**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. Introduction 3](#_Toc495173620)

[2. Game Overview 3](#_Toc495173621)

[2.1Power-ups 3](#_Toc495173622)

[2.1.1Quantum Tunneling 4](#_Toc495173623)

[2.1.2Quantum Superposition 4](#_Toc495173624)

[2.1.3 Quantum Entanglement 4](#_Toc495173625)

[3. Requirement 5](#_Toc495173626)

[3.1 Functional Requirements 5](#_Toc495173627)

[3.2 Non-Functional Requirements 5](#_Toc495173628)

[3.3 Pseudo Functional Requirements (Constraints) 6](#_Toc495173629)

[4. System models 7](#_Toc495173630)

[4.1 Use case model 7](#_Toc495173631)

[4.1.1 Help 7](#_Toc495173632)

[4.1.2 Start Game 8](#_Toc495173633)

[4.1.3. Credits 9](#_Toc495173634)

[4.2 Dynamic Models 10](#_Toc495173635)

[4.2.1 Activity Chart 10](#_Toc495173636)

[4.2.2 State Chart 11](#_Toc495173637)

[4.2.3 Sequence Diagrams 13](#_Toc495173638)

[4.3. Object Model 21](#_Toc495173639)

[4.4. User Interface - Navigational Paths and Screen Mock-ups 24](#_Toc495173640)

# 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 normal chess that is known by everybody. In addition to accepted rules of 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.

In the analysis report, it is mentioned that the overview of the game that leads users to gain knowledge about the game. In the following, the functional requirements of the game that show the duty of the system and non-functional requirements of the game that describe work of the system are presented in the paper. Eventually, the system models of the projects that contains Use Case Model, Dynamic Model, Object and Class Model, and User Interface are demonstrated.

# 2. Game Overview

Quantum chess has no certain implications. Instead of implications, there are possibilities. Some of the place of pieces are uncertain. Rival cannot know the certain place of the pieces. S/he can only know the possibilities of where objects can move.

Quantum chess can be regarded as the developed version of chess. There are new movements that are shaped by quantum physic. In the game, there are power-ups that offers players these movements for their pieces.

## 2.1Power-ups

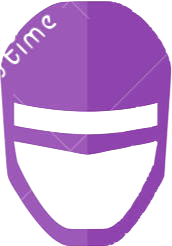
Players have three different power-ups that are quantum superposition, quantum tunneling and quantum entanglement.

### 2.1.1Quantum Tunneling



In the classic chess, if there is a piece in front of piece that you want to move, you would not move it. Yet, in the quantum chess, you can pass the piece in front of the piece that you move. It is called Quantum Tunneling.

### 2.1.2Quantum Superposition



In the game, some pieces can also have super features. In the classic chess, the king can move as bishop and rook move. But, in our game, the king can also move as knight moves. Other pieces have also abilities that lead them to move as another piece move. It is called Quantum Superposition.

### 2.1.3 Quantum Entanglement



When you move your one of pieces, you can also force your rival’s piece to rotate in same movement thanks to quantum entanglement.

# 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.**

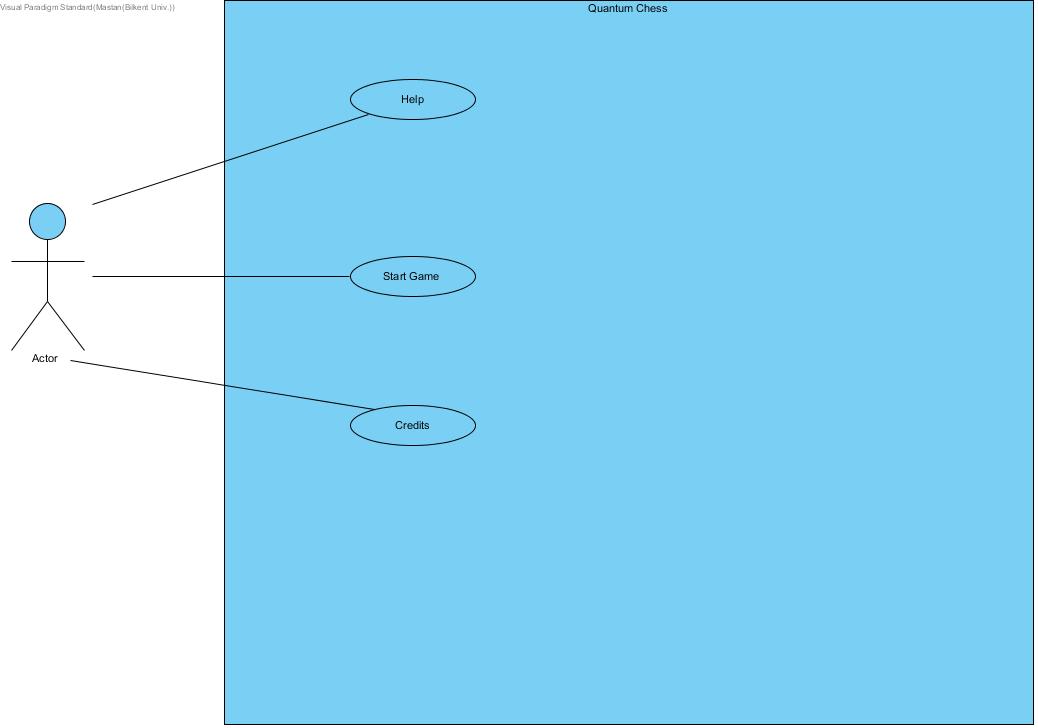
## 3.3 Pseudo Functional Requirements (Constraints)

* 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 wil start and 2 player start to play with each other.

**Post-condition:** Playerwho 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
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

1. 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:**

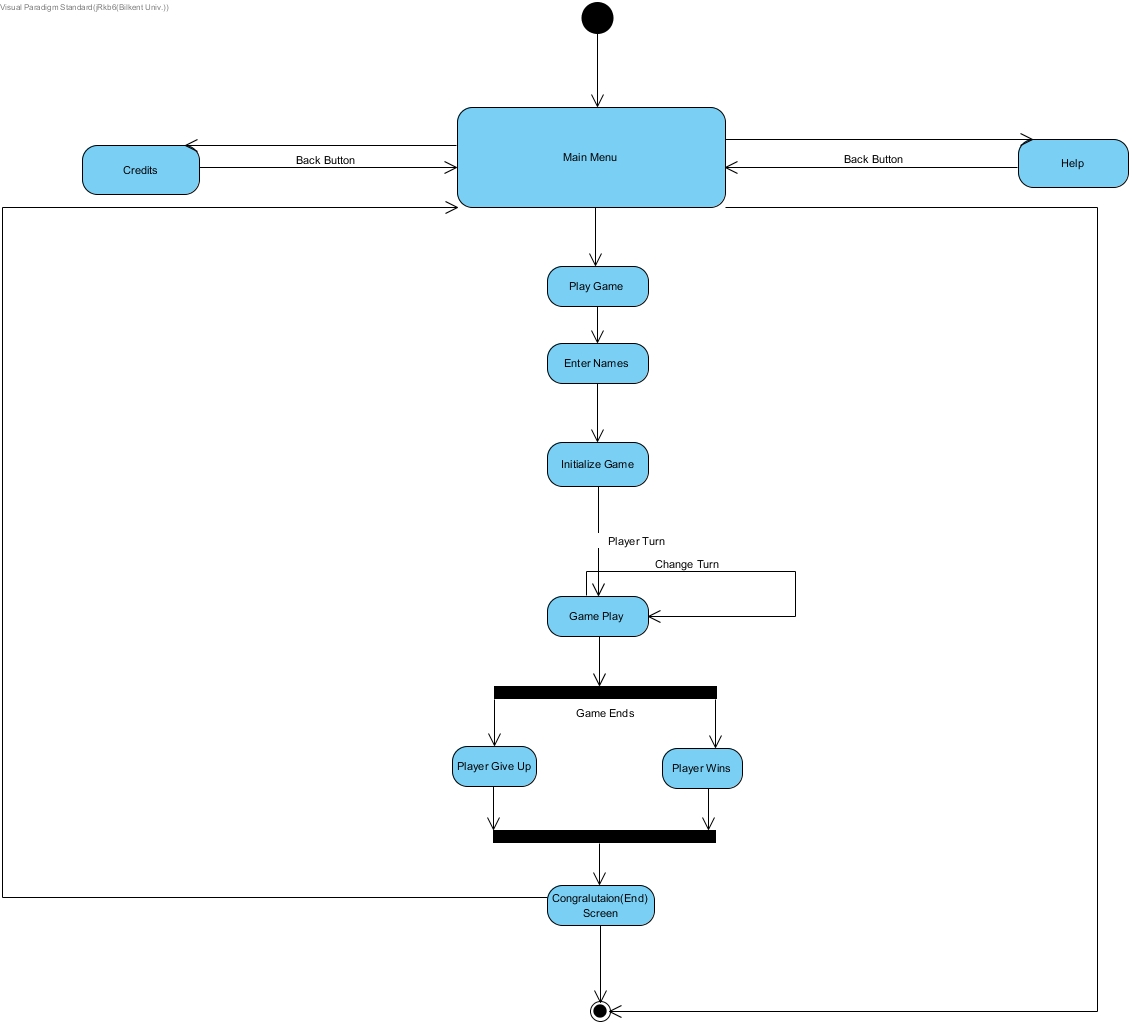
A: If player wants to back to the main menu:

A1: Player selects “Back” button

A2: System displays Main Menu

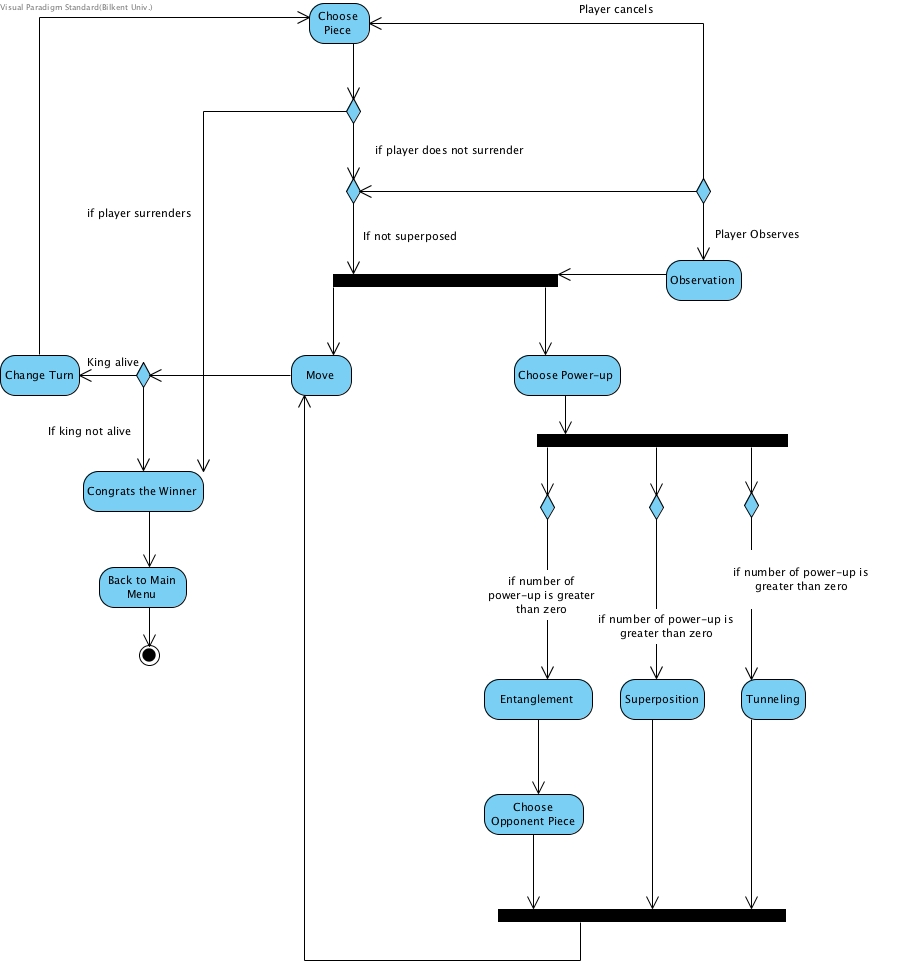
## **4.2 Dynamic Models**

### **4.2.1 Activity Chart**



**Figure 5.2.1.1: Illustrates the activity diagram for the player**

### **4.2.2 State Chart**

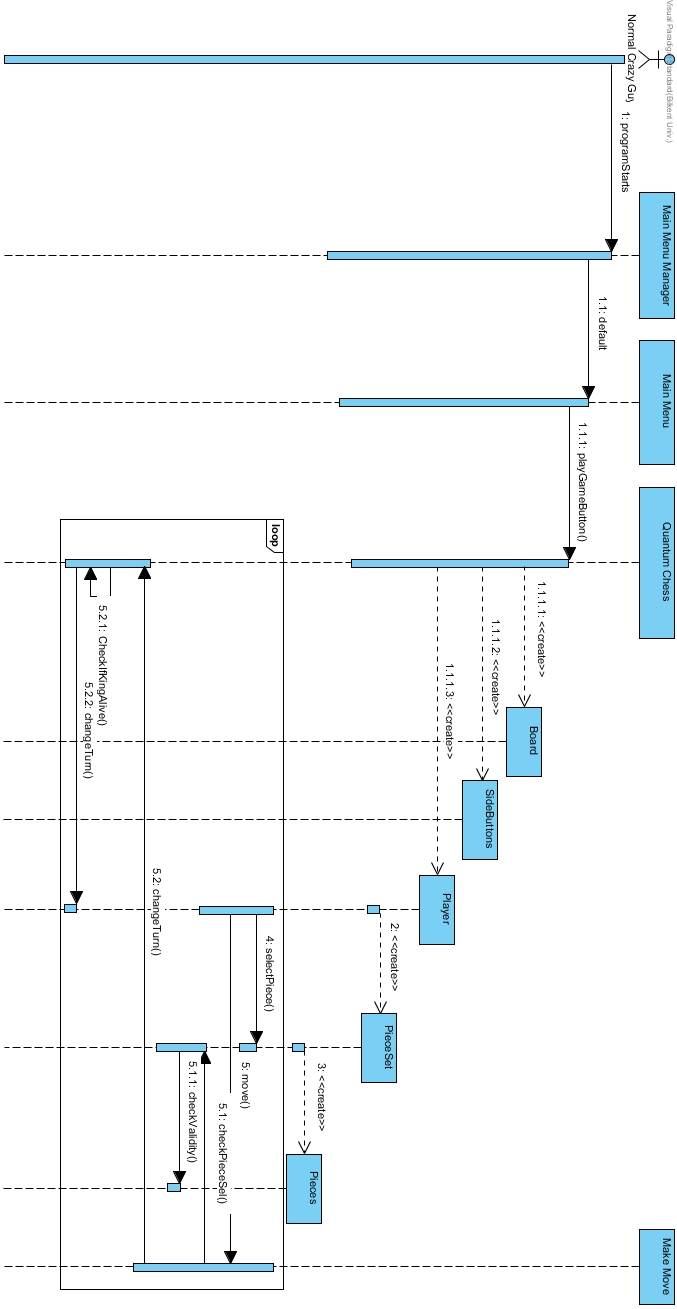


**Figure 5.2.1.2: Illustrates the state chart diagram for the player**

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.

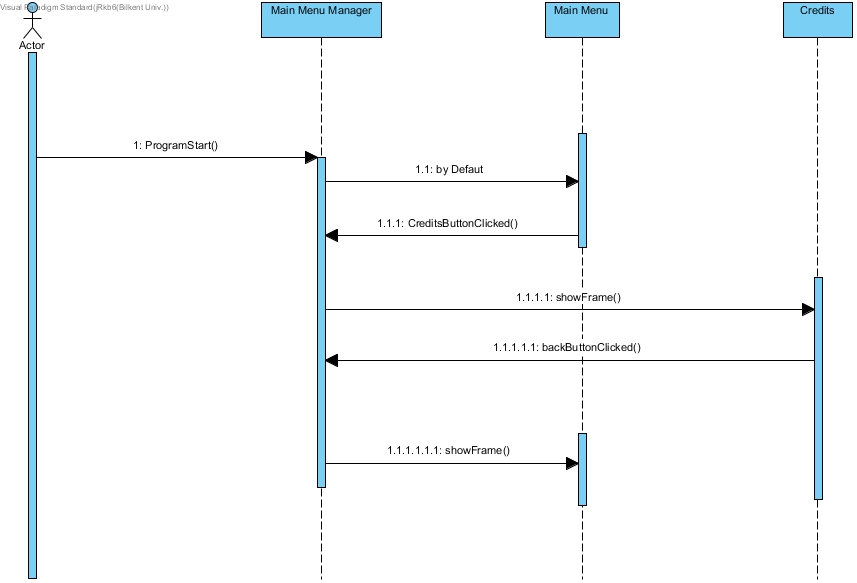
### **4.2.3 Sequence Diagrams**

#### Scenario #1: The General Gamepla**y**



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.

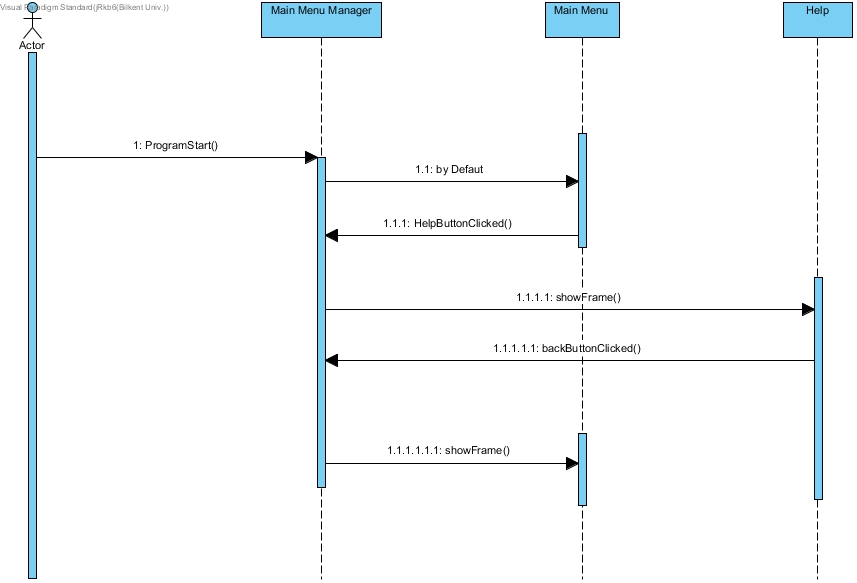
#### **Scenario #2 - Credits:**

****

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

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

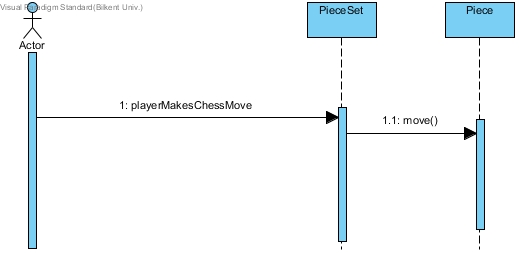
#### **Scenario #3 - Help:**



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

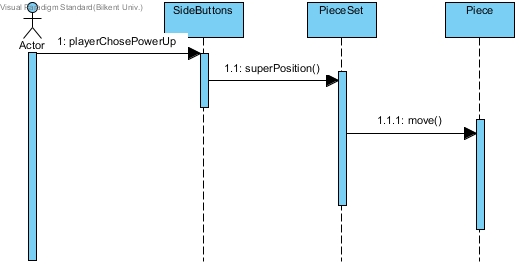
Players want to learn how game is played, after game starts one of them clicks the “Help” button. In help frame, the game will be explained with a tutorial and text. After short information and tutorial, player can go to main menu by pressing back button.

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



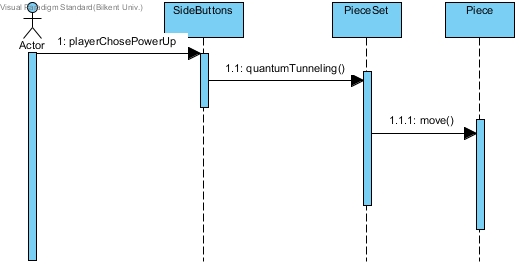
When the player decides to simply to a chess move, he does not select any power up. He just continues with the selection of the place where his piece will be moved if valid (the validity is check is shown in The General Game Play).

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

****

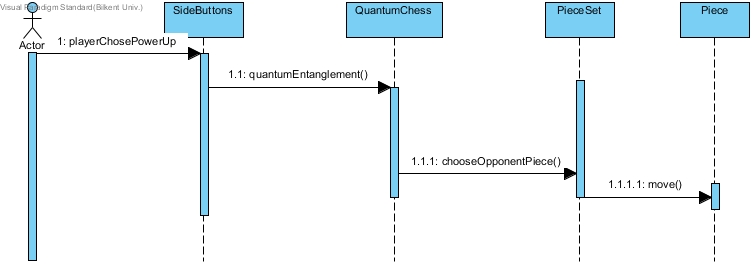
When the player decides to use the Quantum SuperPosition power up. The superPosition button is pressed thus making it possible for the piece to be in super position. The piece will perform the movement if it is valid (the validity check is shown in The General Game Play).

#### **Scenario #7 – Moves Possible: Quantum Tunneling:**



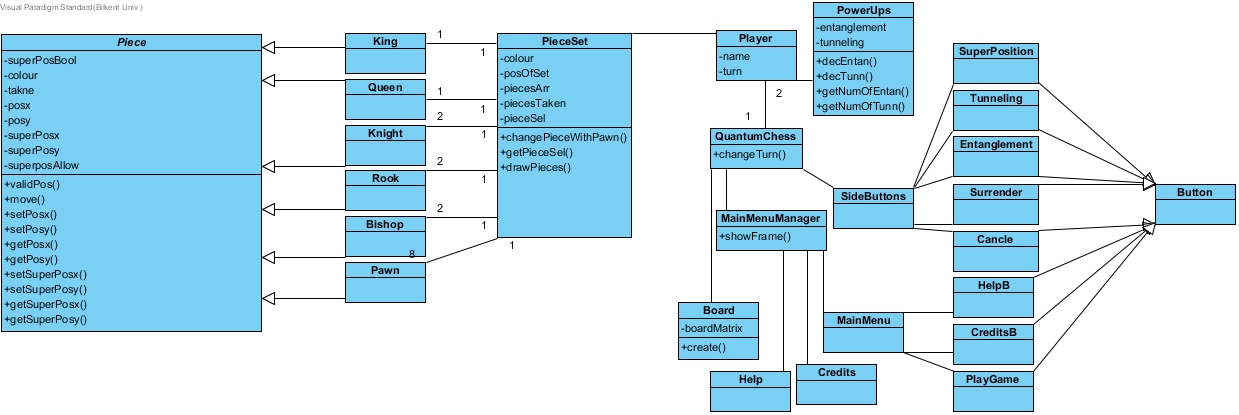
When the player decides to use the Quantum Tunneling power up. The tunneling button is pressed thus making it possible for the piece to move through other pieces. The piece will perform the movement if it is valid (the validity check is shown in The General Game Play).

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



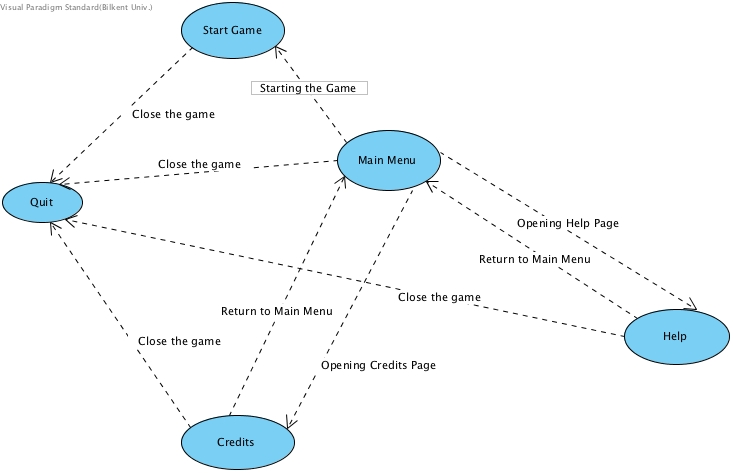
When the player decides to use the Quantum Tunneling power up. The tunneling button is pressed thus making it possible for the piece be entangled with an opponent’s piece. After choosing the power up the player will choose an opponent’s piece and then perform the move if valid (the validity check is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

## C:\Users\Rubini-pc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Object Model.jpg4.3.Object Model



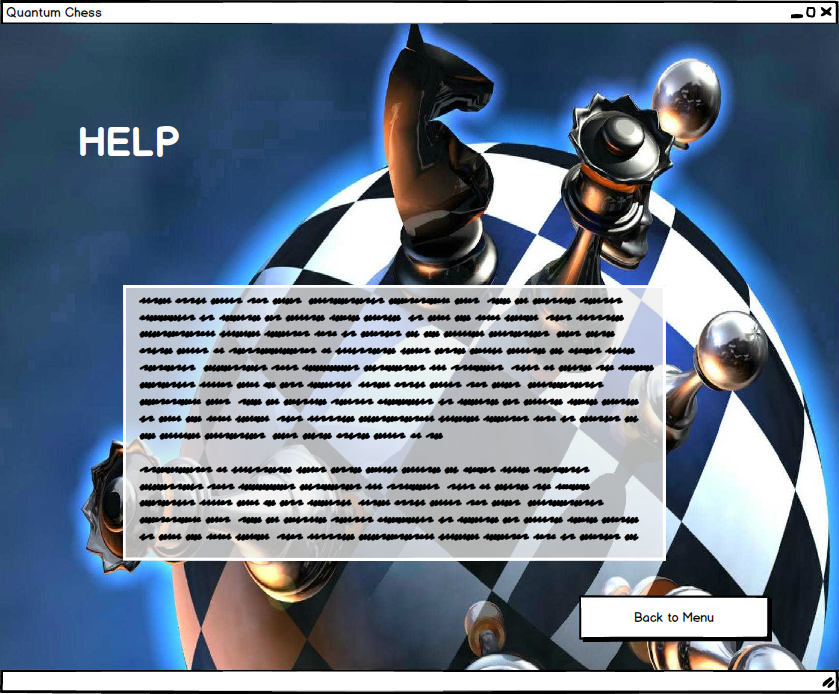
* **Piece:**  Is an abstract class that has the necessary attributes and operations for different chess pieces.
* **King:** Is a class which will extend the Piece class and implement the functions in the appropriate way for the specific piece
* **Queen:** Is a class which will extend the Piece class 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.
* **SuperPosition:** will activate the quantum super position power.
* **Tunneling:** will activate the quantum tunneling power.
* **Entanglement:** will activate the quantum entanglement power.
* **Cancel:** will reset everything like it was the start of the players turn.
* **Surrender:** will give up the game.
* **Back:** will return to the main menu either from the credits or help screen.
* **HelpB:** will send the user to the help frame from main menu.
* **CreditsB:** will send the user to the help frame from the main menu.
* **PlayGame:** will start the game.
* **Button:** is the java library button where all the buttons will inherit from.

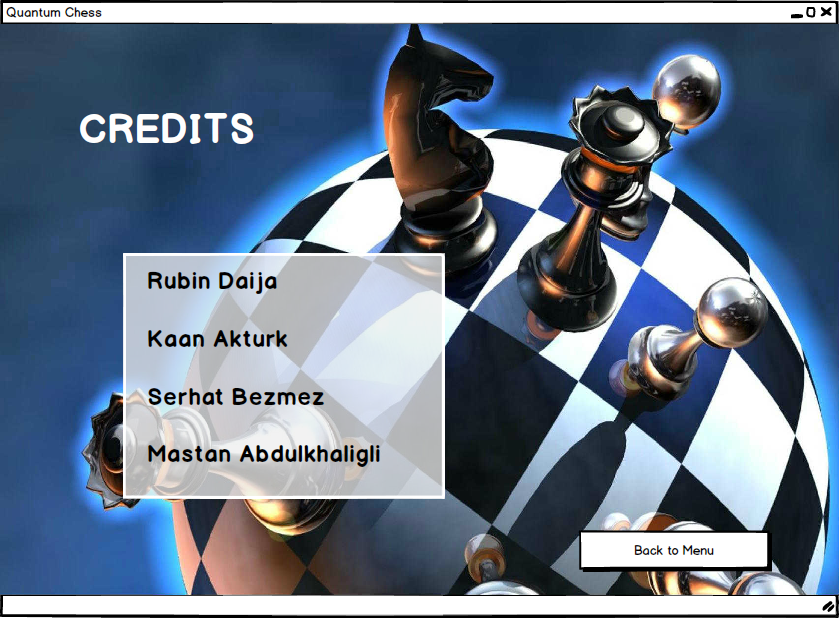
## 4.4.User Interface - Navigational Paths and Screen Mock-ups



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.

****

****

****

****

