



**Bilkent University**

Department of Computer Engineering

CS 319- Object Oriented Software Engineering

# Quantum Chess

Incredible in Thought

## Analysis Report

First Draft

Group 2A

October 7, 2017

# Contents

1. Implementation .....	3
2. Game Overview .....	3
2.1Power-ups.....	4
2.1.1Quantum Tunneling.....	4
2.1.2Quantum Superposition .....	4
2.1.3 Quantum Entanglement.....	4
3. Requirement .....	5
3.1 Functional Requirements.....	5
3.2 Non-Functional Requirements .....	5
3.3 Pseudo Functional Requirements (Constraints).....	6
4. System models.....	7
4.1 Use case model .....	7
4.1.1 Help.....	8
4.1.2 Start Game .....	8
4.1.3. Credits .....	9
4.2 Dynamic Models.....	12
4.2.1 Activity Chart.....	12
4.2.2 State Chart.....	13
4.2.3 Sequence Diagrams .....	15
4.3. Object Model.....	21
4.4. User Interface - Navigational Paths and Screen Mock-ups.....	23

# 1. Introduction

In the project, it is planned to implement a chess game that can be played by two players called Quantum Chess. There will be some similarities and some significant differences between Quantum Chess and the classic chess. In addition to accepted rules of the chess, it has been decided to add some of the quantum physics rules into the game so that it will be more challenging and more entertaining. Sometimes, the users cannot predict the moves of the rival. Since quantum rules make the movements unpredictable. So that, the game becomes difficult and ambiguous. In the game, quantum rules are demonstrated as power-ups. There are 3 power-ups. Users can use these power-ups if they do not exceed the limit of power-up. If one of the user surrenders or is defeated, the game finishes. In the analysis report, it is mentioned that the overview of the game, the functional and non-functional requirements of the game and UML diagrams of the project.

## 2. Game Overview

Quantum chess has no certain implications. Instead of implications, there are possibilities.

During the gameplay the position of a piece may become uncertain leaving the player of the piece and the opponent in the dark until the pieces true position is observed.

Quantum chess employs new knowledge from quantum physics to make the traditional game of chess more entertaining. The new different moves are represented as power ups and can be used in different ways during the game. The power-ups are limited. Users can not use the them if they surpass the limit. Except the superposition, each power-ups can be used 5 times.

## 2.1 Power-ups

The power-ups are quantum superposition, quantum tunneling and quantum entanglement (see Appendix A for more info).

### 2.1.1 Quantum Tunneling



In classic chess, if there is a piece in front of piece that you want to move, you would not be able move it. Yet, in the quantum chess, you can pass the piece in front of the piece that you move through the use of this power up.

### 2.1.2 Quantum Superposition



Quantum Superposition allows a piece to occupy two blocks at the same. However this piece cannot be moved unless it is observed by an attack of the enemy or by the player itself.

### 2.1.3 Quantum Entanglement



Quantum Entanglement allows the player to entangle one of his pieces with another opponent's piece such that the movement of the players who activated this power up is mirrored by the selected piece.

## 3. Requirement

### 3.1 Functional Requirements

- Players can start a new game.
- Players can control the pieces on the board by using the mouse.
- The game is played on one screen and computer.
- For each turn, players can choose the piece that they want to move according to Quantum Chess rules.
- All players can close the game whenever they desire.
- Players can use their limited number of power-ups during their turns so that they can challenge the other player or comfort herself/himself.
- Players can press "Help" button so as to comprehend what the rules of the game are.

### 3.2 Non-Functional Requirements

- Users can take rival's piece when the player's piece moves and encounters the rival's piece.
- **Usability:**
  - Players must be able to play a game without any registration.
  - Quantum power ups can affect the usability of the game. Therefore players have limited number of power ups to use in a game.
  - The game does not need installment.
- **Reliability:**
  - This game will not ship with any known treats that cause to be defected.
  - There is no data storage in game in order to be shared by third person.
- **Portability:**
  - Quantum Chess will be taking different run-time environments into account.

- **Performance:**
  - Player moves should be acknowledged within 1 second.
- **Licensing:**
  - Game will have open source license (GPL) in order to be freely used and shared.

### 3.3 Pseudo Functional Requirements (Constraints)

- Quantum Chess will be implemented in Java.
- The game will not need any network connection.

## 4. System models

### 4.1 Use case model

This section gives information about the main use case model of Quantum Chess game, more comprehensive explanation is below.

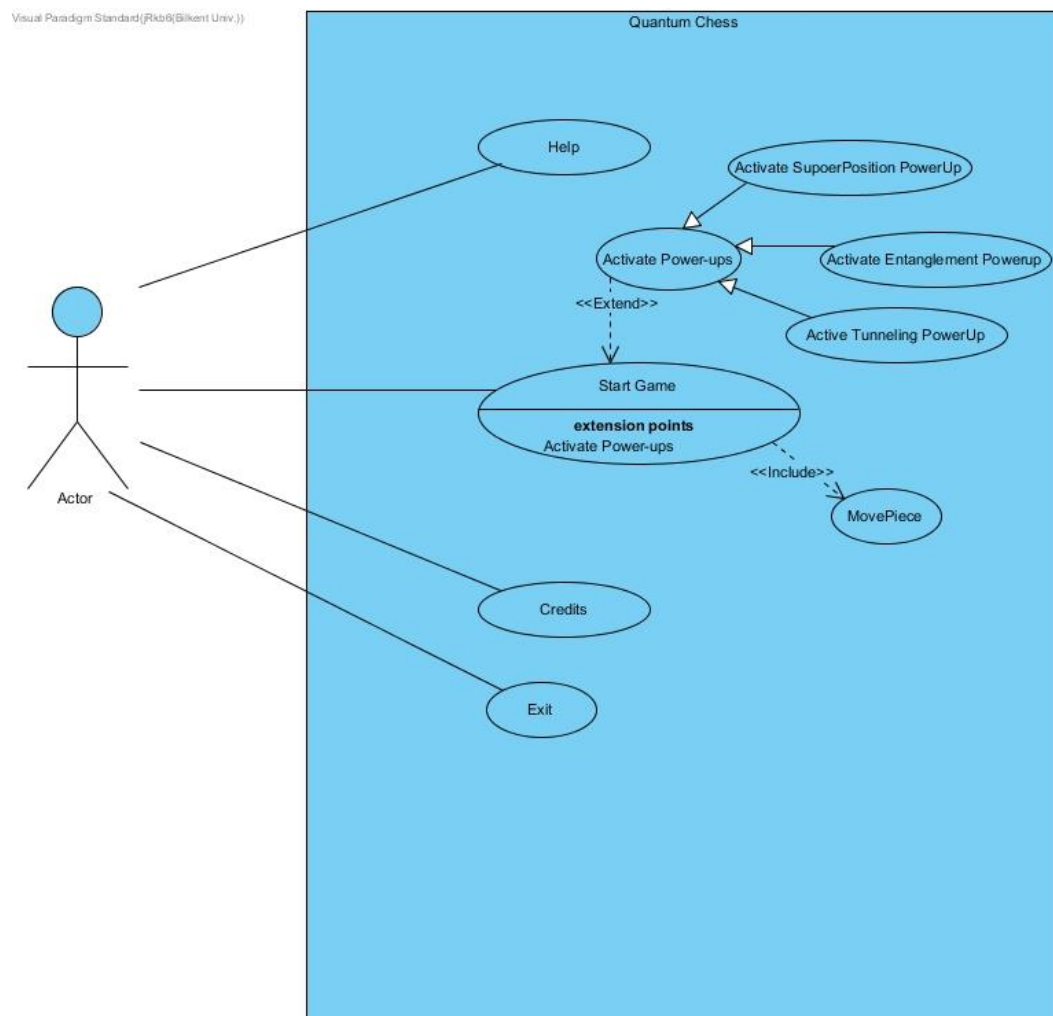


Figure 5.1- Illustrates the use case model of Quantum Chess

#### 4.1.1 Help

---

**Use Case Name:** Help

---

**Primary actor:** Player

---

**Stakeholders and Interests:**

- Player wants to learn about “Quantum Chess”
  - System shows how to play and short tutorial
- 

**Pre-condition:** -

**Post- condition:** -

---

**Entry Condition:** Player selects “Help” from Main Menu.

**Exit Condition:** Player selects “Back” to return Main Menu.

---

**Success Scenario Event Flow:**

1: System displays information about “Quantum Chess” and short tutorial

---

**Alternative Flows:**

A: If player wants to back to the main menu:

A1: Player selects “Back” button

A2: System displays Main Menu

---

#### 4.1.2 Start Game

---

**Use Case Name:** Start Game

---

**Primary Actor:** Player

---

**Stakeholders and Interests:**

- 2 Player want to play “Quantum Chess” with each other
  - System keeps status of the game
- 

**Pre-condition:** After player select “Start Game”, game will start and 2 player start to play with each other.

**Post-condition:** Player who win the game, system shows that for example Player1 is the winner

---

**Entry Condition:** Player selects “Start Game” button from Main Menu.

**Exit Condition:** Player selects “X” button top of the page.

---



#### Success Scenario Event Flow:

1. Game start after one of the player push "Start Game" button.
2. Players enter their name.
3. Player starts playing each other with use case move piece.
4. Players play normal chess until one of them push "Power-Ups".
5. Player should choose which limited number of Power-Ups he/she will use  
Steps 3 and 4 repeats until one of them win the game.
6. System shows winner's name.

*If player want to revenge they should repeats these steps again*

---

#### Alternative Flows:

---

#### 4.1.3. Credits

---

**Use Case Name:** Credits

---

**Primary Actor:** Player

---

**Stakeholders and Interests:**

-Player wants to learn information about developers of "Quantum Chess" and how to contact with them.

-System displays contact information about developers.

---

**Pre-condition:** Player should enter this case from main menu.

**Post-condition:** -

---

**Entry Condition:** Player selects "Credits" from main menu.

**Exit Condition:** Player selects "Back" to return Main Menu.

---

#### Success Scenario Event Flow:

System displays contact information about developers of "Quantum Chess"

---

#### Alternative Flows:

A: If player wants to back to the main menu:

A1: Player selects "Back" button

A2: System displays Main Menu

---

#### 4.1.4. Exit

---

**Use Case Name:** Exit

---

**Primary Actor:** Player

---

**Stakeholders and Interests:**

-Player wants to exit to the game.

-System disposes the game.

---

**Pre-condition:** Player should start the game.

**Post-condition:** -

---

**Entry Condition:** Player clicks to the exit button.

**Exit Condition:** -

---

**Success Scenario Event Flow:**

System exits and disposes the game regarding to players action. There is no alternative flow for the disposing game.

---

#### 4.1.4. Activate PowerUps

---

**Use Case Name:** Activate PowerUps

---

**Primary Actor:** Player

---

**Stakeholders and Interests:**

-Player wants to use one of the power ups.

-System disposes the game.

---

**Pre-condition:** Player should start the game and click the one of the power up buttons if there is available.

**Post-condition:** -

---

**Entry Condition:** Player clicks one of the power up buttons.

**Exit Condition:** Player does the move with the selected power up if it's allowed by the game.

---

**Success Scenario Event Flow:**

Player selects a power up from the panel, then selects a piece that would be used with power ups. Then moves the piece or superpose it according to the power up rules.

---

**Alternative Flows:**

A: If player wants to use power up.

A1: Player selects one of the available power ups.

A2: Player choses a piece.

A3: Player decides the cancel movement with power up.

#### 4.1.5. MovePiece

---

**Use Case Name:** MovePiece

---

**Primary Actor:** Player

---

**Stakeholders and Interests:**

-Game turn is on the player.

-System displays the turn on player.

---

**Pre-condition:** Player should play the game and wait his turn.

**Post-condition:** Player takes his and opponents movement in consideration.

---

**Entry Condition:** Player selects piece from the board.

**Exit Condition:** Player moves the piece according to the system rules.

---

**Success Scenario Event Flow:**

1. Player selects the piece.
  2. System shows the available movements to the player.
  3. Player decides one of the movements or selects another piece.
  4. Players move the piece to the available location.
  5. System changes player turn to his opponent.
-

## 4.2 Dynamic Models

### 4.2.1 Activity Chart

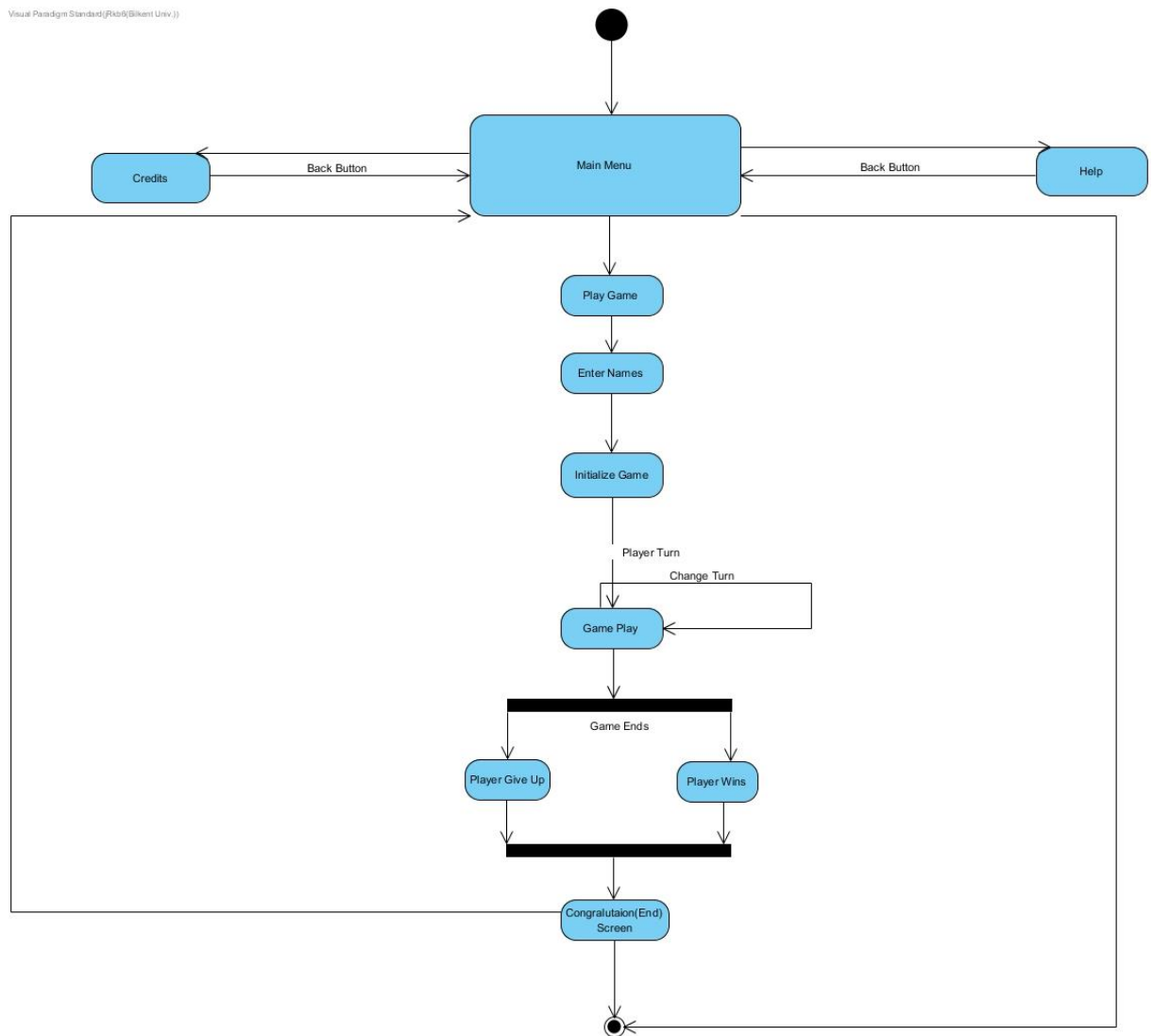


Figure 4.2.1.1: Illustrates the activity diagram for the player

## 4.2.2 State Chart

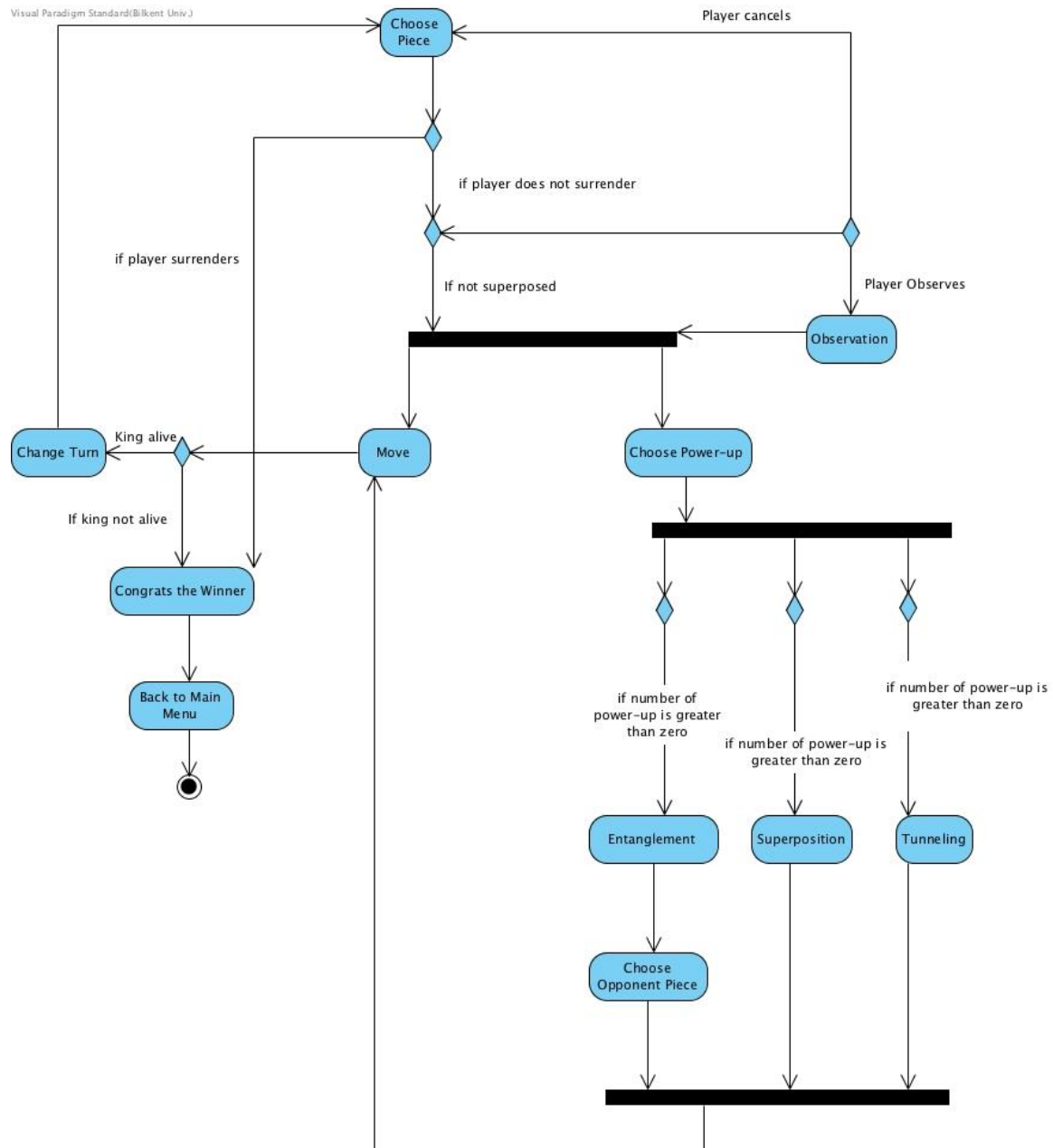


Figure 4.2.1.2: Illustrates the state chart diagram for the player

This state diagram describes the behavior of the Game Play in the activity diagram in details. If the player wants to do normal chess move rule, it moves the piece. If the piece is superposed already, player has to observe the piece in order to move it. If player wants to use Quantum PowerUps, it has three option for power ups and each player has limited number of power ups. Player first selects the power up which he/she wants to use, then moves the piece. If Quantum Entanglement is chosen, player will choose one of the opponent pieces. If the movements are valid, player will be allowed to move its piece. After movement, game turn will change to other player if its king is still alive. Same process will be done for the second player, until one of the kings die or one of the players gives up. Congratulations screen will be shown after game ends and program redirects to the main menu screen.

## 4.2.3 Sequence Diagrams

Scenario #1: The General Gameplay

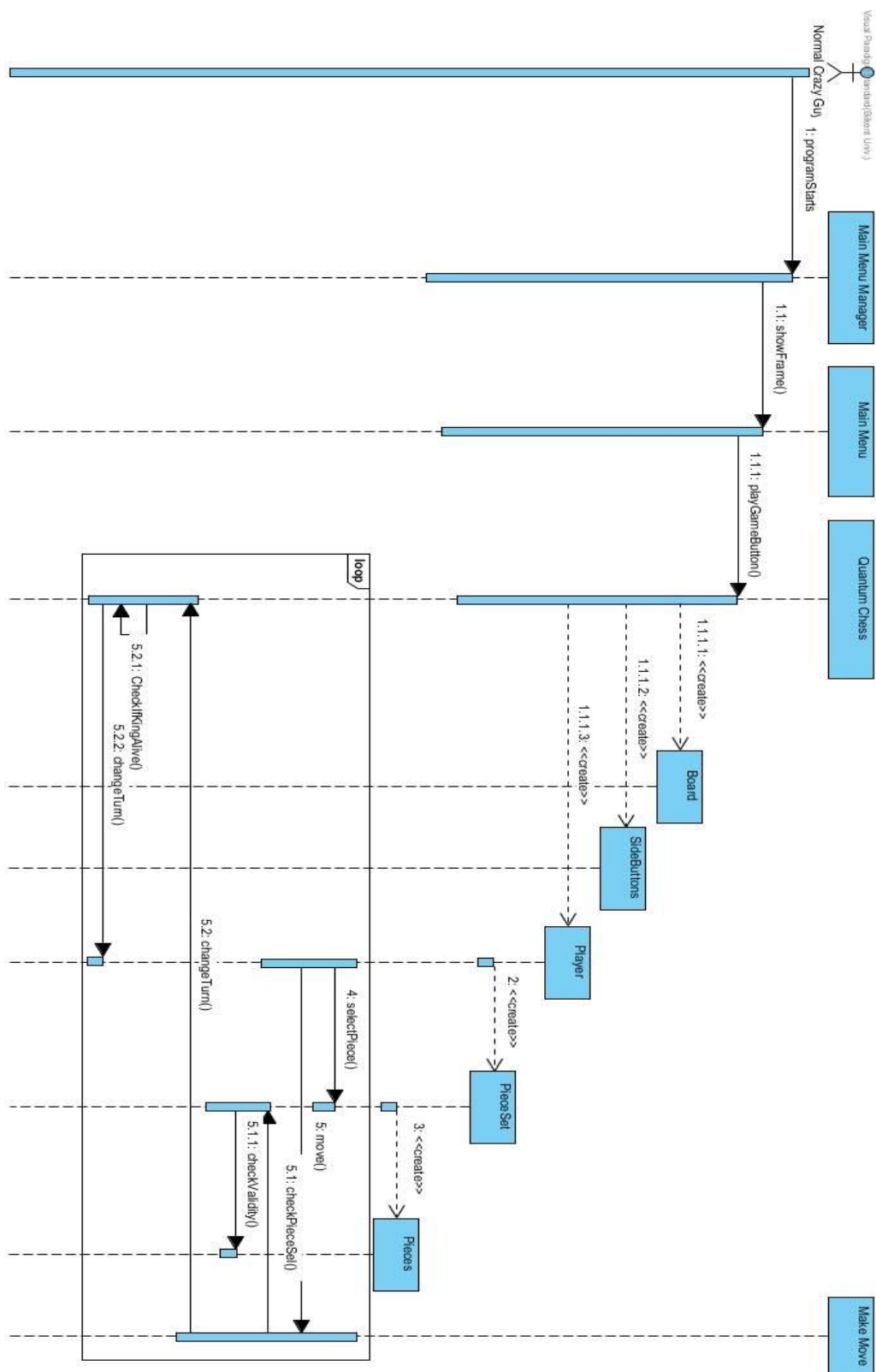
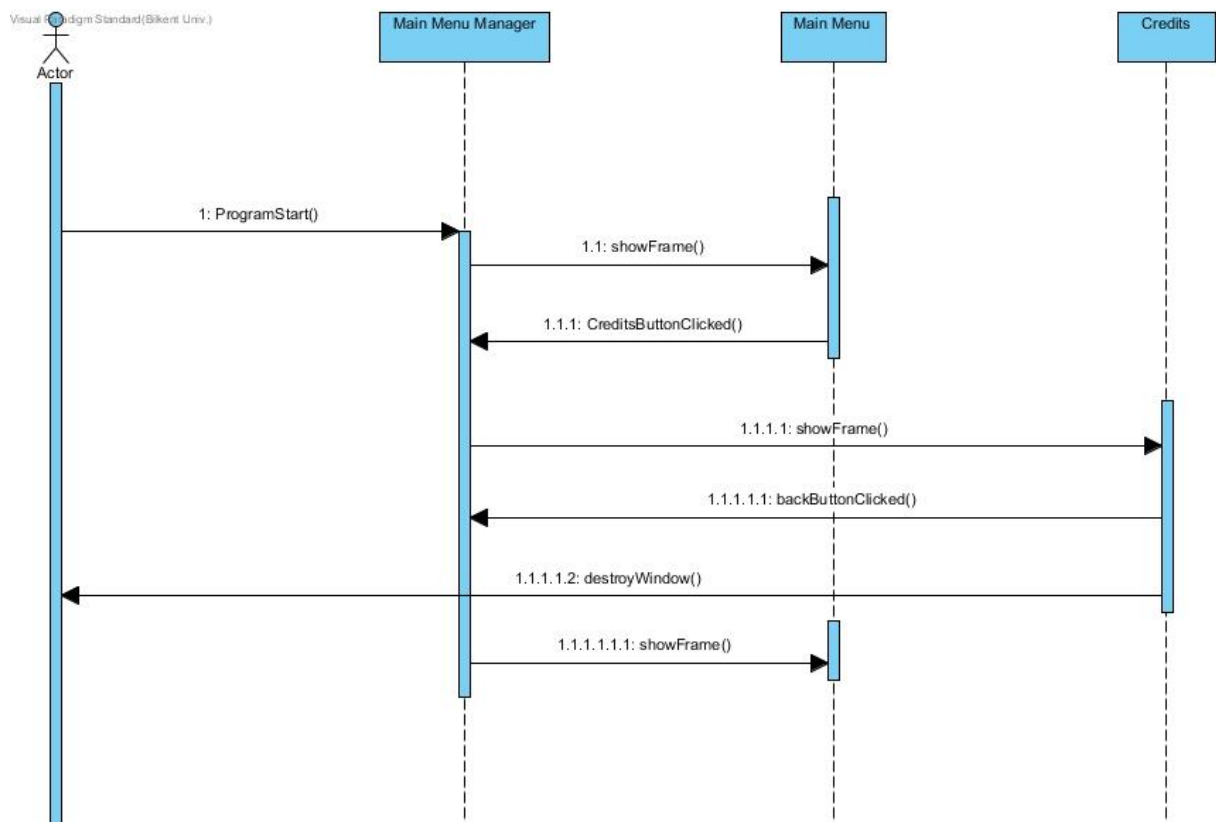


Figure 4.2.3.1: Illustrates the general gameplay sequence

One of the players starts the program and is introduced immediately to the Main Menu. The different frames are handled by the Main Menu Manager. The player presses the Start Game button, starting the game. The QuantumChess creates the Board, the SideButtons and the Players. The Board is responsible for the visuals of the chess board. The SideButtons will have the power up buttons, cancel and surrender which will have in game use. The creation of the Players creates also the PieceSet which will draw the pieces on top of the board and which will in turn create the Pieces needed for the game. After everything has been created the first player can proceed with their turn. They select a piece (4) and then decide to move the piece selected (5). There are several types of movements that can be done thus they are discussed in different scenarios collectively named Make Move. The system checks the piece selected in the PieceSet (5.1) and then checks for the validity (5.1.1) of the move through the piece itself. If the move is valid it will be made, whereby with the move done the turn will change to the other player. This will continue until one of the player loses their King or surrenders, thus losing the game.

#### Scenario #2 - Credits:

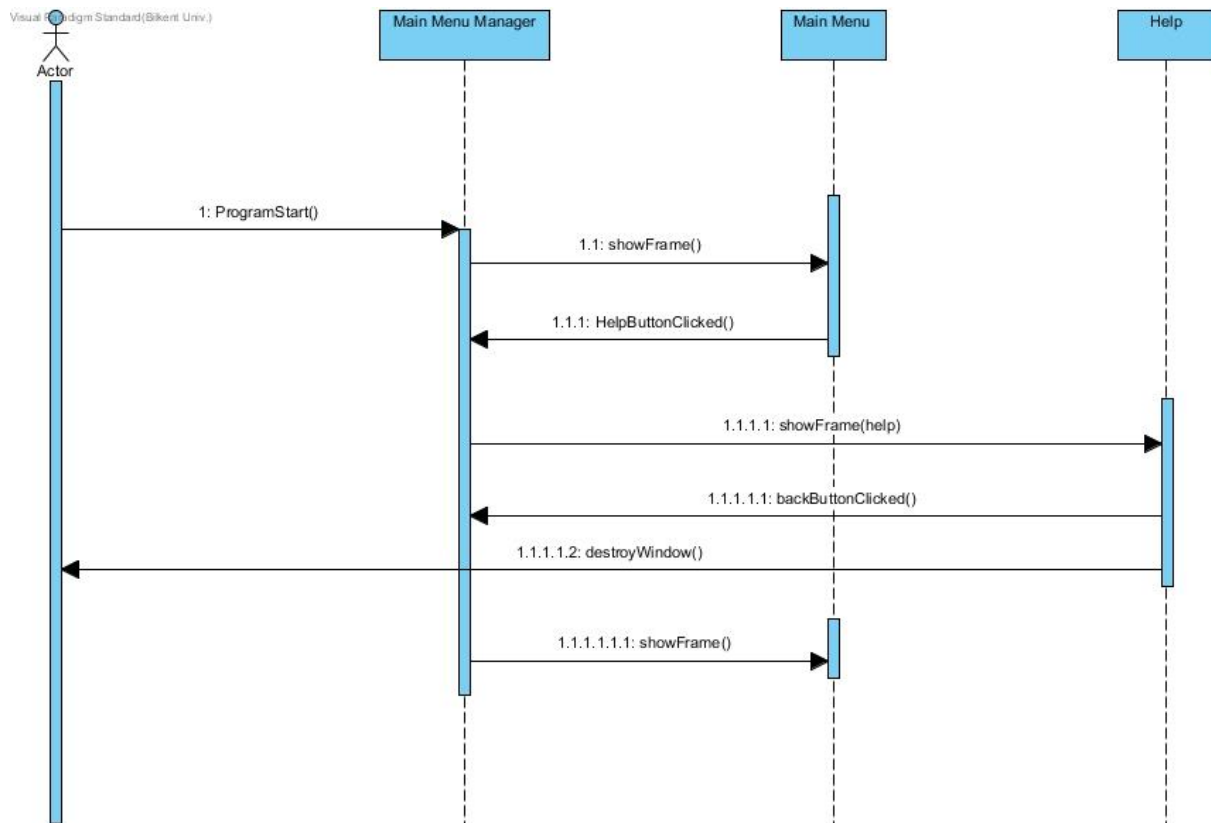




**Figure 4.2.3.2: Illustrates the scenario for Credits**

Players who want to see the developers of the game, clicks the “Credits” button on the main menu. Credits frame will be shown and give information about the developers of the game. Player will go to the main menu with the “Back” button on the credits frame after the screen shows the desired information.

Scenario #3 - Help:



**Figure 4.2.3.3: Illustrates the scenario for Help**

Players who want to learn how game is played, after game starts one of them clicks the “Help” button. In help frame, the game will be explained with a tutorial and text. After short information and tutorial, player can go to main menu by pressing back button.

#### Scenario #5 – Moves Possible: Normal Chess Move:

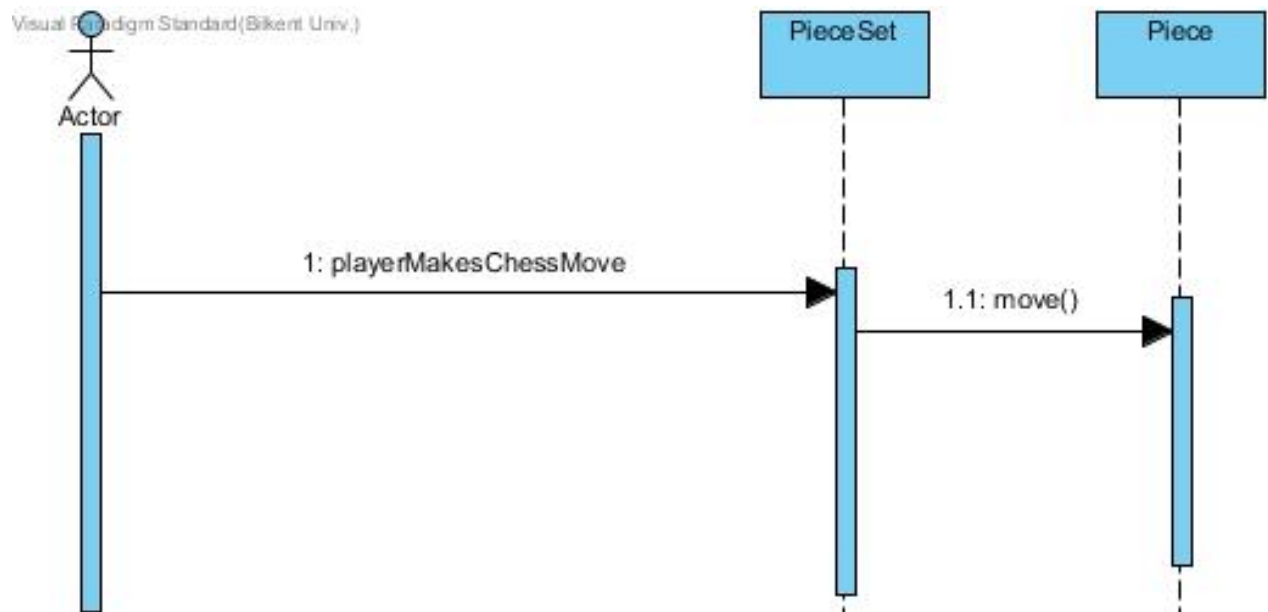
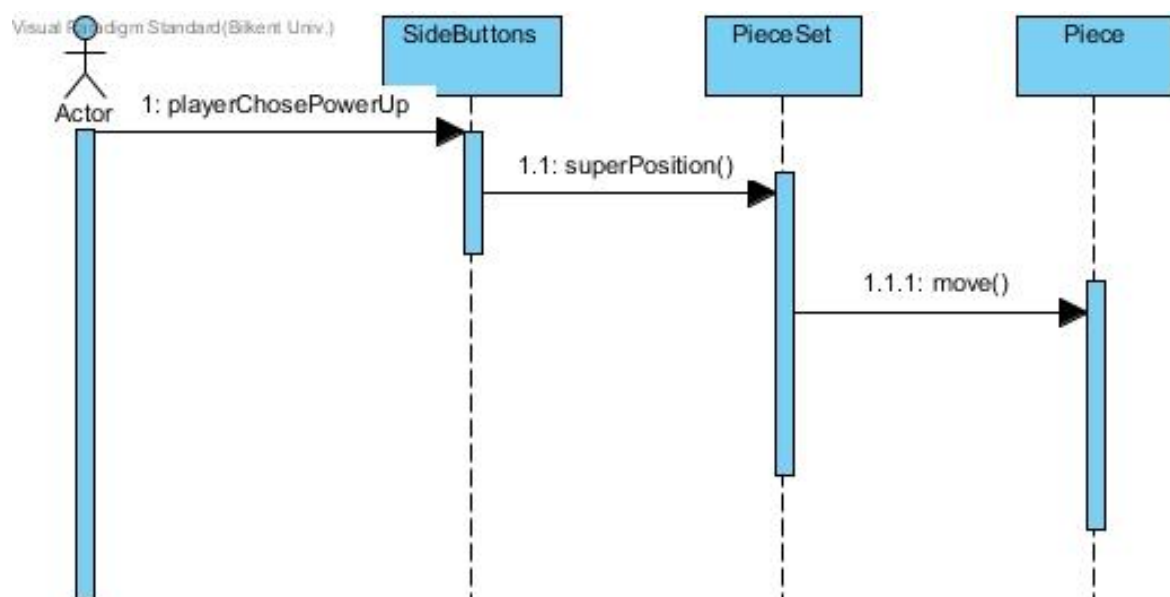


Figure 4.2.3.4: Illustrates the normal chess move scenario

When the player decides to simply to a chess move, he does not select any power up. He just continues with the selection of the place where his piece will be moved if valid (the validity is check is shown in The General Game Play).

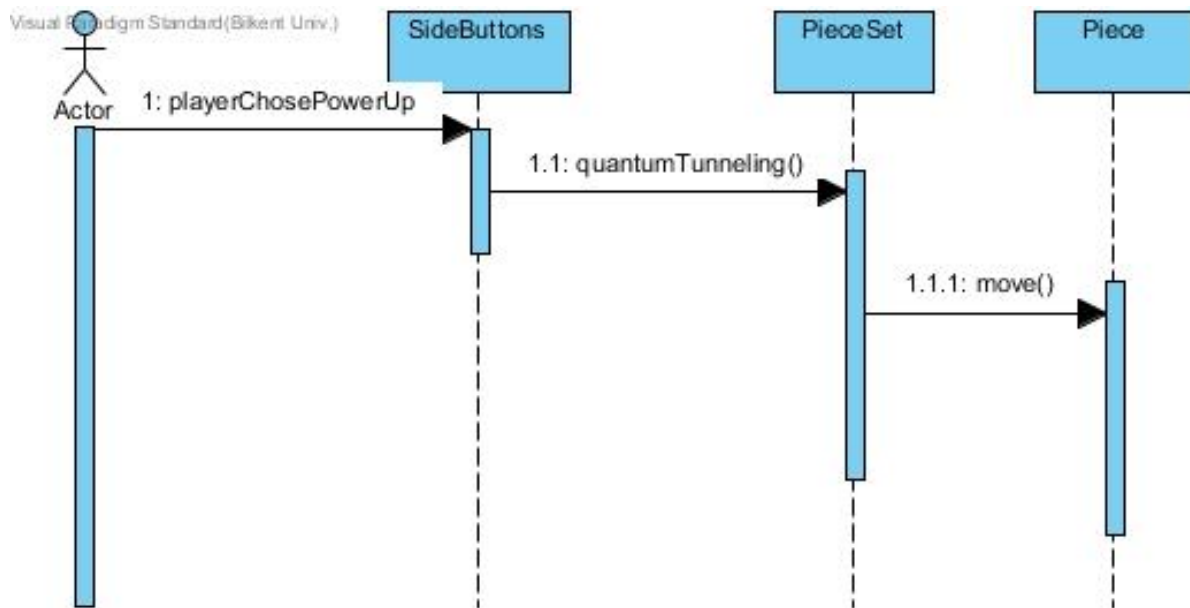
#### Scenario #6 – Moves Possible: Quantum SuperPosition:



**Figure 4.2.3.5: Illustrates the quantum superposition use scenario**

When the player decides to use the Quantum SuperPosition power up. The superPosition button is pressed thus making it possible for the piece to be in super position. The piece will perform the movement if it is valid (the validity check is shown in The General Game Play).

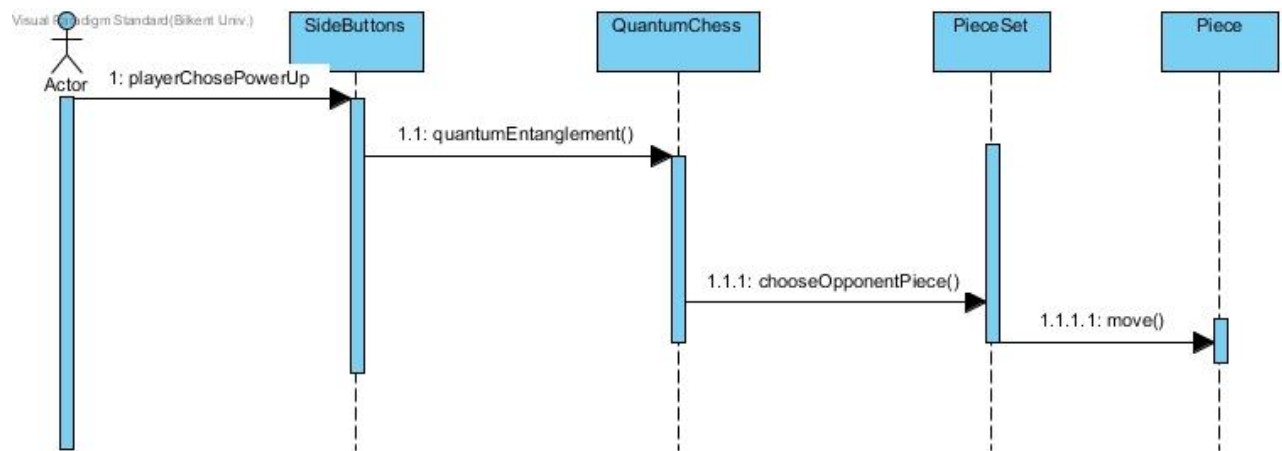
Scenario #7 – Moves Possible: Quantum Tunneling:



**Figure 4.2.3.6: Illustrates the quantum tunneling use scenario**

When the player decides to use the Quantum Tunneling power up. The tunneling button is pressed thus making it possible for the piece to move through other pieces. The piece will perform the movement if it is valid (the validity check is shown in The General Game Play).

#### Scenario #8 – Moves Possible: Quantum Entanglement:



**Figure 4.2.3.: Illustrates the quantum entanglement use scenario**

When the player decides to use the Quantum Tunneling power up. The tunneling button is pressed thus making it possible for the piece be entangled with an opponent's piece. After choosing the power up the player will choose an opponent's piece and then perform the move if valid (the validity check is shown in The General Game Play).

## 4.3. Object Model

Visual Paradigm Standard (Bilkent Univ.)

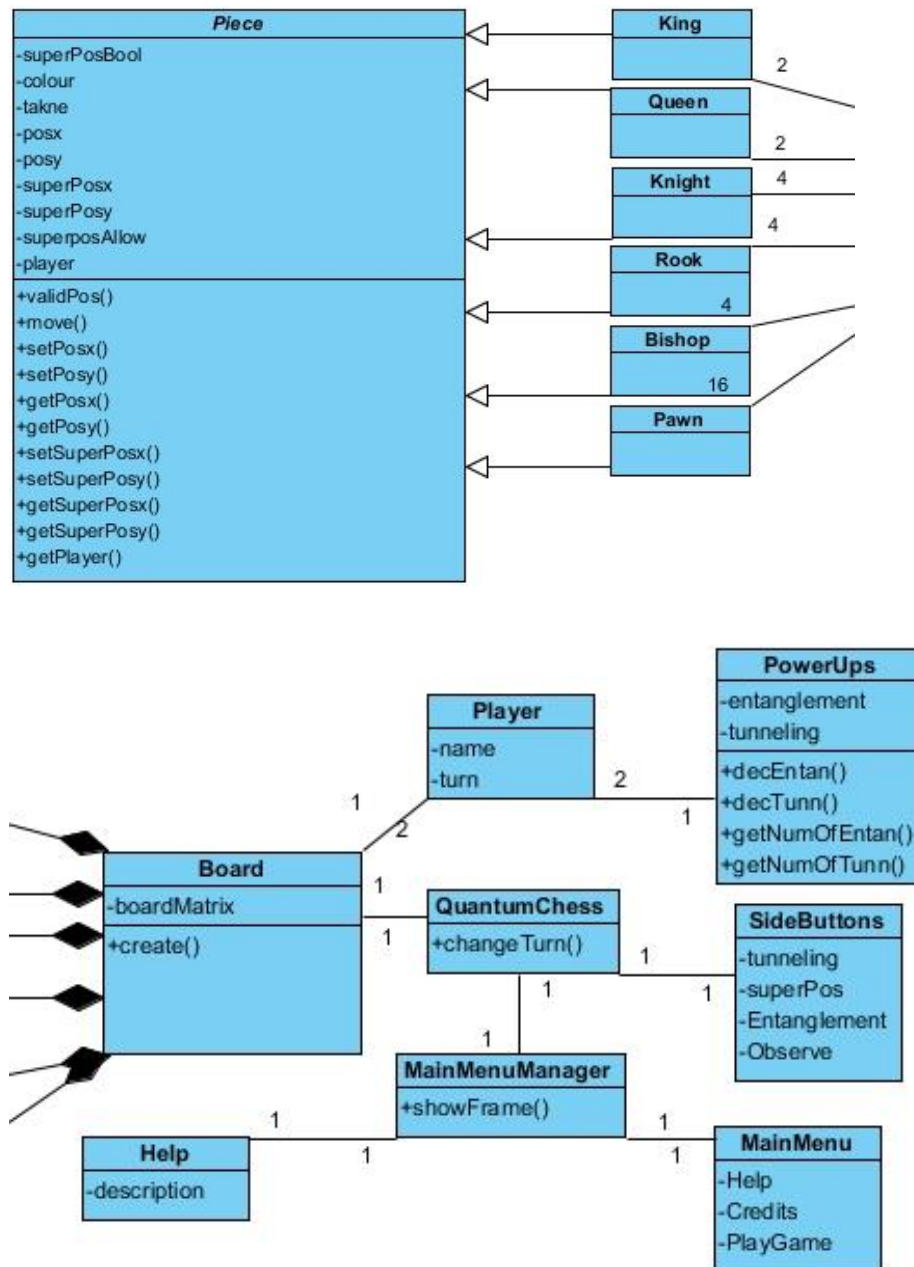
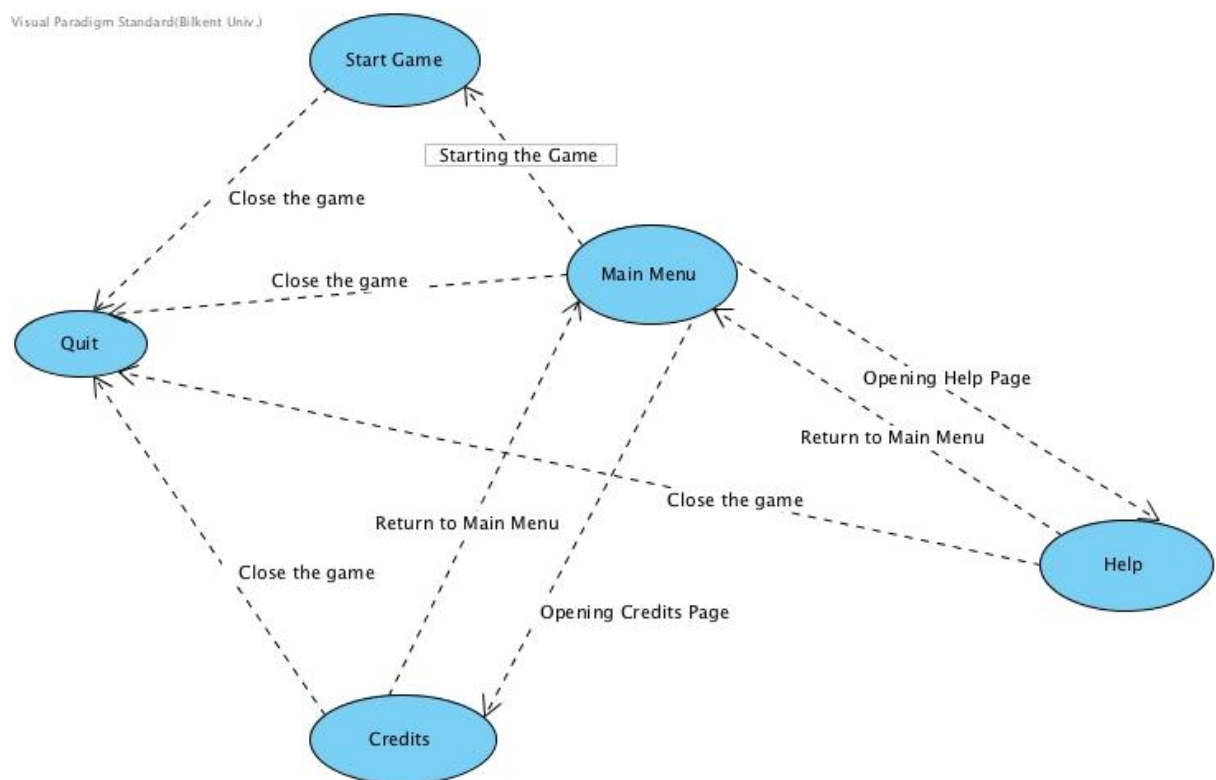


Figure 4.3.1: Illustrates the object model diagram

- **Piece:** Is an abstract class that has the necessary attributes and operations for different chess pieces.
- **King:** Is a class which will extend the Piece class and implement the functions in the appropriate way for the specific piece
- **Queen:** Is a class which will extend the Piece class and implement the functions in the appropriate way for the specific piece
- **Rook:** Is a class which will extend the Piece class and implement the functions in the appropriate way for the specific piece
- **Bishop:** Is a class which will extend the Piece class and implement the functions in the appropriate way for the specific piece
- **Knight:** Is a class which will extend the Piece class and implement the functions in the appropriate way for the specific piece
- **Pawn:** Is a class which will extend the Piece class and implement the functions in the appropriate way for the specific piece
- **PieceSet:** its main function is to have an array which will hold the instances of different pieces in an appropriate amount such that the player will be able to interact with his pieces appropriately. The PieceSet is also responsible of drawing the figures.
- **Player:** has a pieceSet as well as the player name and turn whereby the pieces can be connected to the appropriate player.
- **PowerUps:** will hold the amounts left for the power ups to be used.
- **QuantumChess:** is the class which will hold the complete game including board, two players and the side buttons.
- **Board:** will generate the board.
- **SideButtons:** will include all the power up buttons and other function buttons that will be used during the gameplay.
- **MainMenuManager:** is the manager of all the frames that will be shown in the software.
- **Help:** Will display information helpful for the player, being that he does not know how to play the game.
- **Credits:** Will display general information about the game and creators.
- **MainMenu:** will include navigation buttons to start the game or go to the help frame or credits frame.

The first stage analysis was done by trying to make the design as close to MVC as possible. However after part of the implementation was done, it was decided that the ease of implementation would be preferred over maintainability. Thus the view and the controller will be implemented together.

## 4.4. User Interface - Navigational Paths and Screen Mock-ups



**Figure 4.4.1** illustrates the user interface

When the program is started, the home page is shown at first. The users have three options that are Start Game, Help and Credits. By pressing Start Game, it is desired from players to write their names. Then, after pressing Start Game button Quantum Chess starts. If they press Help button, it represents the description of the game. Authors are demonstrated if they choose Credits button. Users can quit the game by pressing the X button that is at right corner of the screen.

#### 4.4.1 The Main Menu

The main menu welcomes the player and provides him with some actions that the player can take. He can ask for help in understanding the game. Look at the credits. Start immediately the game.

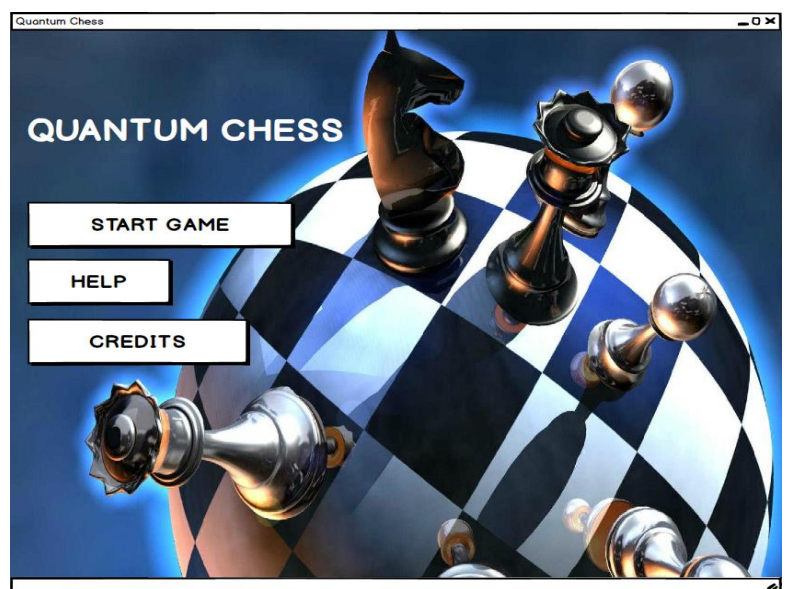


Figure 4.4.1.1: Illustrates how the main menu will look

After clicking the start game button, the player will have to write the name of the first player and the second player. Then click start game to start the game.

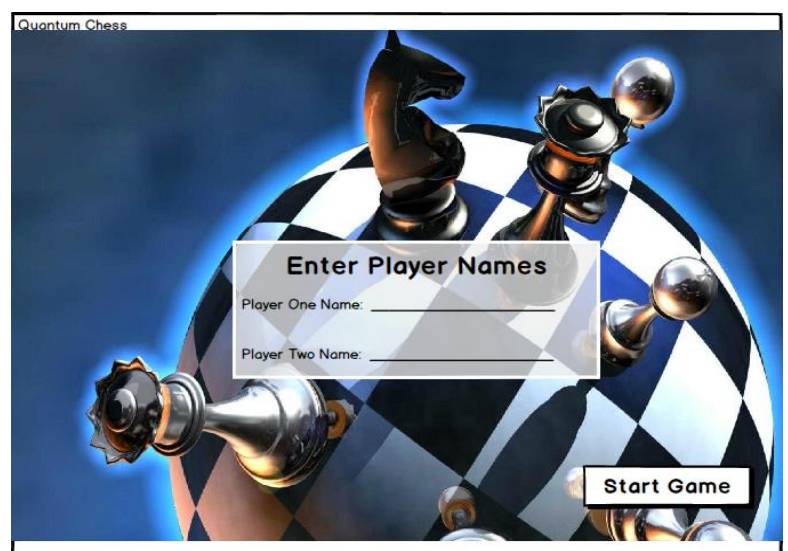




Figure 4.4.1.2: Illustrates how the frame in which the player will write the two players name, will look like.

If the player clicks credits he will be presented with a screen similar to figure 4.4.1.3. The player has the opportunity to go back to the main menu by clicking back to menu.

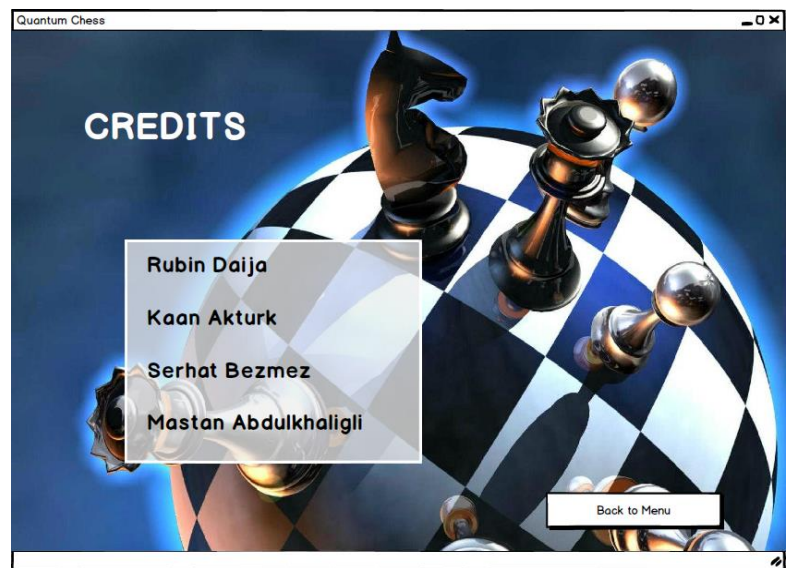


Figure 4.4.1.3: Illustrated the credits frame

If the player clicks help he will be presented with a screen similar to figure 4.4.1.4. Information about the game mechanics of Quantum Chess will be presented. The player has the opportunity to go back to the main menu by clicking back to menu.



Figure 4.4.1.4: Illustrates the help frame

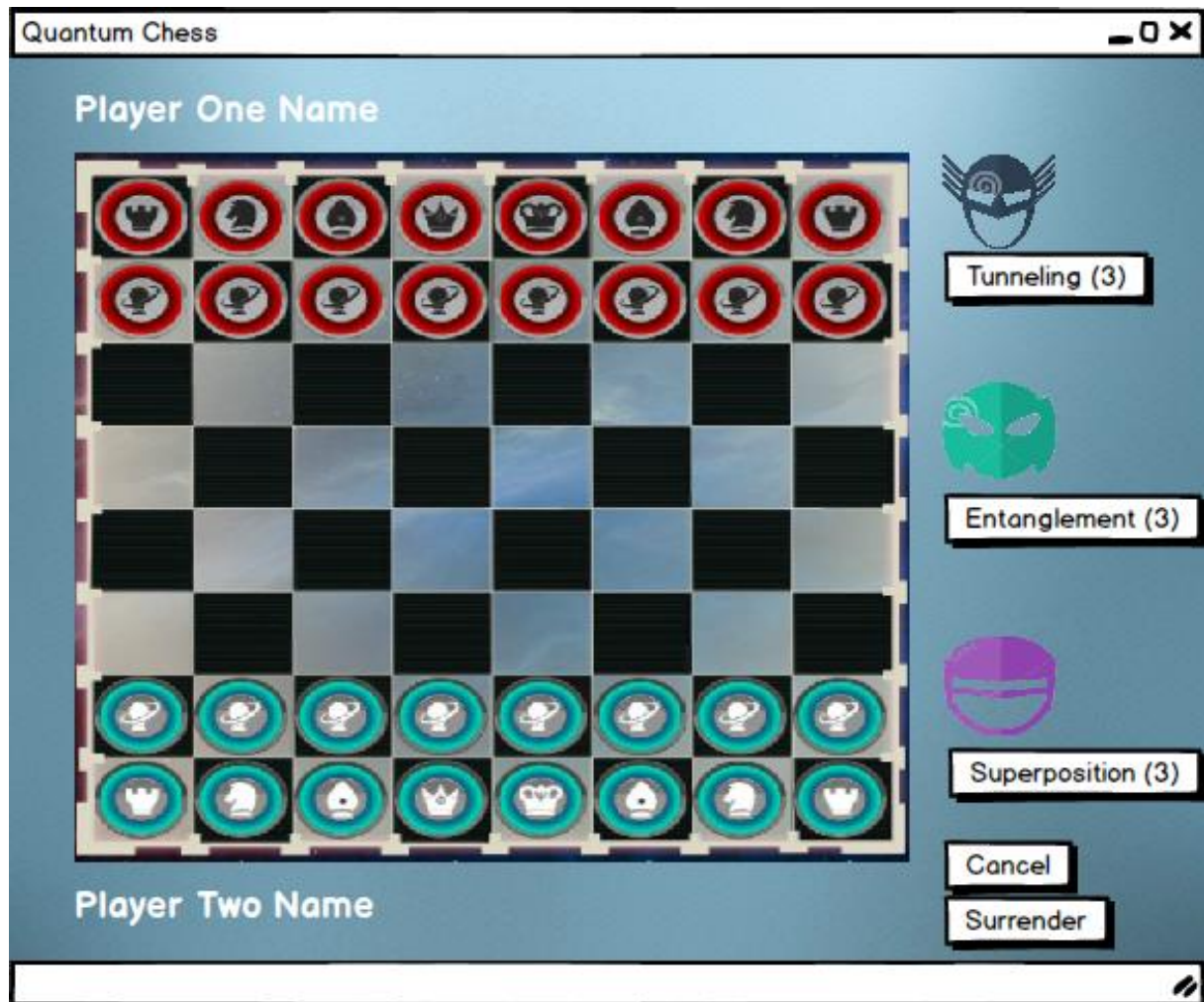


Figure 4.4.1.5: illustrates how actual gameplay can look like

When the game starts the players will be presented with a screen similar to Figure 4.4.1.5.

They can continue the gameplay until one of them surrenders or loses.

# Appendix A

The three power ups that are to be implemented in this game have inspiration from real Quantum Physics principles.

The following two links show examples of how this game has been implemented by different parties.

[https://www.youtube.com/watch?v=Hi0BzqV\\_b44](https://www.youtube.com/watch?v=Hi0BzqV_b44)

[http://store.steampowered.com/app/453870/Quantum\\_Chess/](http://store.steampowered.com/app/453870/Quantum_Chess/)

The Quantum Chess that will be implemented has completely different game mechanics from the two provided above. Though Quantum Super Position is similar the use in the game play changes drastically.