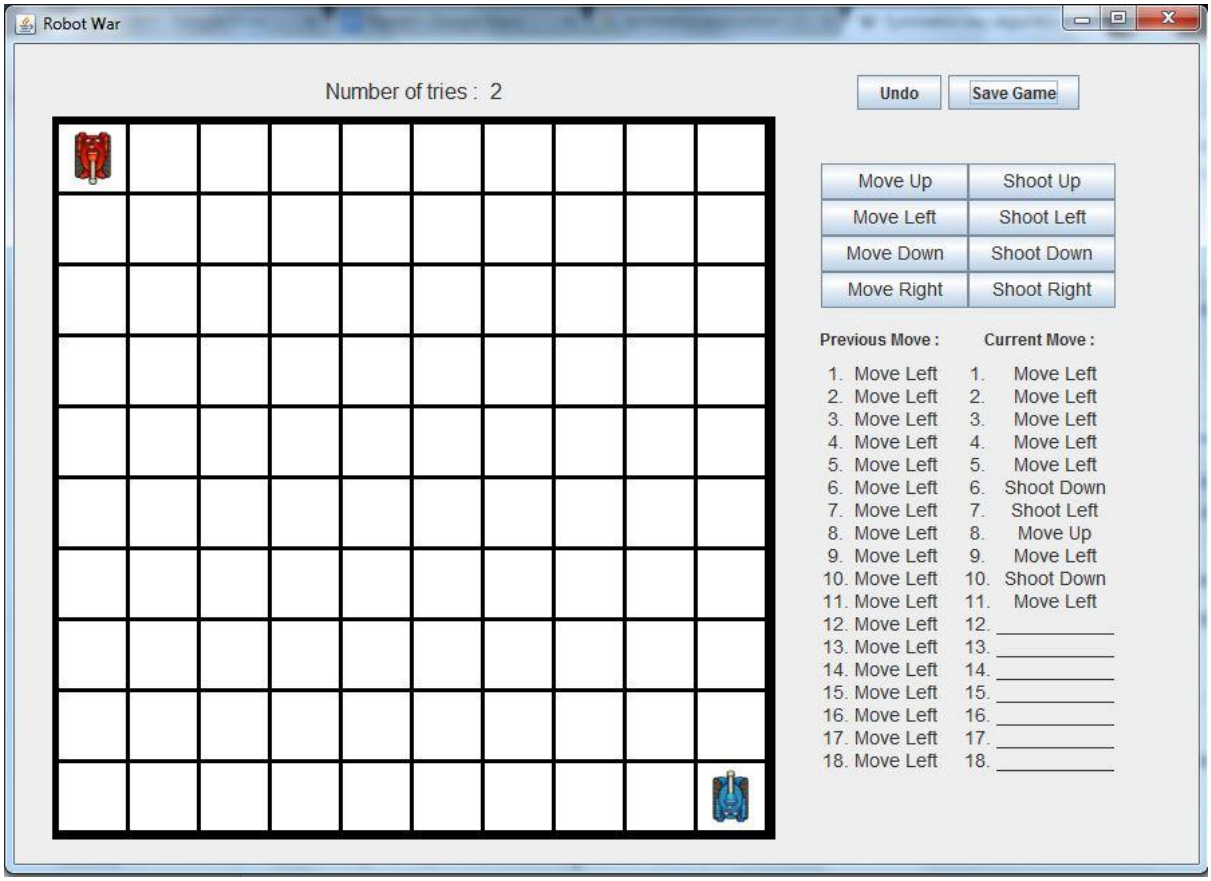
7. If you still failed after 10 tries a game over message is displayed.



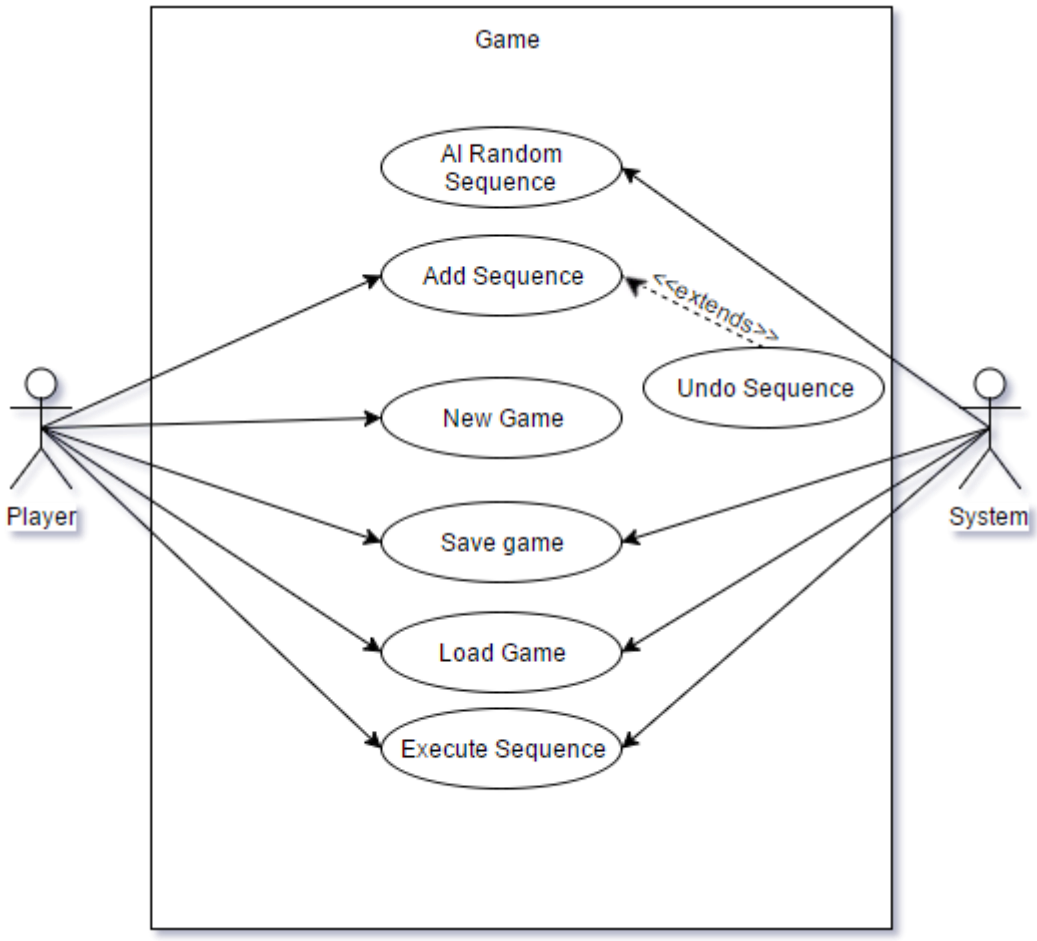
8. After winning or after 10 failed tries an option dialog pops up.



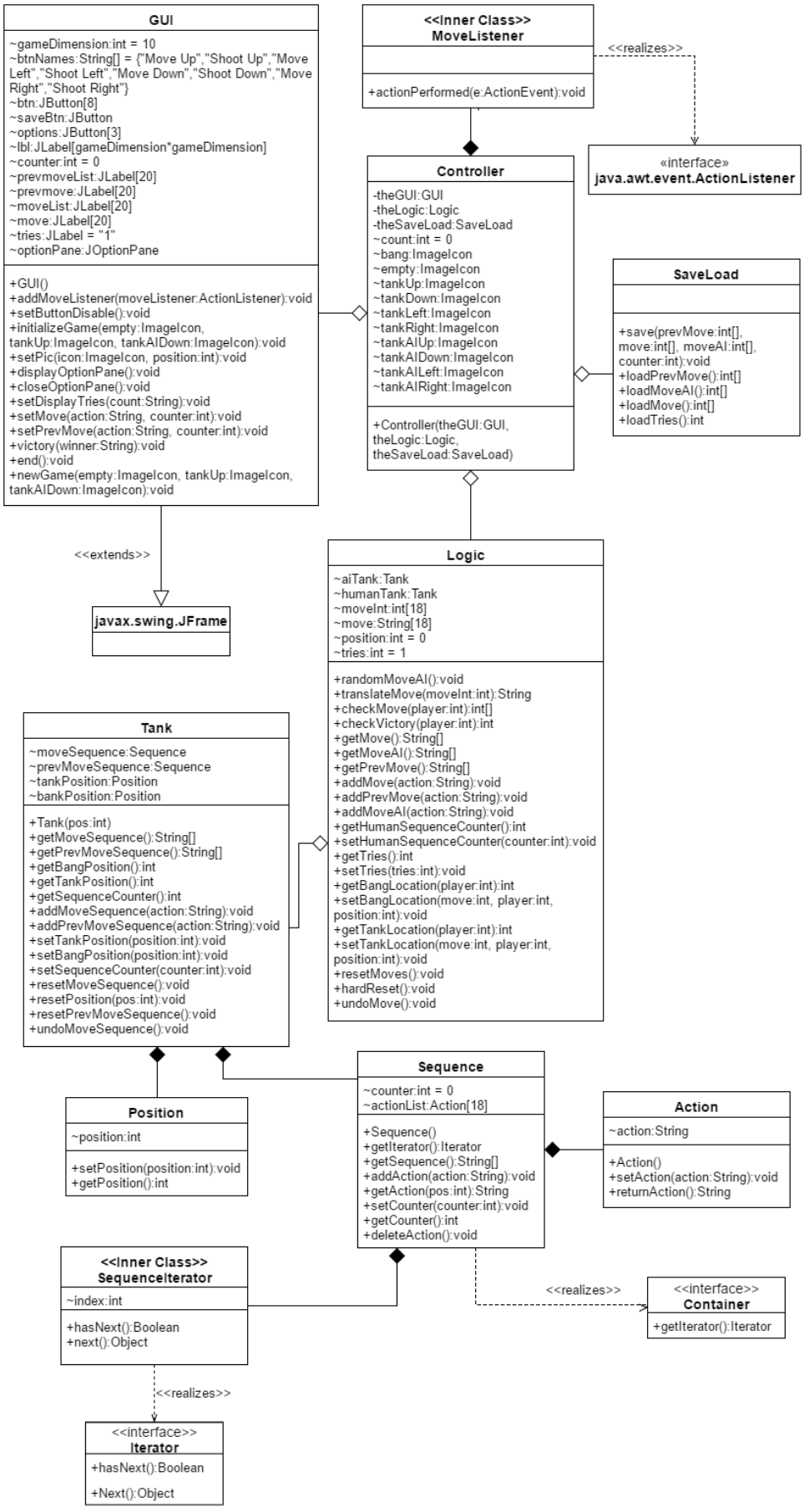
9. If you select save game, the program will override the current load game with the new save.



Use Case Diagram

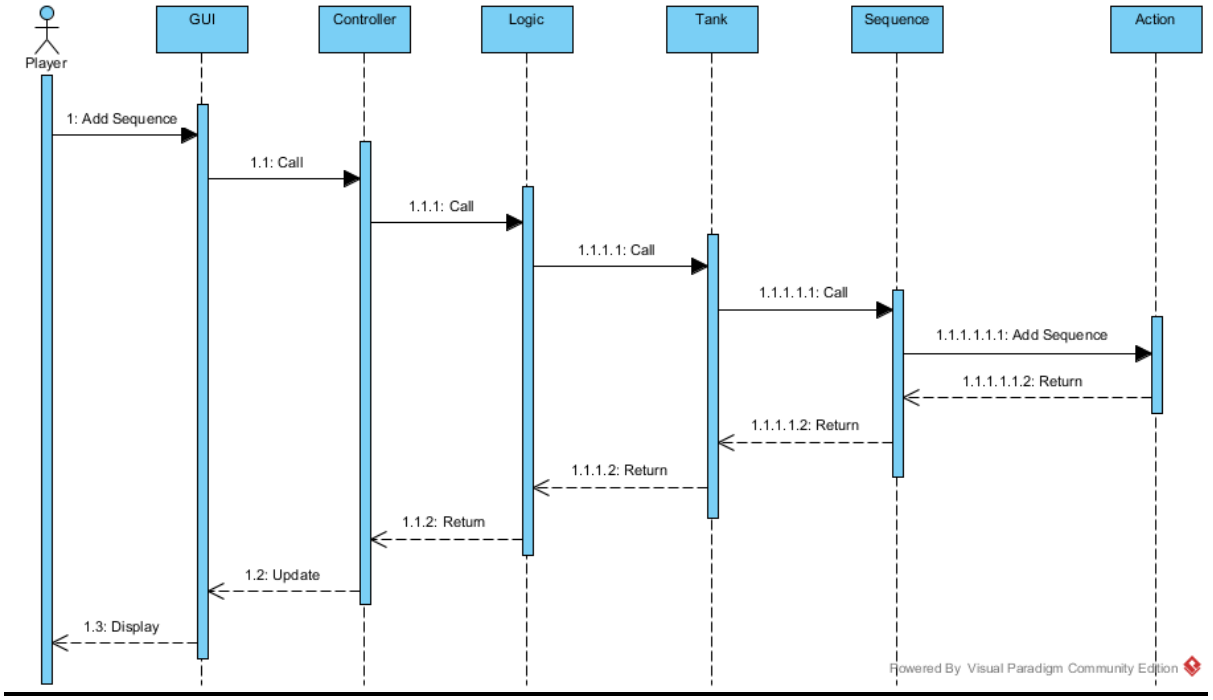


Class Diagram

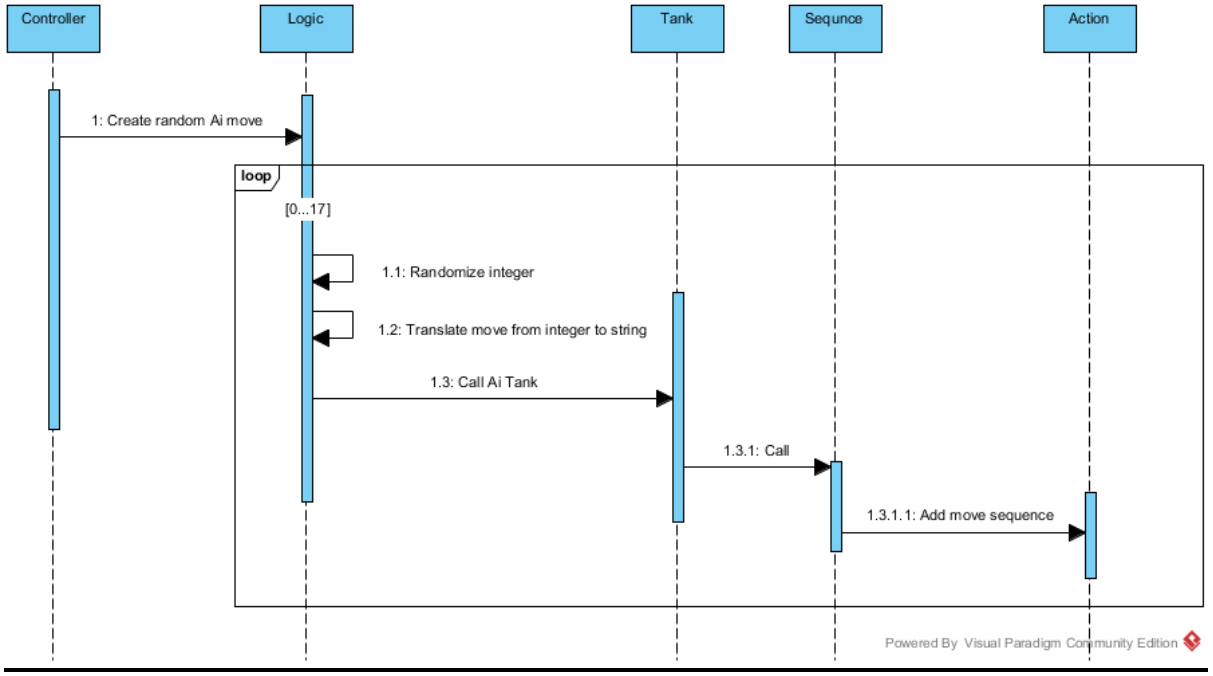

GUI
~gameDimension:int = 10
~btnNames:String[] = {"Move Up","Shoot Up","Move Left","Shoot Left","Move Down","Shoot Down","Move Right","Shoot Right"}
~btn:JButton[8]
~saveBtn:JButton
~options:JButton[3]
~lbl:JLabel[gameDimension*gameDimension]
~counter:int = 0
~prevmoveList:JLabel[20]
~prevmove:JLabel[20]
~moveList:JLabel[20]
~move:JLabel[20]
~tries:JLabel = "1"
~optionPane:JOptionPane
+GUI()
+addMoveListener(moveListener:ActionListener):void
+setButtonDisable():void
+initializeGame(empty:ImageIcon, tankUp:ImageIcon, tankAIDown:ImageIcon):void
+setPic(icon:ImageIcon, position:int):void
+displayOptionPane():void
+closeOptionPane():void
+setDisplayTries(count:String):void
+setMove(action:String, counter:int):void
+setPrevMove(action:String, counter:int):void
+victory(winner:String):void
+end():void
+newGame(empty:ImageIcon, tankUp:ImageIcon, tankAIDown:ImageIcon):void
<<Inner Class>>
MoveListener
+actionPerformed(e:ActionEvent):void
<<realizes>>
«interface»
java.awt.event.ActionListener
Controller
-theGUI:GUI
-theLogic:Logic
-theSaveLoad:SaveLoad
~count:int = 0
~bang:ImageIcon
~empty:ImageIcon
~tankUp:ImageIcon
~tankDown:ImageIcon
~tankLeft:ImageIcon
~tankRight:ImageIcon
~tankAIUp:ImageIcon
~tankAIDown:ImageIcon
~tankAILeft:ImageIcon
~tankAIRight:ImageIcon
+Controller(theGUI:GUI, theLogic:Logic, theSaveLoad:SaveLoad)
SaveLoad
+save(prevMove:int[], move:int[], moveAI:int[], counter:int):void
+loadPrevMove():int[]
+loadMoveAI():int[]
+loadMove():int[]
+loadTries():int
<<extends>>
javax.swing.JFrame
Logic
~aiTank:Tank
~humanTank:Tank
~moveInt:int[18]
~move:String[18]
~position:int = 0
~tries:int = 1
+randomMoveAI():void
+translateMove(moveInt:int):String
+checkMove(player:int):int[]
+checkVictory(player:int):int
+getMove():String[]
+getMoveAI():String[]
+getPrevMove():String[]
+addMove(action:String):void
+addPrevMove(action:String):void
+addMoveAI(action:String):void
+getHumanSequenceCounter():int
+setHumanSequenceCounter(counter:int):void
+getTries():int
+setTries(tries:int):void
+getBangLocation(player:int):int
+setBangLocation(move:int, player:int, position:int):void
+getTankLocation(player:int):int
+setTankLocation(move:int, player:int, position:int):void
+resetMoves():void
+hardReset():void
+undoMove():void
Tank
~moveSequence:Sequence
~prevMoveSequence:Sequence
~tankPosition:Position
~bankPosition:Position
+Tank(pos:int)
+getMoveSequence():String[]
+getPrevMoveSequence():String[]
+getBangPosition():int
+getTankPosition():int
+getSequenceCounter():int
+addMoveSequence(action:String):void
+addPrevMoveSequence(action:String):void
+setTankPosition(position:int):void
+setBangPosition(position:int):void
+setSequenceCounter(counter:int):void
+resetMoveSequence():void
+resetPosition(pos:int):void
+resetPrevMoveSequence():void
+undoMoveSequence():void
Position
~position:int
+setPosition(position:int):void
+getPosition():int
Sequence
~counter:int = 0
~actionList:Action[18]
+Sequence()
+getIterator():Iterator
+getSequence():String[]
+addAction(action:String):void
+getAction(pos:int):String
+setCounter(counter:int):void
+getCounter():int
+deleteAction():void
Action
~action:String
+Action()
+setAction(action:String):void
+returnAction():String
<<Inner Class>>
SequenceIterator
~index:int
+hasNext():Boolean
+next():Object
<<realizes>>
<<interface>>
Container
+getIterator():Iterator
<<realizes>>
<<interface>>
Iterator
+hasNext():Boolean
+Next():Object

Sequence Diagram

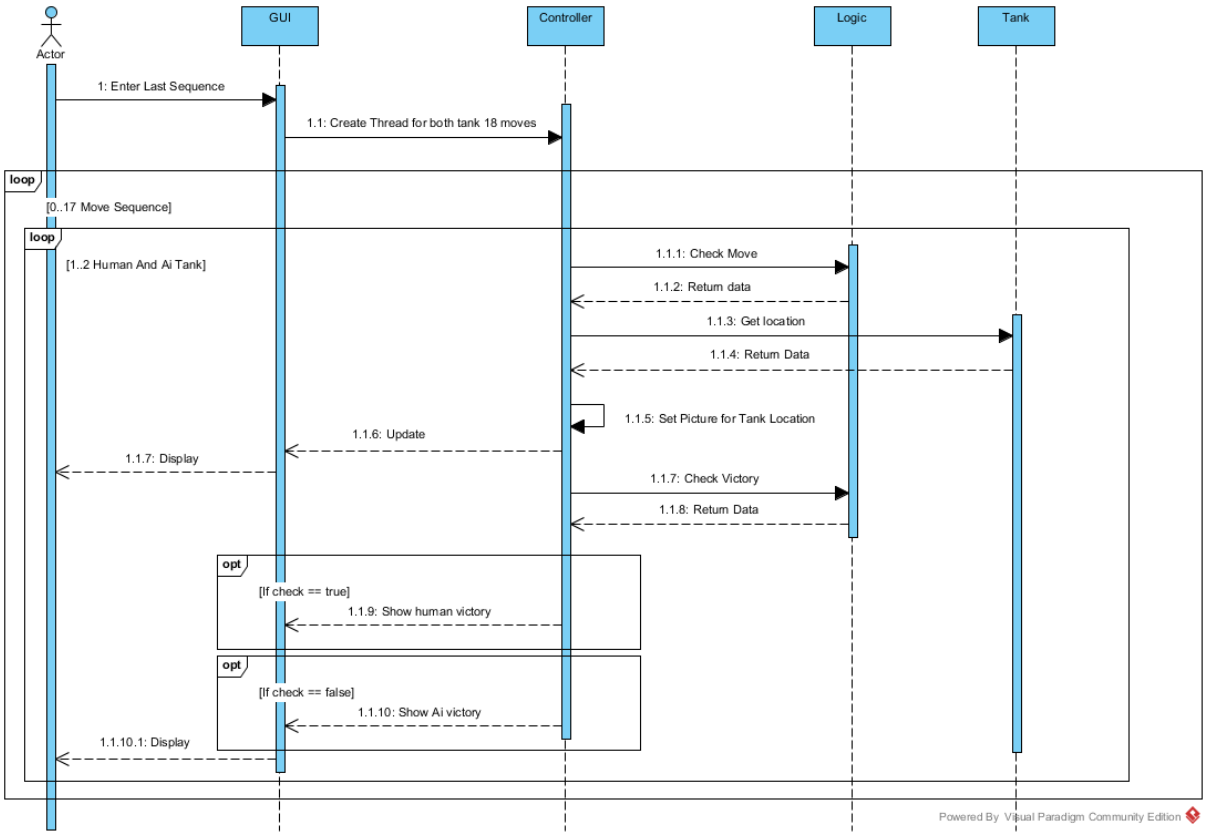
Add Sequence



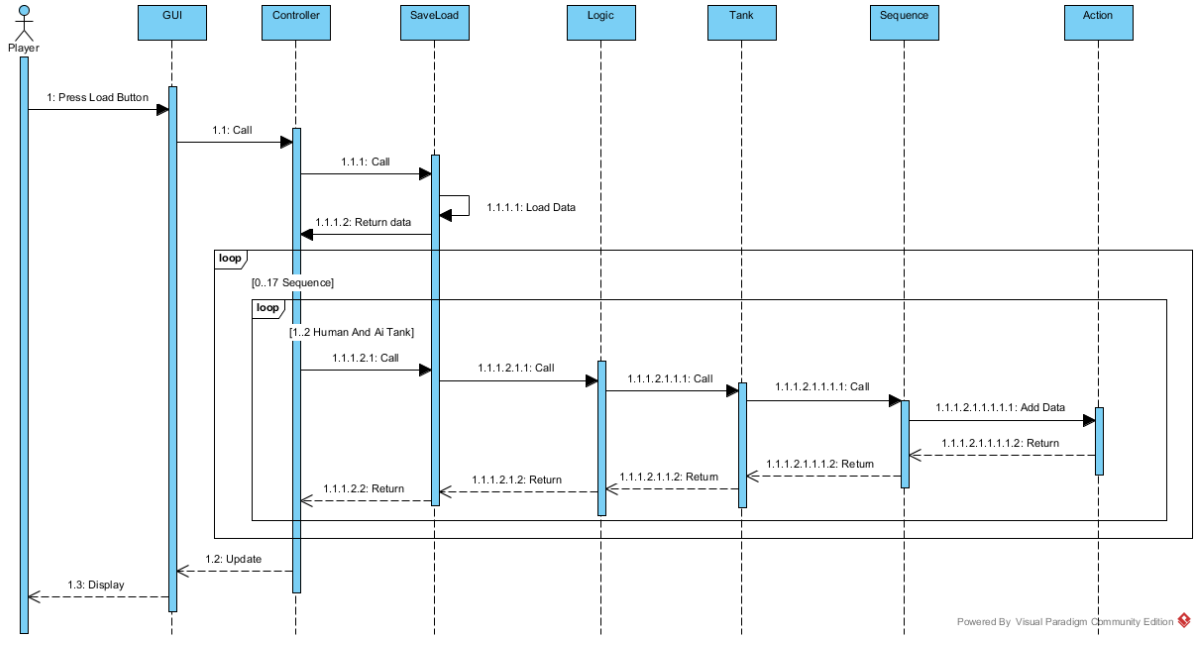
Ai Random Sequence



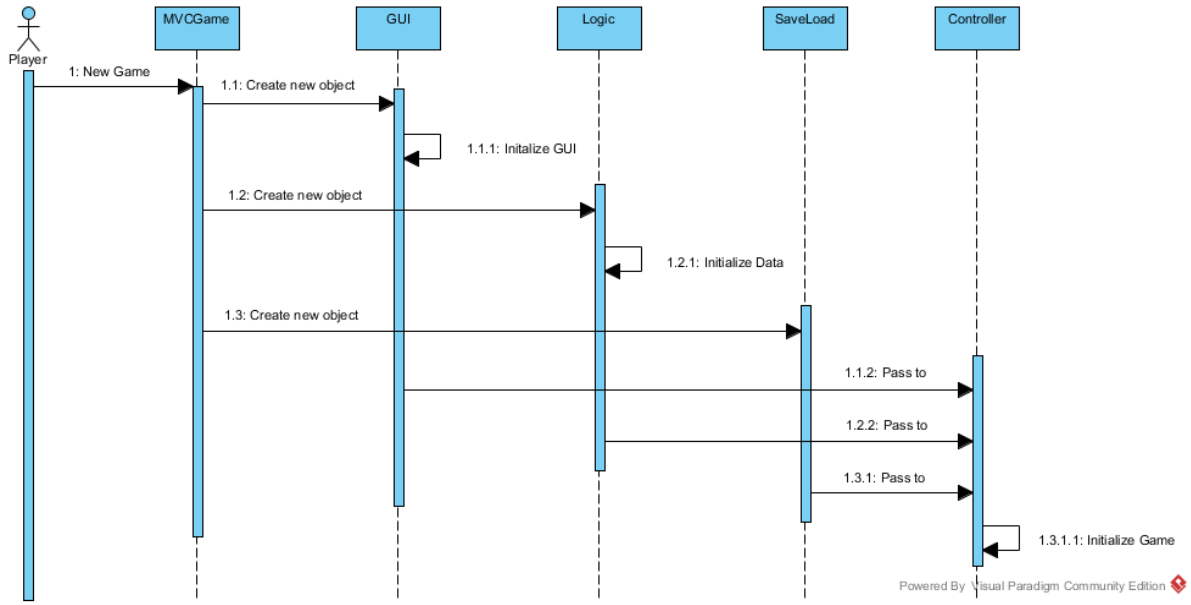
Execute Sequence



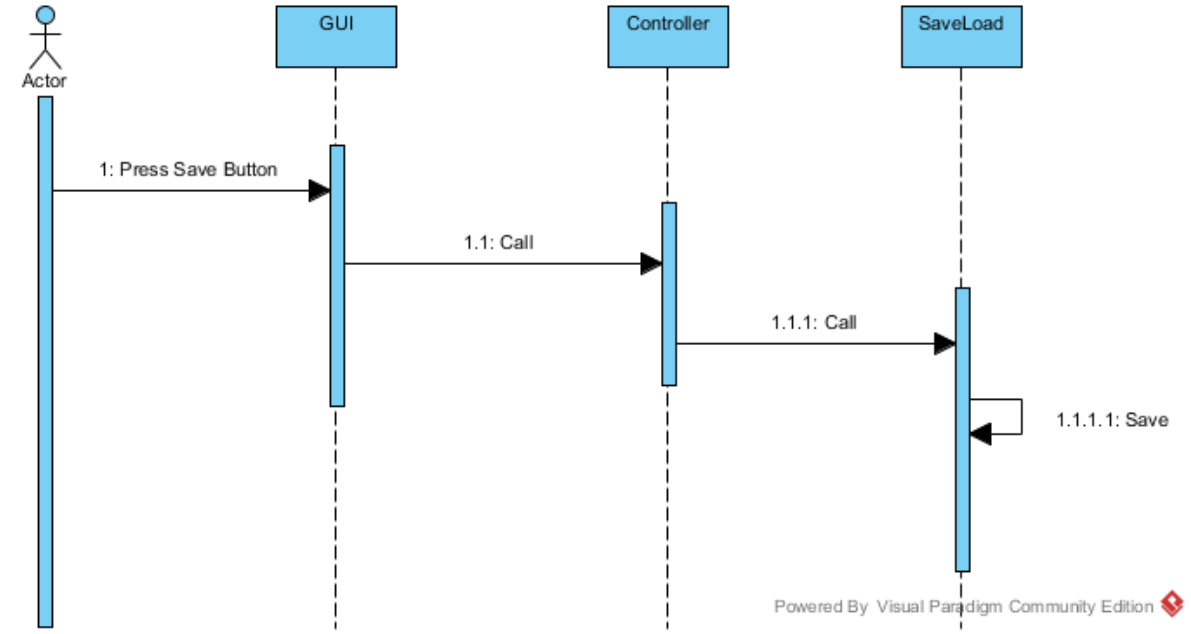
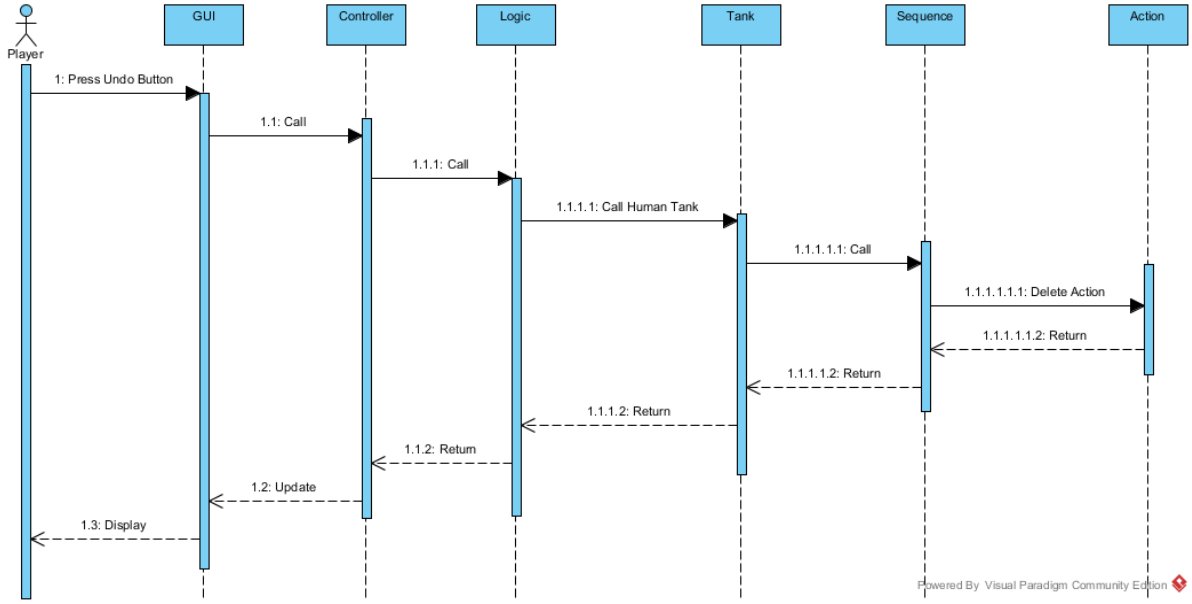
Load



New Game



Save



Undo Sequence