

CS-319 SOFTWARE REQUIREMENTS SPECIFICATION

Group 1-H Risk

Group Members

Kaan Atakan Array – 21703187

Rüzgar Ayan – 21801984

Emre Derman – 21703508

Kamil Kaan Erkan – 21601166

Cankat Anday Kadim – 21802988

Umut Ada Yürüten – 21802410

	1
1. Introduction	2
2. Overview	2
2.1 Gameplay	2
2.1.1 World Domination Risk	2
2.2 Game World	3
2.3 Cards	3
2.4 Game Modes	4
2.5 Additional Features	4
3. Functional requirements	5
3.1 Multiplayer Game	5
3.2 Time Limits for Turns	5
3.3 Pause and Resume	5
3.4 Settings	5
3.5 Difficulty Levels	5
3.6 Saving Games	6
3.7 How to Play	6
4. Nonfunctional requirements	6
4.1 Simple User Interface	6
4.2 Extensibility	6
4.3 Game Performance	6
5. System models	7
5.1. Use case model	7
5.2. Dynamic models	12
5.2.1 Activity Diagram	12
5.2.2 Sequence Diagrams	13
5.2.2.1 Main Menu Sequence Diagram	13
5.2.2.2 Gameplay Sequence Diagram	14
5.2.3 State Diagram	15
5.2.3.1 State Diagram for Win	15
5.2.3.2 State Diagram for Lose	16
5.3. Object and class model	17
5.3.1 Menus and Screens	17
5.3.2 Core Mechanics	17
5.4. User Interface	18
6. Glossary & references	26

1. Introduction

Risk is a turn based militaristic strategy game. A player can add reinforcements to the battlefield, attack to try to occupy a region and then finally maneuver their troops before ending their turn. The original map of the game is a world map, divided into continents, with each one having several regions. The game has one of the two objectives: completing the given secret mission before the other players or dominating the world by controlling all of the regions of the map. Risk is a game of wits, tactics, planning and foresight.

In many aspects, we will adhere to the original game rules when implementing the game. Our game will have 2 game modes which are "World Domination" and "Secret Mission". We will implement troops, a card system, and dice rules same as the original game. Nevertheless, we would also include an additional map to play on: The Bilkent Campus. Although, the tabletop game is meant to be played as a group, we would also implement A.I. opponents to allow players to play alone. A prominent feature will be the faction system, where players would pick a faction with different starting bonuses to add variety and another layer of planning in our game. We would like to keep the bonuses to the minimum as to not spoil the classical core gameplay. Another feature would like to improve the alliance system where players can negotiate deals that allow them additional options; however, much like the original game, betrayal and treachery is always an option.

2. Overview

2.1 Gameplay

2.1.1 World Domination Risk

Setup and Army Placement

In our implementation of the risk game the army and territory placement will be done automatically not by the players.

Playing

Like in the original game each turn will consist of 3 steps: getting and placing new armies, attacking and fortifying. These 3 stages will be the same as the original board game and the digital adaptation.

Getting and Placing New Armies

This will be similar to the digital and board versions. Each player will get reinforcements at the end of each turn. The reinforcement quantity will be calculated by dividing the number of territories that a player has by 3 also some group of territories will form continents and they will have continental advantages so this will be also added to the reinforcement count, also a number of reinforcements of a player have at the end of a tour cannot be less than 3.

Attacking & Defending

In their turns, after placing their reinforcements, players will be able to attack the territories of other players. The rules of attacking and defending will be the same as the original rule.

Winning

For “World Domination” game mode, the player who is successful to capture all existing territories on the map and owns them wins the game. For “Secret Mission” game mode, the player that is successful to accomplish the mission that mission card gave first will win the game.

2.2 Game World

A game world will consist of territories (regions) and continents which will be implemented as nodes on a graph. The territories which share borders will be adjacent and players will be able to move only between adjacent nodes.

2.3 Cards

Card system will be the same as the board game version. There will be two types of cards: Mission Cards, Territory Cards.

Mission Cards

There will be 12 mission cards and they will be used in the “Secret Mission” game mode to specify the mission of the players.

Territory Cards

The number of territory cards will be equal to the number of territories in the map that are the players currently playing, plus 2 wild cards. Territory cards will be used as a currency and under certain conditions a player can trade them to have additional reinforcement during their turn. Territory cards will be in the both of the game modes , World Domination and Secret Mission, one card will be given to each player at the end of every turn who are successful to capture at least one new territory during that turn. The cards will have either a infantry, cavalry or an artillery symbol on them. Players will be able to trade any 3 cards they have with the following combinations: 3 of the same symbols, 1 of each symbol and any two symbols and a wildcard. After this trade, the cards will return to the deck and the player will be provided with reinforcements. After each card trade of any player in the game the reinforcement provided will be increased. The first six trade will be:

The first set traded in - 4 armies The
second set traded in - 6 armies The
third set traded in - 8 armies The
fourth set traded in - 10 armies The

fifth set traded in - 12 armies The
sixth set traded in - 15 armies

And after six each reinforcement will be increased to 5 soldiers until the end of the game.

2.4 Game Modes

There will be two game modes which are “World Domination” and “Secret Mission”. In the “World Domination” mode the players will compete to own and capture every territory on the map. In “Secret Mission” game mode the players will be provided with secret missions at the beginning of the game and they will try to accomplish their mission first.

2.5 Additional Features

Currently we are planning to add three additional features to the game: Alliance System, Factions and AI.

Alliance System

Currently the alliance system in the original game is considerably basic, so we want to add some additional features to it. Firstly, allies will be able to move the troops that they have on their allies' territories. This will give them the ability to move their troops to a territory they have which they have no other adjacent territory with. Also, allies will be able to supply their allies with the reinforcement they received.

Factions

We want to add some factions to the game which have some additional starting bonuses to diversify the gameplay.

AI

We want to add a simple AI to the game for controlling the bots. Player will be able to choose the difficulty of the AI before beginning the game.

3. Functional requirements

3.1 Multiplayer Game

This game will be playable by 2 to 7 players. Users will be able to choose the number of players when starting a new game. Also, they will be able to choose how many of these will be human players and how many will be computer-controlled players. But at least one of these players must be human.

Multiplayer games will be limited to only local. This is not a problem since this is a turn-based game where each player will be playing only when their turn comes.

3.2 Time Limits for Turns

Each turn in the game will have a time limit so that the player must decide on his/her strategy in a limited time. If the time ends during the attacking or fortifying phases, the turn will simply pass to the next player. However, if the time ends in the placing armies' phase before the player did not place all their armies, then these armies will be placed randomly.

This time limit will be chosen when starting a new game and it can be later changed in the settings screen. Completely removing the time limit will also be an option.

3.3 Pause and Resume

Since there are time limits for turns, it is important to be able to pause the game. Pausing the game will be possible with both a key and a button on the screen. When the game is paused, the user will see a pause screen where they can resume the game, save the game, change the settings, or exit to the main menu.

3.4 Settings

The game will have a settings screen where the users will be able to access from either the main menu or from the pause screen during an ongoing game. Currently, these settings include sound and music levels.

3.5 Difficulty Levels

When the users start a game including computer-controlled players, they will be asked to choose a difficulty level. These difficulty levels will affect the computer-controlled players' decision mechanisms and/or give them more army.

3.6 Saving Games

Players will be able to save the game from the pause menu at any time. Then, they can load any saved game from the main menu instead of starting a new one. This is an important functionality since games can last long sometimes.

3.7 How to Play

There will be a how to play page accessible from the main menu which explains in general the game rules and the UI of the game. The additional features such as factions and alliances that are not in the original game will be explained in more detail.

4. Nonfunctional requirements

4.1 Simple User Interface

The game must be directive for the users so that one with no knowledge of the game can understand and play the game only by reading the how to play page. This requirement will be satisfied by the design and the positioning of the UI components. This requirement can be tested by having people play the game.

4.2 Extensibility

First, the game should support adding new maps and possibly new game modes without causing changes in the core code. And, the AI of the game must be open to improvements so that the game can be more realistic when played with the computer-controlled players.

4.3 Game Performance

Since this is a board game and it does not involve much graphical motion, having high frame rates or low response times will not affect the performance of the players a lot. But again, since the game has basic graphics, it is expected that it must run in 60 frames per seconds and players must get response upon an action in no more than 5 ms to satisfy the players.

5. System models

5.1. Use case model

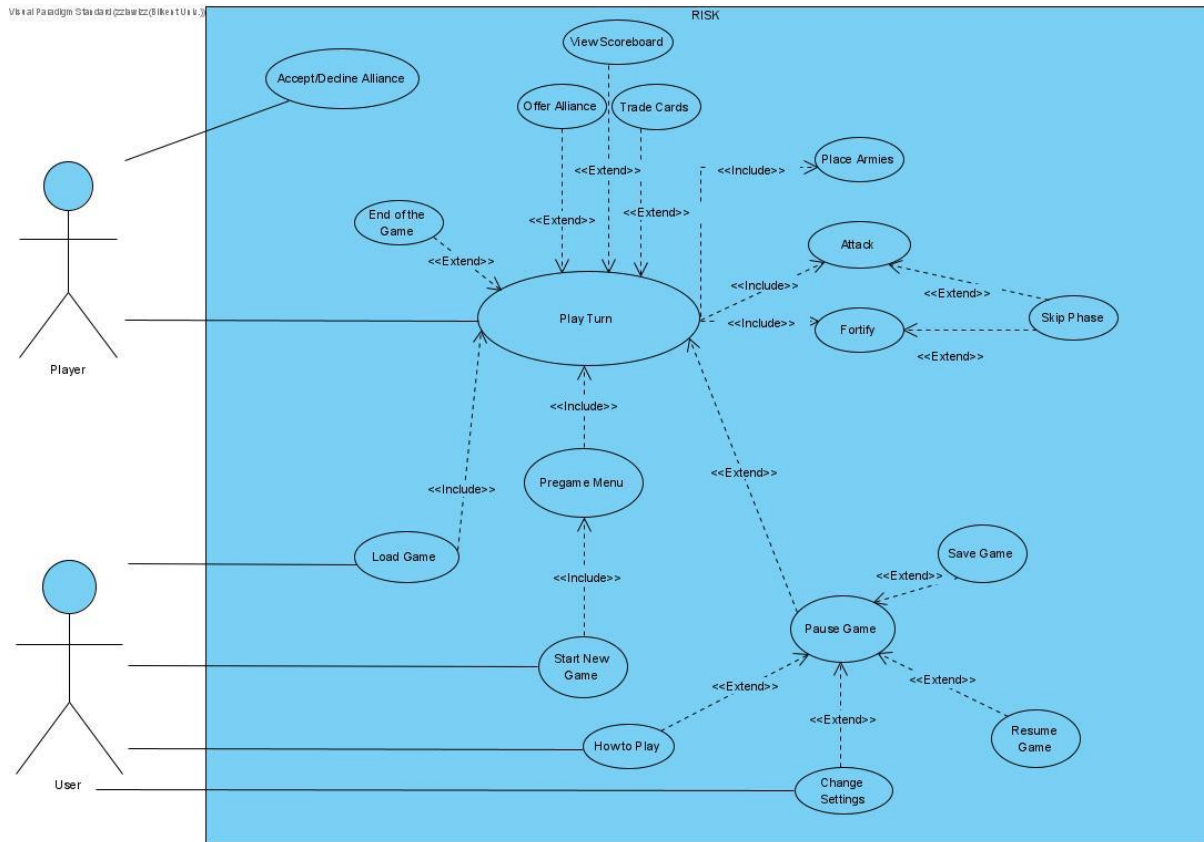


Figure 1: The Use Case Diagram

The Use Case Diagram is a representation of user action and responds on the system in different cases. In our model, the main case and the simplest use case descriptions between the cases has been shown. The explanations for some of the important use cases are given below.

Use Case #1

Use Case: Start New Game

Primary Actor: User

Stakeholders and Interests:

- User wants to start a game.
- User wants to move to the pregame menu case.
- System opens the pregame menu.

Pre-conditions:

- User must be in the main menu.

Post-conditions:

- User is in the Pregame Menu.

Entry-conditions:

- User selects the “New Game” button in the main menu.

Exit conditions:

- No exit conditions.

Success Scenario Event Flow:

1. User selects the “New Game” button in the main menu.
2. System displays the available maps in the main menu.
3. User selects the map.
4. System brings the user to the Pregame Menu.

Use Case #2

Use Case: Pregame Menu

Primary Actor: User

Stakeholders and Interests:

- User wants to configure the game.

Pre-conditions:

- No pre-conditions.

Post-conditions:

- The game is initialized and started.

Entry-conditions:

- User opens the Pregame Menu by starting a new game from the main menu.

Exit conditions:

- User presses ESC on the keyboard.

Success Scenario Event Flow:

1. User comes to the Pregame Menu from the main menu.
2. User selects the game rules including the game mode, AI difficulty and the turn timer.
3. User configures the number of computer-controlled and human players.
4. User selects the start button to start a new game with these configurations.
5. System initializes and starts the game.

Use Case #3

Use Case: How to Play

Primary Actor: User

Stakeholders and Interests:

- User wants to learn how to play the game.

Pre-conditions:

- User must be in the main menu.

Post-conditions:

- No post conditions.

Entry-conditions:

- User selects the “Learn to Conquer” button in the main menu.

Exit conditions:

- Player selects the “Exit to the Main Menu” button on the how to play screen.
- Player presses ESC on the keyboard.

Success Scenario Event Flow:

1. User selects the “Learn to Conquer” button in the main menu.
2. System displays the how the play screen.
3. User presses ESC on the keyboard.
4. System brings the user back to the main menu.

Use Case #4

Use Case: Load game

Primary Actor: User

Stakeholders and Interests:

- User wants to load a game that has been saved before.

Pre-conditions:

- User must be in the main menu.

Post-conditions:

- The loaded game is initialized and started.

Entry-conditions:

- User selects the “Load Game” button in the main menu.

Exit conditions:

- No exit conditions.

Success Scenario Event Flow:

1. User selects the “Load Game” button in the main menu.
2. User chooses the game which is going to load.
3. The system initializes and starts the loaded game.

Use Case #5

Use Case: Change Settings

Primary Actor: User

Stakeholders and Interests:

- User wants to change the settings of the game.

Pre-conditions:

- User must be in the main menu.

Post-conditions:

- Some of the settings of the game have changed.
- User is brought back to the main menu.

Entry-conditions:

- User selects the “Options” button in the main menu.

Exit conditions:

- User selects the “Back” button in the options menu.

Success Scenario Event Flow:

1. User selects the “Options” button in the main menu.
2. System displays the options menu.
3. User changes the volume settings.
4. User selects the “Back” button in the options menu.
5. System brings the User back to the main menu.

Use Case #6

Use Case: Pause Game

Primary Actor: Player

Stakeholders and Interests:

- Player wants to pause the game so that the turn timer will stop.

Pre-conditions:

- The game is currently ongoing.

Post-conditions:

- No post conditions.

Entry-conditions:

- Player selects the pause button on the screen.
- Player presses ESC on the keyboard.

Exit conditions:

- Player clicks the “Resume” or the “Exit to the main menu” buttons on the pause menu.
- Player presses ESC on the keyboard.

Success Scenario Event Flow:

1. Player selects the pause button on screen while the game is ongoing.
2. Turn timer stops and the pause menu appears on the screen.
3. Player selects the “Change Settings” button on the pause menu.
4. After changing the settings, Player comes back to the pause menu.
5. Player selects the “Resume” button on the pause menu.
6. Turn timer starts from where it left, and the game continues.

Alternative Event Flows:

1. Player selects the pause button on screen while the game is ongoing.
2. Turn timer stops and the pause menu appears on the screen.
3. Player selects the “Save Game” button on the pause menu.
4. Player selects the “Exit to the Main Menu” button on the pause menu.
5. Current game ends and the main menu comes to the screen.

Use Case #7

Use Case: Play Turn

Primary Actor: Player

Stakeholders and Interests:

- Player wants to play their turn in the game.

Pre-conditions:

- No pre-conditions.

Post-conditions:

- Game is finished.

Entry-conditions:

- A game is started.

Exit conditions:

- Game finishes.
- Players exit to the main menu from the pause menu.

Success Scenario Event Flow:

1. A new game is initialized and started.
2. Until the game finishes:
 - a. Player plays their turn. This can include:
 - i. Placing armies
 - ii. Attacking
 - iii. Fortifying
 - iv. Ending phases
 - v. Trading cards
 - vi. Showing the scoreboard
 - vii. Pausing the game
 - viii. Offering alliances
 - b. After ending the fortify phase, the turn of the Player ends.
 - c. Next Player does the same things.
3. After the game finishes, the system displays the ending screen.

Use Case #8

Use Case: Accept/Decline Alliance

Primary Actor: Player

Stakeholders and Interests:

- Player wants to answer an alliance offer.

Pre-conditions:

- The game is currently ongoing.

Post-conditions:

- An alliance is formed between two Players if the offer is accepted

Entry-conditions:

- Another Player offers alliance to this Player

Exit conditions:

- No exit conditions.

Success Scenario Event Flow:

1. An alliance offer comes to the Player
2. A window with Accept and Decline buttons appears on the screen
3. Player selects one of the Accept or Decline buttons

the main menu. Most importantly, users can choose “New Game” or “Load Game” buttons in the main menu which initializes a new game.

When the game is initialized, the system waits for an input from the current player until his/her time limit is exceeded. If the time limit is exceeded, the system checks if the game is finished and if it is not finished, the game proceeds to the next player’s turn. If the time limit is not exceeded, the system processes the input from the player as follows.

First, the type of input is checked. Players can either make movements (including attacking) in the game board or they can end their phases. If the player gives a movement input, the system first checks if this is a valid movement or not and then either performs the movement or displays an error message accordingly. If the player gives a phase ending input, the system ends the phase with the exception of checking if all the armies have been placed in the army place phase. Finally, if a player ends the fortify phase, the turn of that player ends, and the system checks if the game is finished or not. Again, if the game is not finished, the game proceeds to the next player’s turn and the system continues to the same process. However, if the game is finished, the system displays the ending screen and asks the user if they want to play another game. At this point, the system either closes the game or brings the user back to the main menu so that they can start a new game.

5.2.2 Sequence Diagrams

5.2.2.1 Main Menu Sequence Diagram

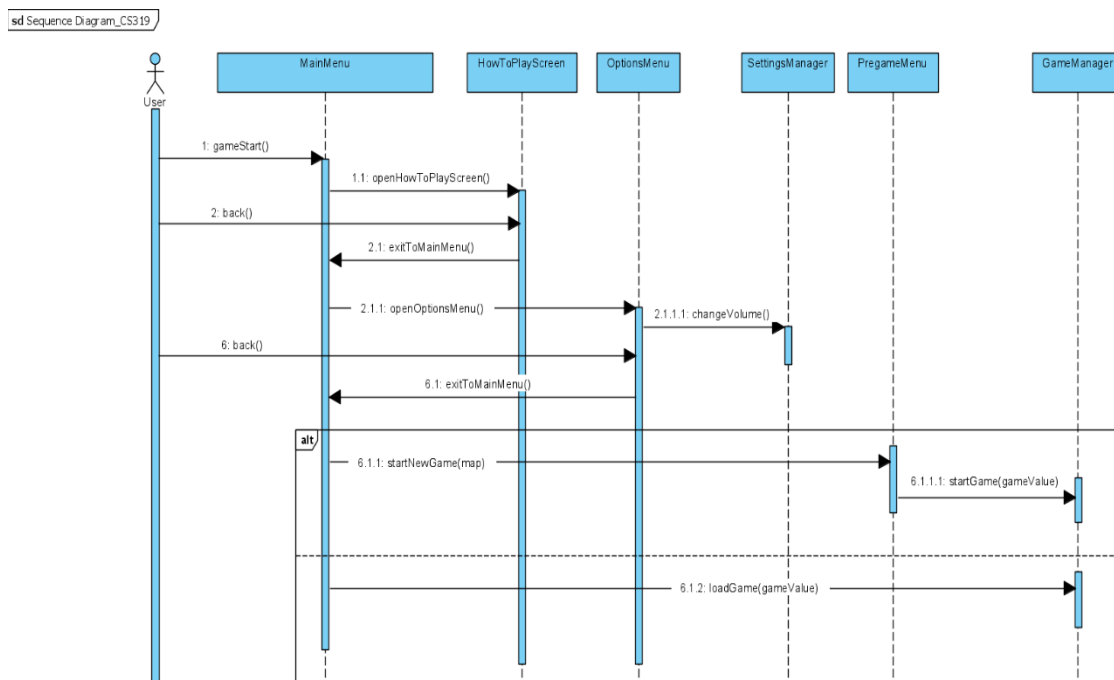


Figure 3: Main Sequence Diagram

The below sequence diagram shows an example interaction of a User with the game. Upon the game starting, User opens the how to play screen from the main menu and then exits back to the main menu. Next, User opens the options menu and changes the volume there which makes a call to the Settings Manager. After exiting to the main menu, User can now either start a new game by selecting a map or load a game.

When starting a new game, User is directed to the Pregame Menu where they can configure the settings for the game whereas the game directly loads into the GameManager.

5.2.2.2 Gameplay Sequence Diagram

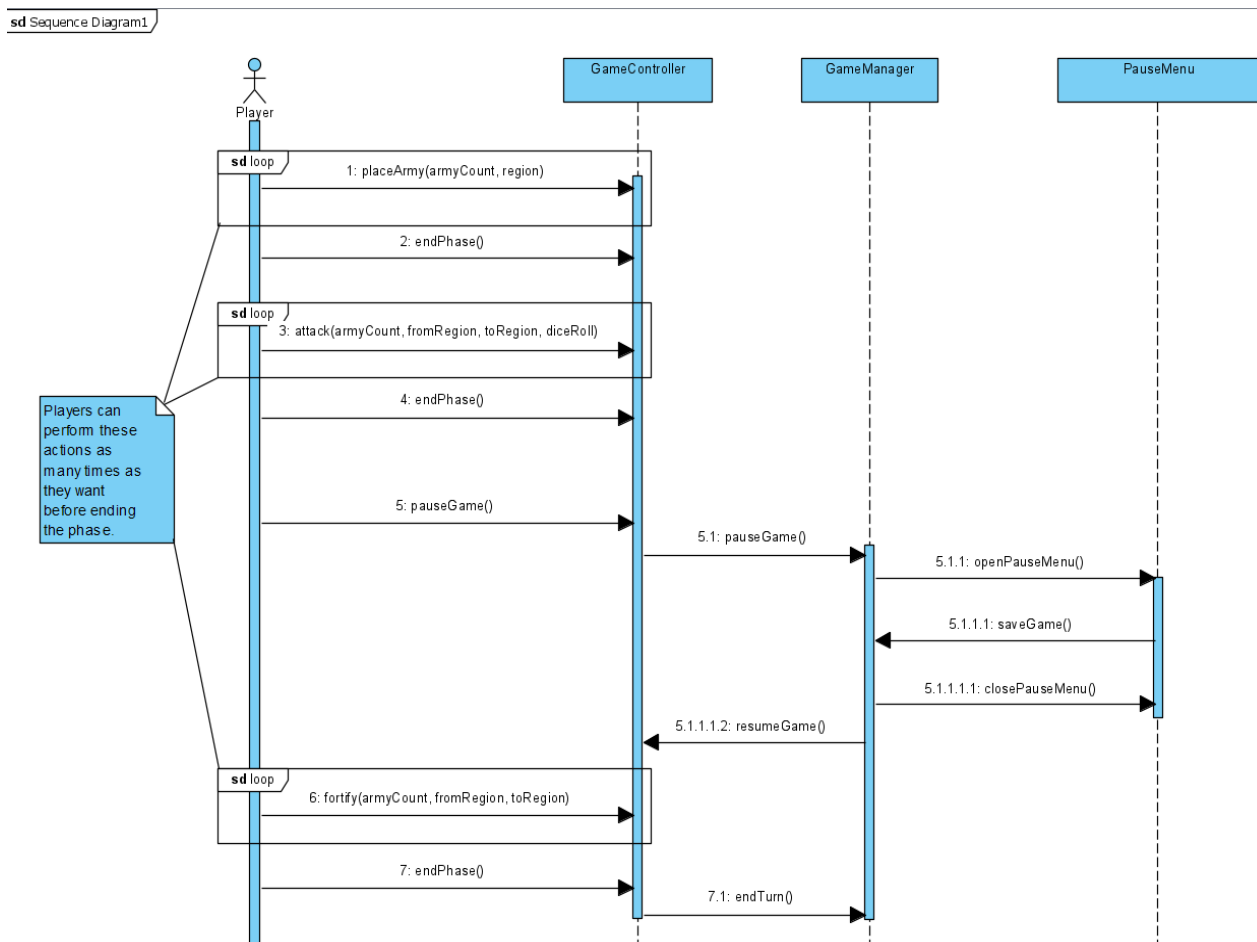


Figure 4: Gameplay Sequence Diagram

The above sequence diagram shows an example series of action of a Player in the game. Player makes `placeArmy()`, `attack()`, `fortify()`, `endPhase()` and `pauseGame()` calls with appropriate parameters to the `GameController` which will then communicate with the `GameManager`. Upon ending the last phase, `GameController` makes an `endTurn()` call to the `GameManager` which proceeds the game to the next turn. Also, making a `pauseGame()` call, lets the player interact with the `PauseMenu` and do actions such as saving the game.

5.2.3 State Diagram

5.2.3.1 State Diagram for Win

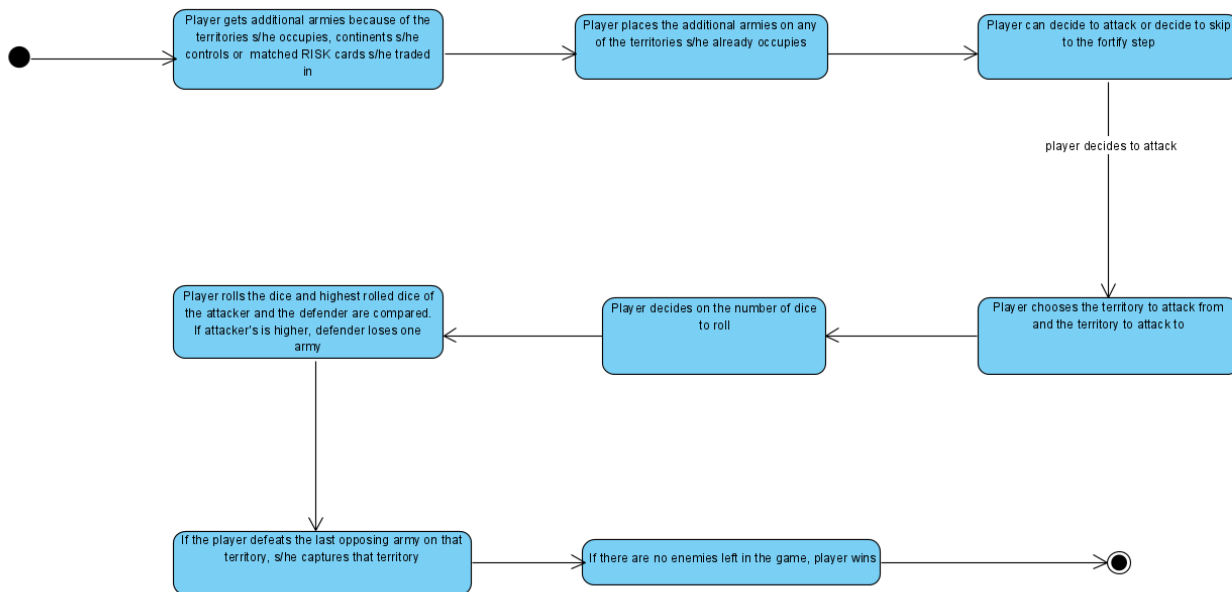


Figure 5: State Diagram for Win

At each turn, the player gets additional armies because of the territories s/he occupies. If it is not the first turn, the player might also get additional armies because of the continents s/he controls or because of the matched cards s/he traded in the previous turn. The player places the additional armies on her/his territory. The player decides to attack and chooses a territory to attack from, which the player already occupies, and a territory to attack to, which is adjacent to the selected territory. Then, the player decides on the number of dice to roll. The player must have at least one more army in her/his territory than the number of dice s/he decides to roll. After the decision, dice are rolled and the highest rolled dice of the attacker and defender are compared. If the attacker's is higher, the defender loses one army. If the defender's is higher or equal, the attacker loses one army. This process is repeated for each attack. At the end, if the attacker defeats the last opposing army on that territory and there are no enemies left in the game, the player wins.

5.2.3.2 State Diagram for Lose

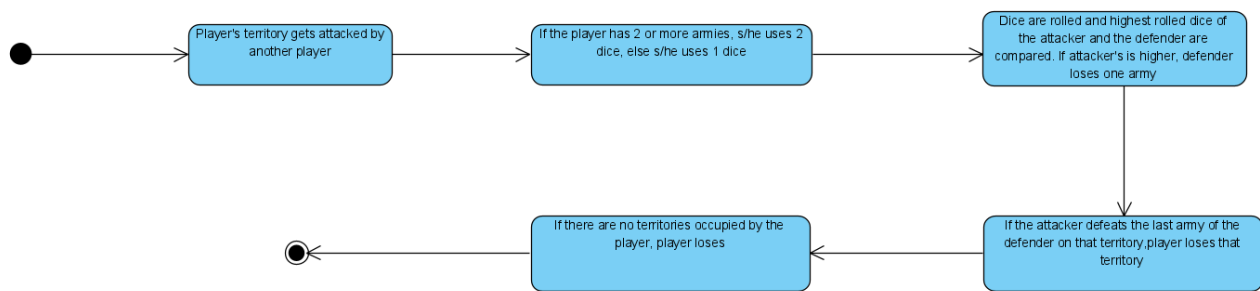


Figure 6: State Diagram for Lose

The player gets attacked by another player. If the player has 2 or more armies s/he uses 2 dice, else s/he uses 1 dice. Then, dice are rolled, and the highest rolled dice of the attacker and defender are compared. If the attacker's is higher, the defender loses one army. If the defender's is higher or equal, the attacker loses one army. If the attacker defeats the last army of the defender on that territory and there are no territories occupied by the player, the player loses.

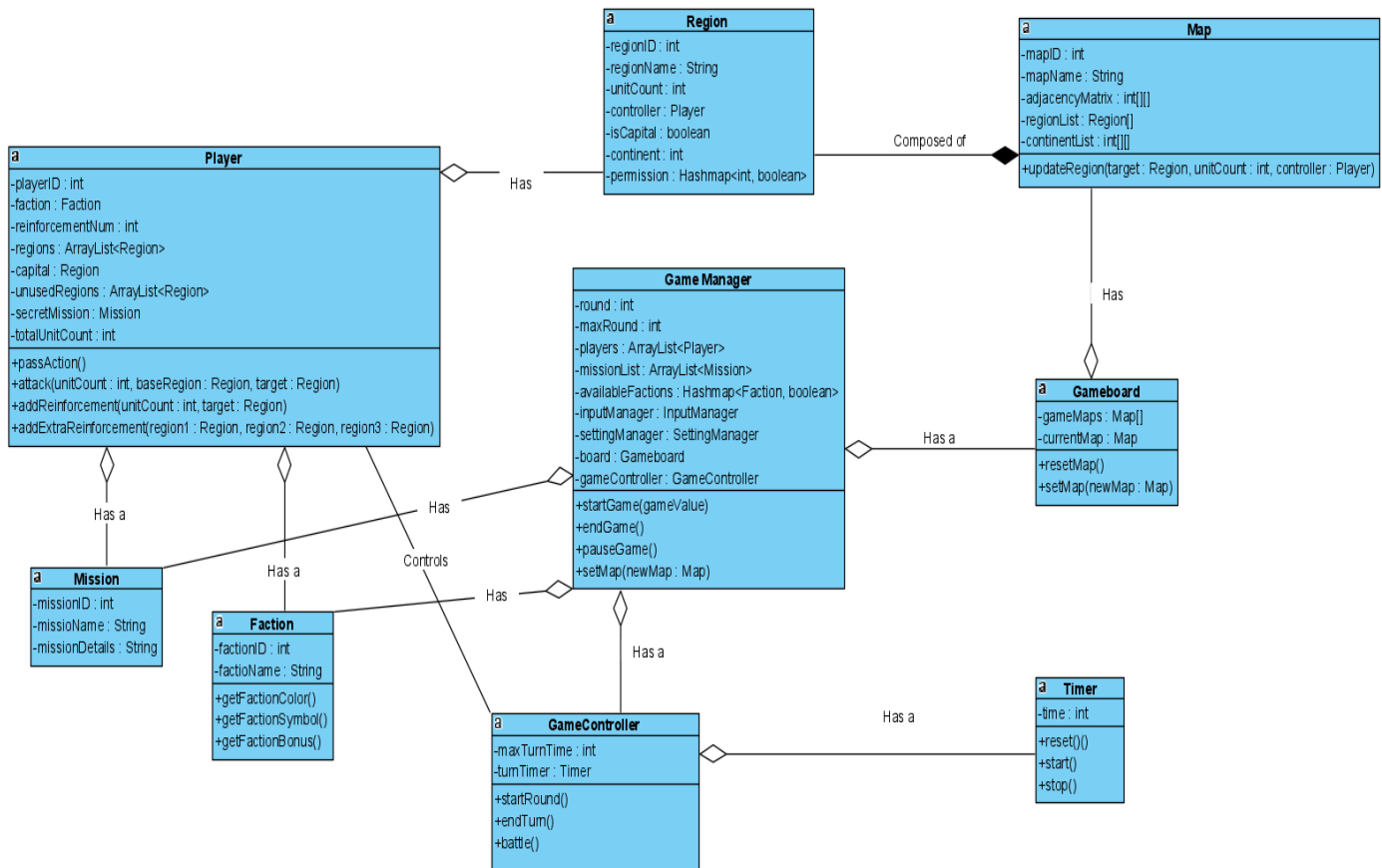


Figure 8: Class diagram of core mechanics

5.4. User Interface

Keeping the aforementioned distinctions between a user and a player we can see how our game's usual flow. If a group of players were to start up our game and the first thing, they see is our main menu screen.



Figure 9: Main Menu Mock-up

The main menu hosts an array of buttons. From bottom to top the “Learn to Conquer” button leads the user to the help menu which hosts the rules and the gameplay of our game. Coming up to the “Options” which holds the sound settings.



Figure 10: Options Menu Mock-up

Coming up to the top two buttons the “New Game” and the “Load Game” options all sequence the user into the startup of the game. If the player chooses the “New Game” option, he will be prompted to choose a map. Our game will have two maps, the first one is the World map that we all expect in a game of RISK, the second one however is the map of our university campus.



Figure 11: Choose Map Screen Mock-up

After choosing either one of the maps or loading his save file the user will be forwarded to the pregame screen where he will be able to see the map of his choosing, the game's settings, such as the “Game Mode”, the “AI Difficulty” and the “Turn Timer”. Apart from that as our game will also host nations with their own pro's and con's we also see the ability to choose the nation and the players color in this screen. The player will move on to the game proper when he is done with his pregame arrangements.



Figure 12: Pregame Menu Mock-up

Fast forwarding a bit we can see a game in action. Our two players with colors, red and blue and two ai counterparts have played the game for a bit now. We can see that it is now the turn of our red player as we can see by the color in his name bar's background. The player is rewarded armies with the rules of the game in mind and the player, if he has the correct lineup of cards, can choose to increase his total reinforcements by trading a set of cards. A legal set of cards after being chosen by the player turn green to indicate such an action is possible.

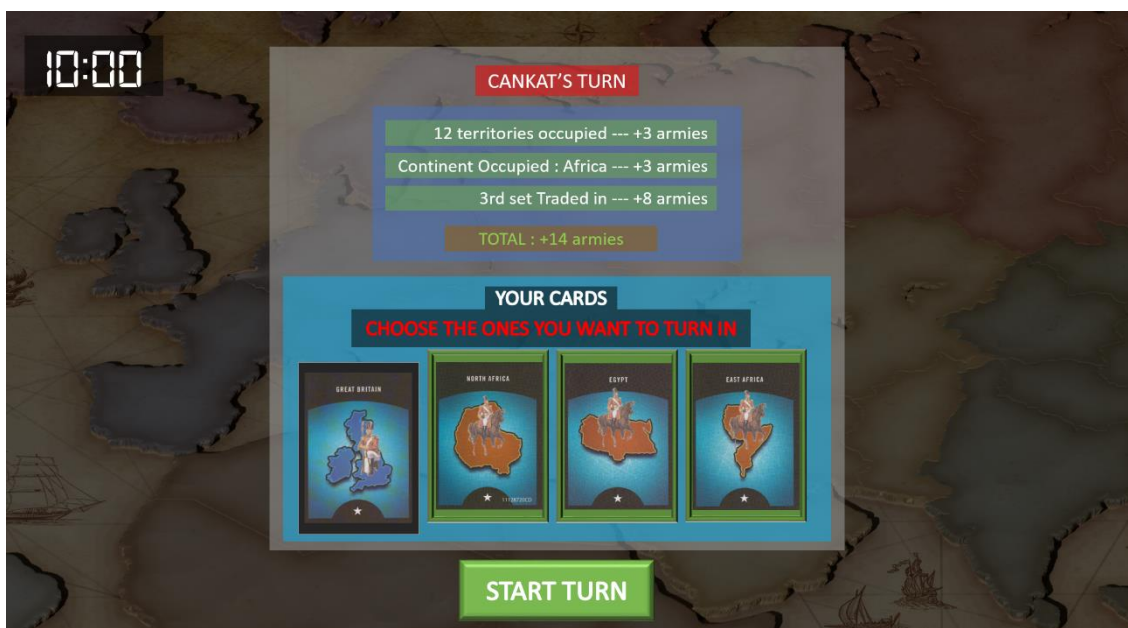


Figure 13: Game New Turn Screen Mock-up

When the player starts his turn, the timer on the top left of his screen begins to count down the 10 minutes he set up earlier on.



Figure 14: In-game Map Mock-up

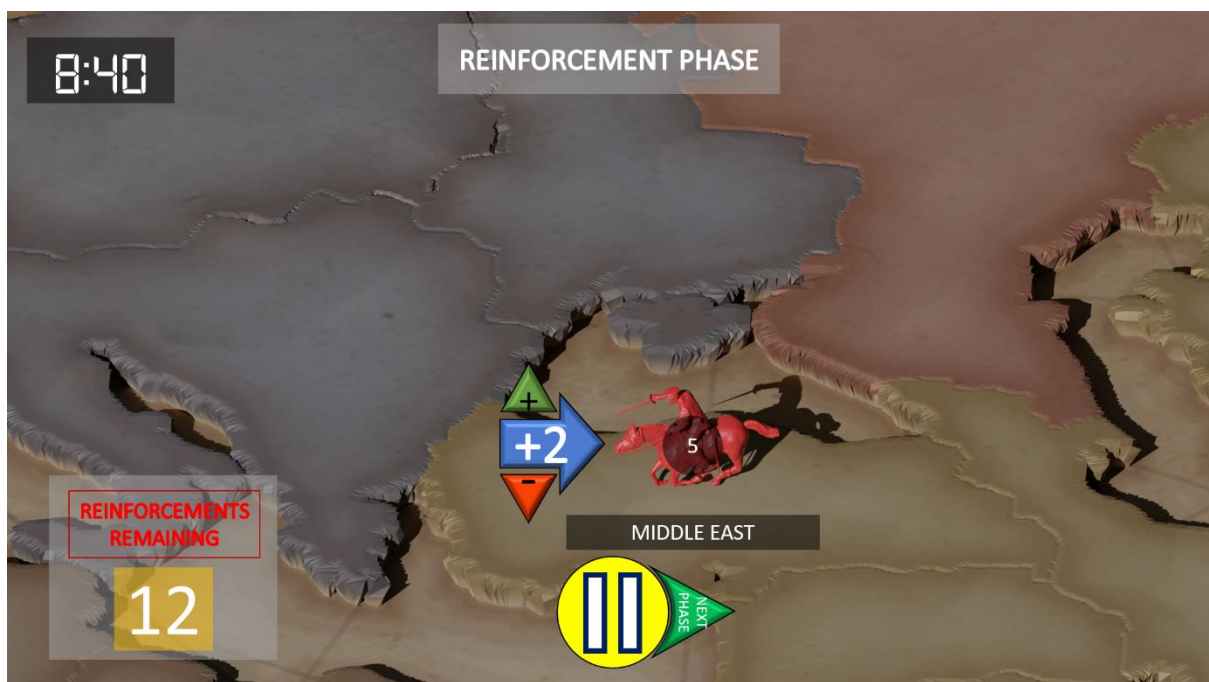


Figure 15: In-game Reinforcement Phase Mock-up

The game starts with the player seeing the entire map. He can zoom in and out from the map. In each of the territories every player can see the name of the territory and the number of armies on that territory with the color of the player that holds the territory. Our player for his turn starts by reinforcing his armies in the "Middle East" then goes on to reinforce his their territories. When he is happy with his decisions, he ends his "Reinforcement Phase" moving onto the "Attack Phase".

In the "Attack Phase" he can either choose to skip his tour or attack a territory held by his enemies. He decides to attack the blue player's territory of "Eastern Europe" with his armies in the "Middle East".

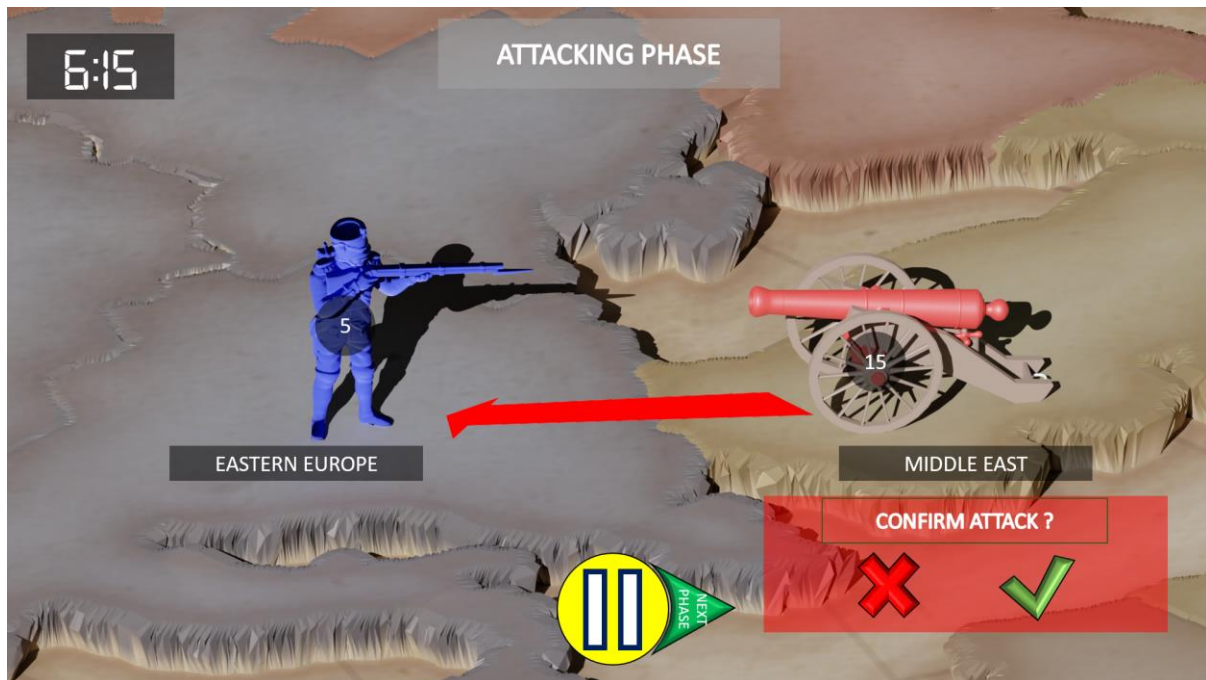


Figure 16: In-game Attack Phase Territory Select Screen Mock-up

He chooses the territory he wants to attack from by clicking on the armies he has on said territory and the territory he will be attacking by again clicking on the armies of his enemy. The arrows denote the flow of attack and the color denotes the color of the player attacking, since we see the red player all these UI elements are red. After choosing the territory he wants to attack, he is prompted to select the number of dice he wants to roll to try and capture the territory. He attacks with 3 dice. This process will result in either the attackers defeating any number of defenders based on their rolls or getting defeated by the defenders. In our case the defenders emerge victorious. Then our player attacks the same region with the 2 dice this time but once again he fails. If he had emerged victorious, he would have taken "Eastern Europe" with his remaining armies.

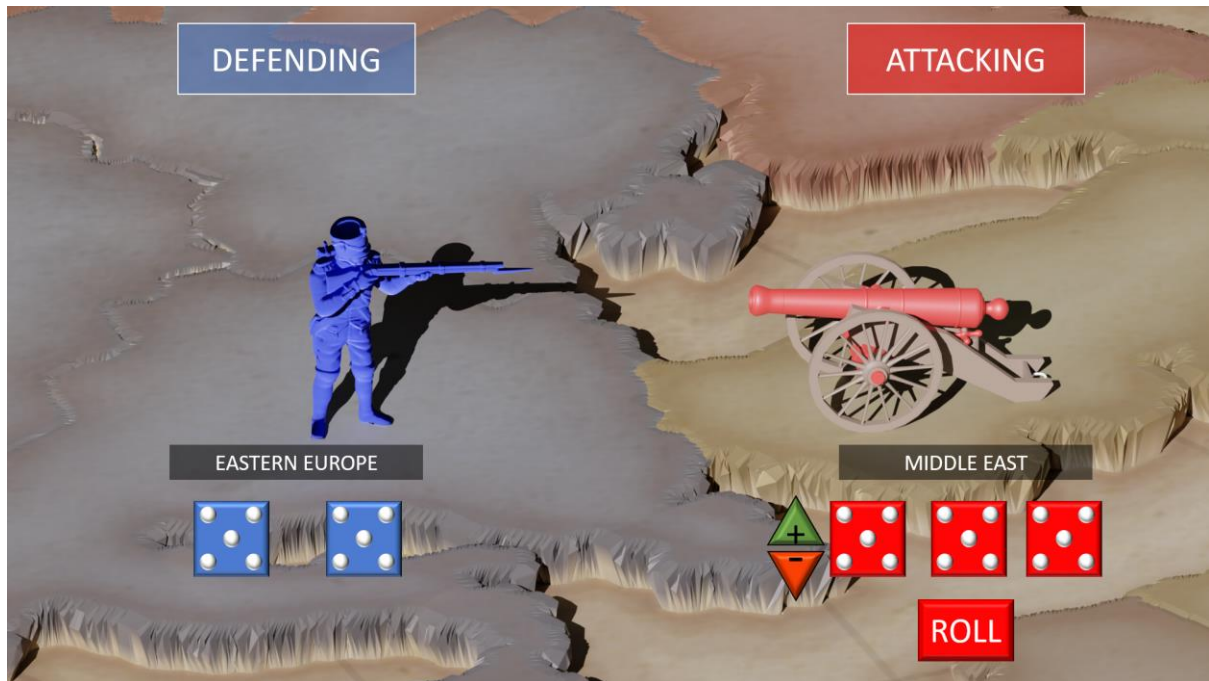


Figure 17: In-game Attack Screen Mock-up

After two crushing defeats in "Eastern Europe", our player decides to go to the next phase which is "Fortify Phase". He decides to fortify "Middle East" after his two unsuccessful attacks and moves his armies in "Ukraine" to join his armies in the "Middle East" territory, moving 5 of his 6 armies in "Ukraine" thus ending his turn.

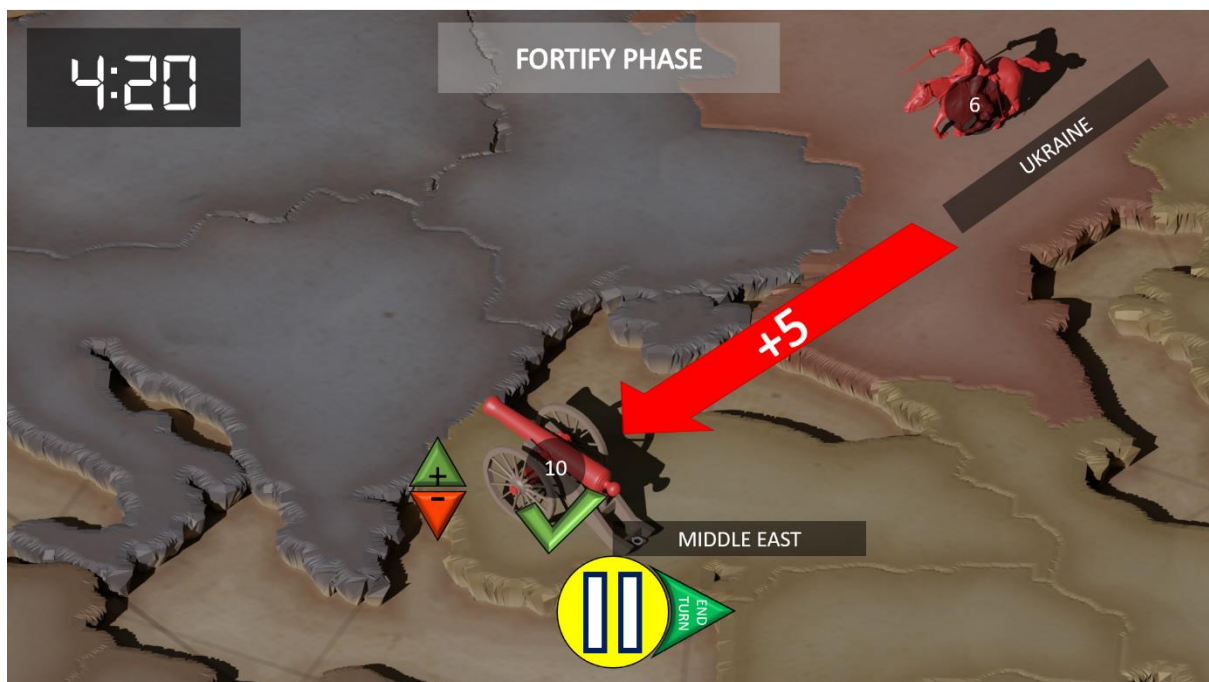


Figure 18: In-game Fortify Phase Territory Select Screen Mock-up

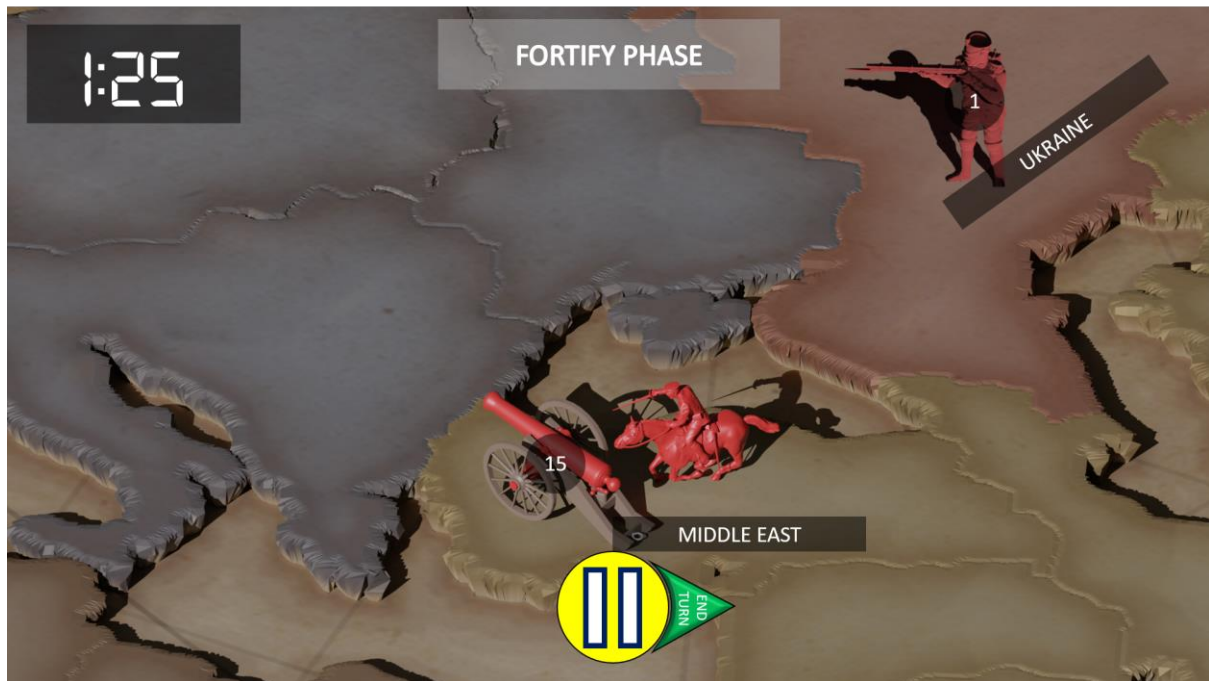


Figure 19: In-game Fortify Screen Mock-up

The game would go on like this until any one of our four players successfully occupy every territory or all players lose.

6. Glossary & references

<https://web.mit.edu/sp.268/www/risk.pdf>

<https://www.hasbro.com/common/instruct/risk.pdf>