



CS 319 - Object-Oriented Software Engineering

Final Report

Ping the Risk

GROUP-3J

Denizhan Kemeröz

Ahmet Ayberk Yılmaz

Süleyman Semih Demir

Mustafa Tuna Acar

Table of Contents

1. Introduction

1.1 Implementation Process

1.2 Current State

2. Lessons Learnt

3. User Manual

3.1 What is Ping The Risk

3.2 How to Start Playing

3.2.1 Start Condition of a Game

3.3 Distributing the Map, Hackers and Colors to Players

3.4 Game Stages

3.4.1 Hiring Stage

3.4.2 Hacking Stage

3.4.3 Fortifying Stage

3.5 Battle System

3.6 Cards

3.7 Credits

3.8 How to Quickly Learn to Play

4. Build Instructions

4.1 System Requirements

4.2 Installation How-To

5. Who Did What

1.Introduction

1.1 Implementation Process

We have separated the duties with our teammates. Two of us started to search and implement the web server and JavaFx, and the other two started to implement class templates, and wrote some functions that are not related to the javafx or server usage. After the separate parts are searched and done, the whole group has started to combine the parts and help each other to debug and solve some problems that had not been solved in the separated coding phase. We used IntelliJ and connected to our main depository named PingTheRisk that is created by Süleyman. We used GitHub to share the changes and merge the different classes/functions.

As a group we decided to make an online game and used client server architecture and model view. However, throughout the implementation process we have encountered some problems related to the group's people's conditions. That is why our implementation process was slower than planned. None of the group members had any experience with online game implementation, so lots of research has been done before and during the implementation process. There were lots of problems like using a timer to upload the server all the time. Even though lots of research has been done for finding the timer problem and starting the game for all users, the limitation of time and slow process made us change our path. To be able to state that a "game" is made, our group decided to implement the game offline. Even though it was a risky decision because of the limited time, we have chosen that because we were already doing the other gameplay related classes. That is why we could make an offline game by just changing the server related codes, and make them based on offline gameplay. To enable that Süleyman created a new depository named PingTheRiskOffline.

We also had problems playing and changing the background music. We used AudioPlay but because it was referred to in lots of forums and tutorials. However, it had lots of problems, and because of its lack of components it was hard to write. Then we have chosen to use MediaPlayer, but it worked for

one of our teammates and nooneeels, so we did not put that feature to our game.

1.2 Current State

We gave up on the online plan and tried to make an offline playable game. Every group member has written most of their codes. The written codes are combined. There are some absent visuals like dice, or player pictures. Game is playable for 2-4 people. Settings can change the volume of the background music and display screen in the online part, but in offline plans there are no settings. For the online, the settings, credits, how to play, lobby parts are working. For the offline the parts that are hire, hack and fortify are working. The manager classes are done except there can be some problems with out of border gameplay. Our data is done, all visuals are prepared but not all of them implemented in code. All reports are submitted. Background music did not work on some IDE's like 2020 IntelliJ so we did not put it.

2. Lessons Learnt

Using the drawn diagrams we were able to plan the game before writing the code of our games. Using the Use case diagram we were able to have a basic structure of how the gameplay should be. The sequence diagram helped us understand the important methods and parameters for our game to function and that we should focus on these methods and parameters. The action diagram lets us apply the lifecycle of the game more easily since the diagram displays easily how the game should start, live and die. The class diagram helped us divide classes into subsystems and packages and then apply design patterns to them.

The design patterns helped us organise the classes for more efficient solutions. We learned to apply a strategy design pattern which made it possible to add different point gain behaviours in a single method which is add points. This made it possible to add or modify the algorithms to the existing code without making code smell. We also learned to add state design patterns because it was suitable to add and made the code easier.

While trying to understand how to implement the game with online multiplayer, we initially thought that using only the database to manage data flow was going to be sufficient enough for the game to be playable. However after experiencing how long the game takes for the data to be transferred we decided to use amazon web services as our server to run classes for the data transfer between the clients and the database.

3. User Manual

3.1 What is Ping the Risk?

Ping the Risk is a turn based local multiplayer cyber warfare simulator which will be played on desktop which is expected to be an entertaining game. Aim of the players, which are called hackers, is to attack the other countries to conquer the world.

3.2 How to Start Playing?

After installing the game and starting executing the game on your computer, the user will encounter an entrance screen which has the “New Game”, “Options”, “How To Play” and “Credits” selections that the user can choose. In order to start playing, the user should enter names into the empty labels in the start menu and then press “Start Game” to initialize the game.



3.2.1 Start Condition of a Game

After entering 2 to 4 player names in the start page, the player can start the game. The game will start in a new scene with evenly distributed resources for every player.

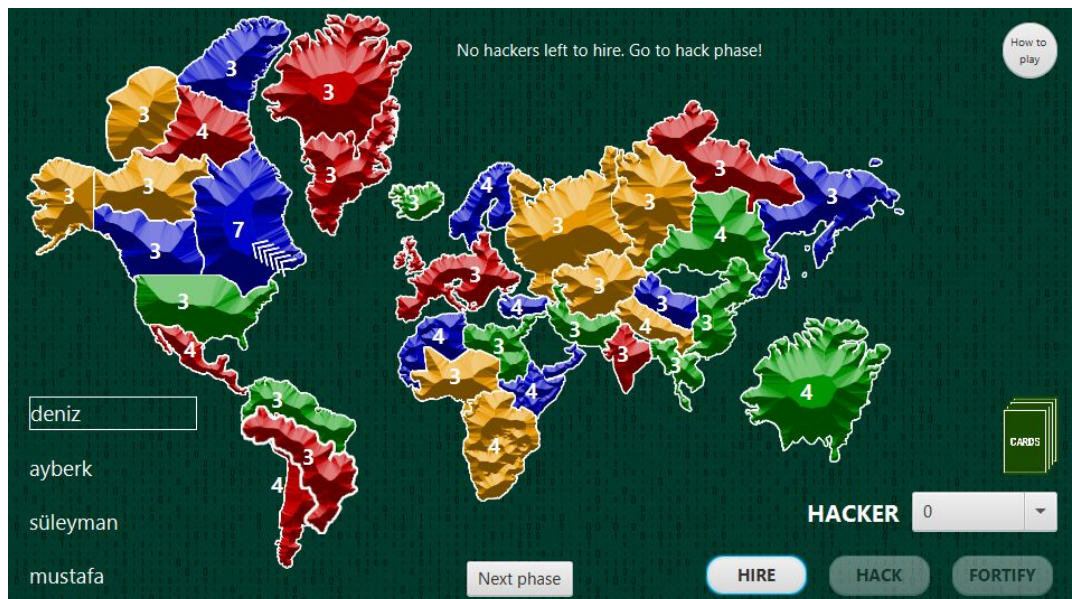
3.3 Distributing the Map, Hackers and Colors to Players



The map of 36 countries will be divided into the number of players which means each of the players will have 9, 12 or 16 countries at the beginning of the game. Each player will have 30 hackers distributed randomly to their own countries which are shown by their assigned colours.

3.4 Game Stages

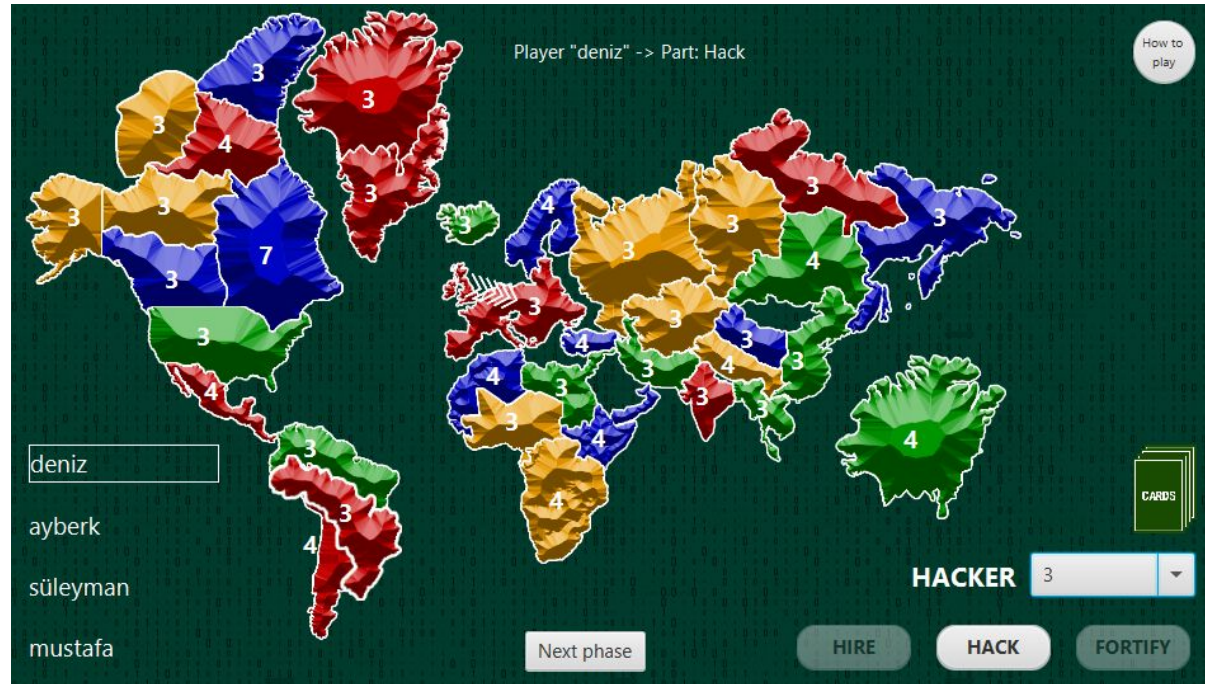
3.4.1 Hiring Stage



Using the cards which are given by the game if a battle is won, or the free 3 hackers that are given to the player every time at the start of their turn,

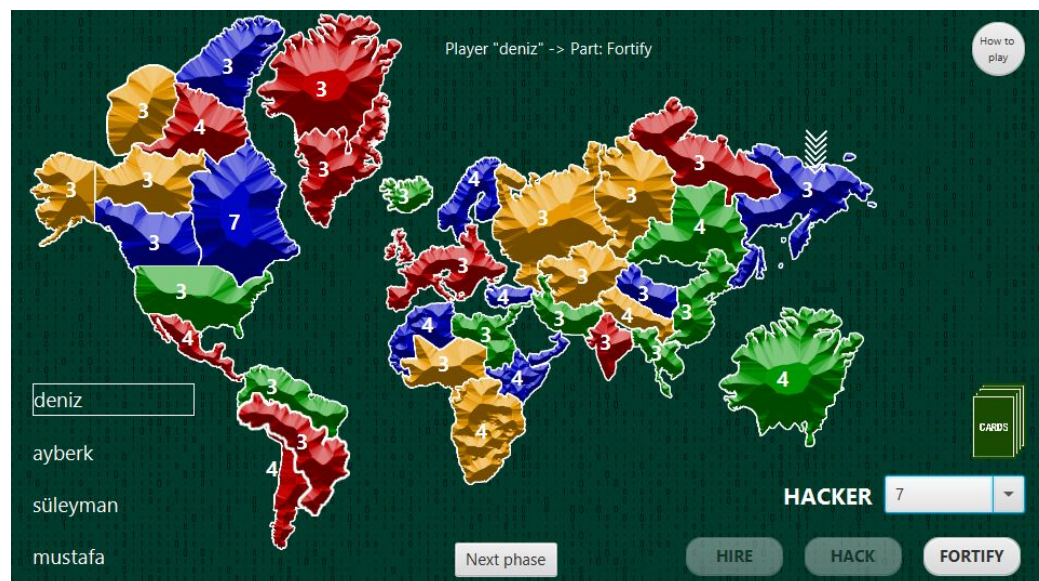
the player should choose 1 of his/her countries to allocate more hackers into there and increase the hackers in that country.

3.4.2 Hacking Stage



The player can choose 1 owned country to attack from and 1 enemy country to attack to. Then the number of hackers that will be used in the attack needs to be chosen and pressing hack will initiate a battle which the player can choose to continue if the battle is not going well for the attacker or keep attacking until the battle is finished and the player won/got defeated.

3.4.3 Fortifying Stage

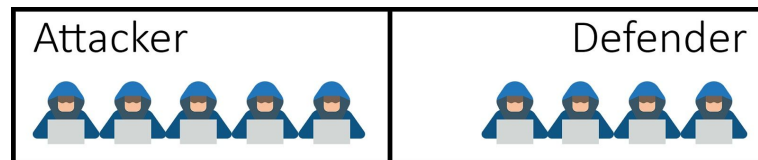


The player can distribute their hackers from one of their countries to another in order to protect that country if it is attacked in another turn. This is done by clicking two of their countries and selecting the amount of hackers

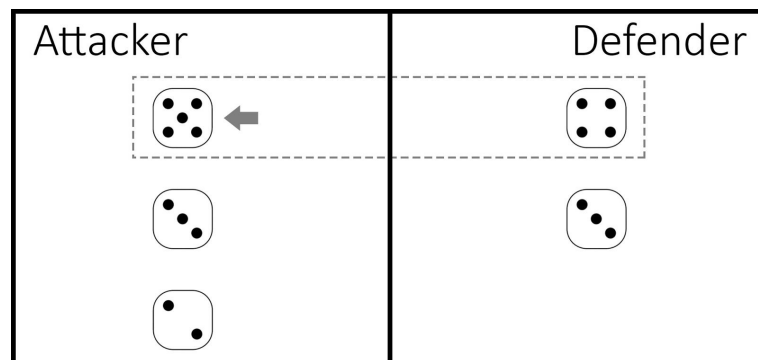
3.5 Battle System

This example scenario explains how many time dice are rolled until one side wins the battle.

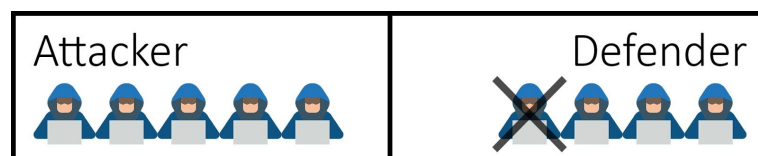
- Let's say there are 5 hackers of the attacker and 4 of the defender.








- The attacker starts hacking with 3 hackers and so rolls 3 dice. The defender rolls 2 dice. The biggest numbers on the dice of each side are compared one by one and the result is in favour of the attacker.





- Thus, the defender loses 1 hacker and there 3 hackers left.








- The second biggest dice are also compared and they are equal to each other. Since the equality is considered as an advantage for the defender side, the attacker loses 1 hacker and there are 4 hackers left.

Attacker	Defender
	
	
	





Attacker	Defender
	

- Again, the attacker and defender roll 3 and 2 dice respectively. This time, the number of the dice of the defender is bigger and so the attacker loses one more hacker. Now he/she will continue hacking with only 3 hackers.

Attacker	Defender
<div>  </div> <div>  </div> <div>  </div>	<div>  </div> <div>  </div>





Attacker	Defender
	

- From now on, the attacker can roll only 2 dice and continue hacking with only 2 hackers because at least one hacker has to be kept out of the war in order to ensure that even if all attacker hackers are lost, the country is not lost.
- In the next rolling stage, both sides are rolled 2 dice. The number on the attacker's highest dice is greater than the number on the defender's highest dice. Now, there are 3 hackers left for attacker and 2 for defender.

Attacker	Defender
<div>  </div> <div>  </div>	<div>  </div> <div>  </div>




Attacker	Defender
	

- The second biggest dice are also compared and the attacker's number is greater than the defender's number, again. Thus, there is only one hacker left for the defender.

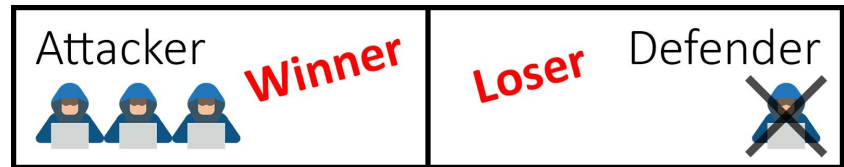
Attacker	Defender
	
 ←	

Attacker	Defender
	

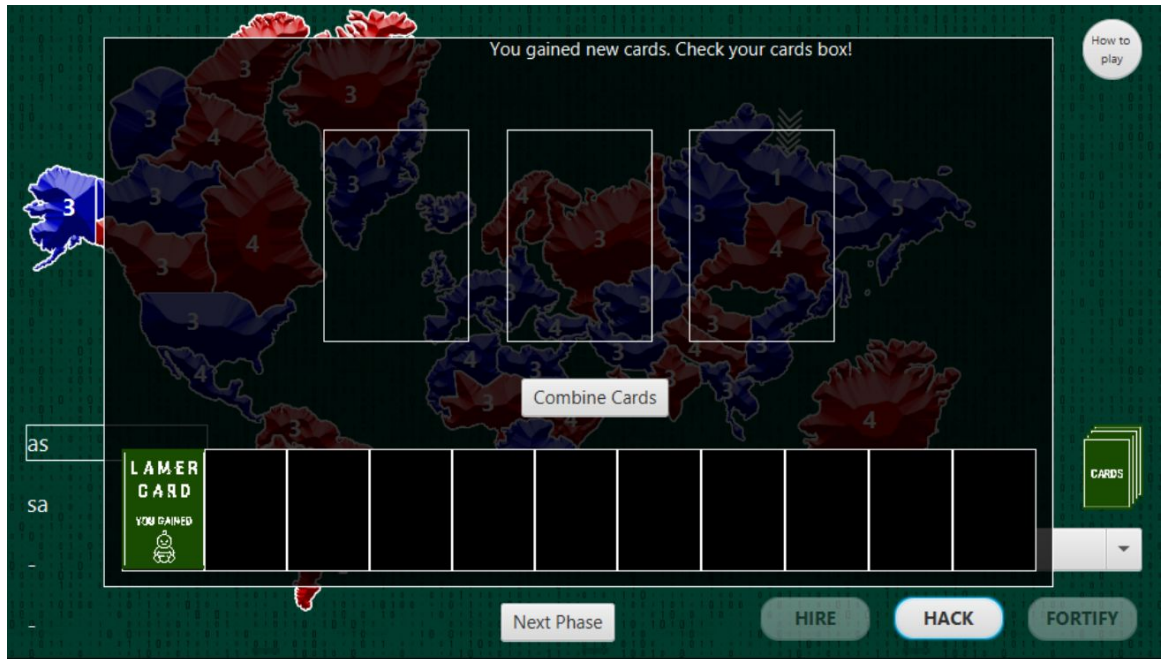
- From now on, the defender cannot roll 2 dice because there is only 1 hacker left and if the player loses this hacker, then the country will be lost.
- In the next rolling stage, the attacker rolls 2 dice and defender rolls 1 dice. Attacker's dice is greater than the defender's dice, so the defender loses his/her last hacker.

Attacker	Defender
 ←	
	

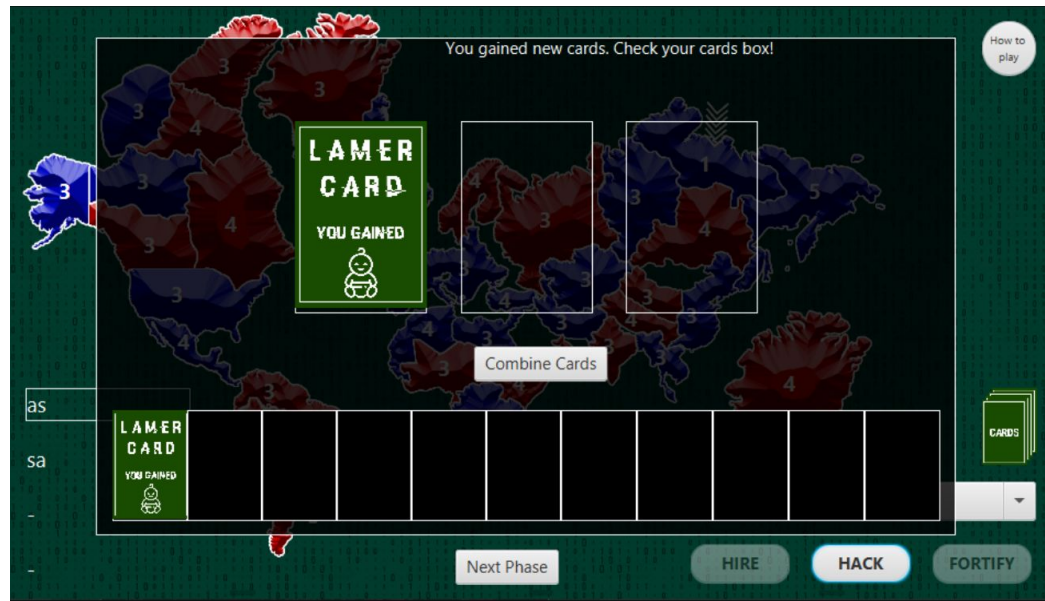
- In this scenario, the winner is the attacker.



3.6 Cards



The game includes 4 types of cards which are Lamer, Black hat, Grey hat and Joker. Cards are gained if a battle is won and a country is conquered. The user can use the cards at the beginning of their turn at the hire part. Lamer card gives 1 hacker, Black hat gives 2 hackers, Grey hat gives 3 and Joker card gives 4 hackers when used and consumed.



3.7 Credits

Players can see the credits from credits.

3.8 How to Quickly Learn Play

Players can get information on how to play the game from pressing the “How to play” button.

4. Build Instructions

4.1. System Requirements

The game works on any operating system that can support and has the latest JRE (Java Runtime Environment).

The game can run on 32 and 64 bit operating systems.

The game requires 512 MB's of RAM.

4.2. Installation How-To

After going to <https://github.com/iBerq/3J-Risk>, the player should download the PingTheRiskOffline folder. After downloading and running this file the game should be installed and started.

5.0 Who Did What

- Ayberk Yilmaz:
 - Prepared the introduction and UseCase Diagram in Analysis Report

- Prepared the introduction and object design trade-offs part in Design Report
- Prepared some visuals for the game
- Focused on implementing the scene classes
- Helped other project mates while debugging.
- Contributed all the classes in online and offline plan
- Süleyman Semih Demir:
 - Prepared the object and class mode, and user interface parts in Analysis Report
 - Prepared the subsystem decomposition, class interfaces, and improvement summary parts in Design Report
 - Prepared some visuals for the game
 - Provided web server and connected it with the game in online plan
 - Focused on implementing the server related classes/function
 - Helped other project mates while debugging.
 - Contributed all the classes in online and offline plan
- Mustafa Tuna Acar:
 - Drawn the sequence and class diagrams for our game.
 - Added functional requirements to the Analysis Report.
 - Prepared the persistent data management and boundary conditions in the Design Report.
 - Implemented the Hack, Dice and Attack classes which executes the hacker battle between two players.
 - Focused on implementing controller and manager classes with Denizhan
 - Helped other project mates while debugging.
 - Contributed all the classes in an online and offline plan.
- Denizhan Kemeröz:
 - Prepared the class diagram, non-functional requirements, and improvement summary in Analysis Report
 - Prepared the hardware/software mapping, access control and security, and final object design part in Design Report
 - Implemented the background music
 - Implemented the Card class and its types
 - Focused on implementing controller and manager classes with Mustafa
 - Helped other project mates while debugging.
 - Contributed all the classes in online and offline plan
 - Prepared some of the parts of the final report