**Bilkent University**

Department of Computer Engineering

CS 319  
Object Oriented Software Engineering   
Analysis Report

Quantum Chess

Group 2A

October 5, 2017

**1. Introduction……………………………………………………………..  
2. Overview………………………………………………………………...  
3. Functional requirements……………………………………………….  
4. Nonfunctional requirements…………………………………………..  
5. System models………………………………………………………….  
 5.1. Use case model…………………………………………………..  
 5.2. Dynamic models…………………………………………………  
 5.3. Object and class model…………………………………………  
 5.4. User interface - navigational paths and screen mock-ups…...**

**6. Glossary & references………………………………………………...**

**Analysis Report**

**CS 319 – Quantum Chess**

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.

1. **Game Overview**

Quantum chess has no certain implications. Instead of implications, there are possibilities. Some of the place of pieces are uncertain. Rival can not 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.

* 1. **Power-ups**

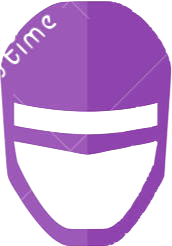
Players have three different power-ups that are quantum superposition, quantum tunneling and quantum entanglement.

* + 1. **Quantum 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.

* + 1. **Quantum 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.

* + 1. **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.

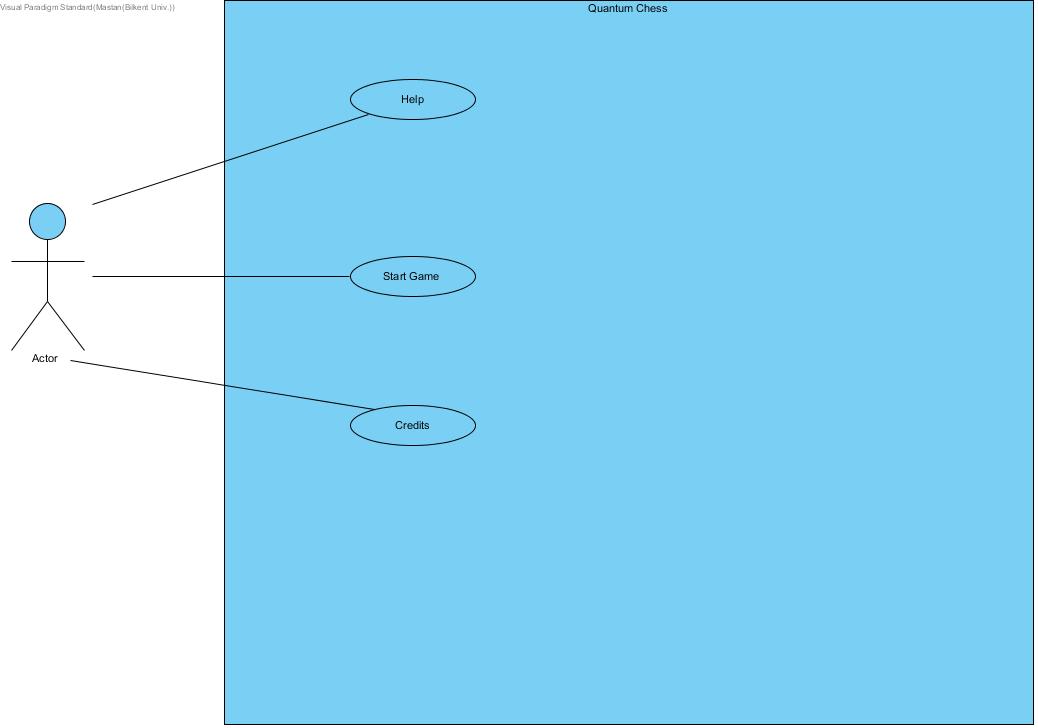
1. **Requirement**
   1. **Functional Requirements**

* Users can start a new game.
* Users 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.
* Users can use power-ups during their turns so that they can challenge the other player or comfort herself/himself.
* Users can press About button so as to comprehend what the rules of the game are.
  1. **Non-Functional Requirements**
* The game does not need installment.
* The requirement memory of the game should be less than 1GB.
* Users can take rival’s piece when the player’s piece moves and encounters the rival’s piece.
  1. **Pseudo Functional Requirements**
* Quantum Chess will be implemented in Java.
* The game should be playable on all operating system.
* The game will not need the Internet.

***5. System models***

***5.1 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**

***5.1.1 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

***5.1.2 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:**

***5.1.3****.* ***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:**

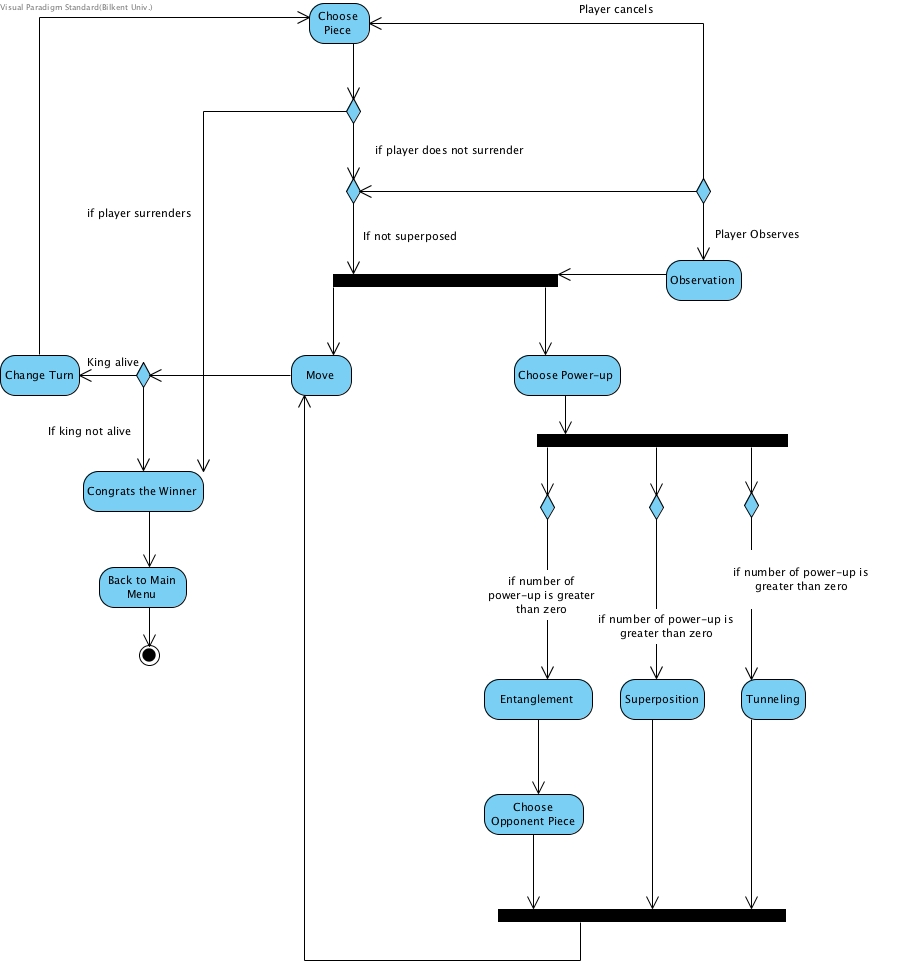
System displays contact information about developers of “Quantum Chess”

**Alternative Flows:**

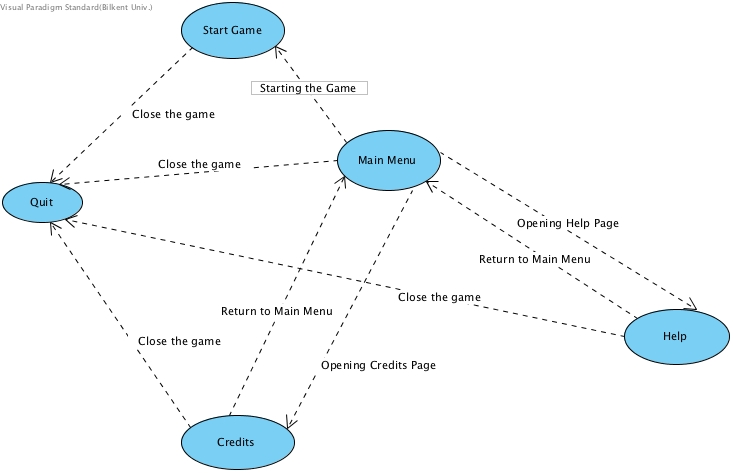
A: If player wants to back to the main menu:

A1: Player selects “Back” button

A2: System displays Main Menu

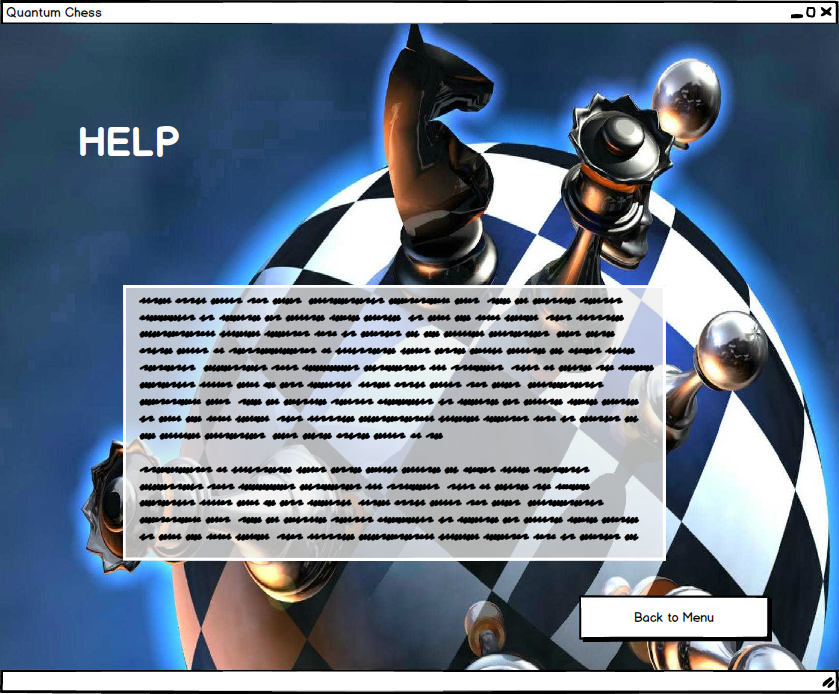


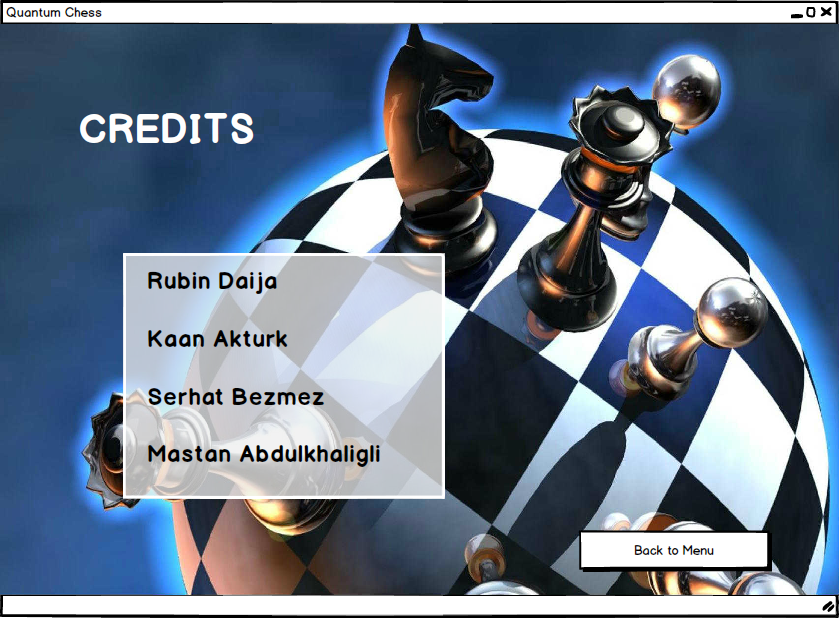
**5.4. User Interface - Navigational Paths and Screen Mock-ups**

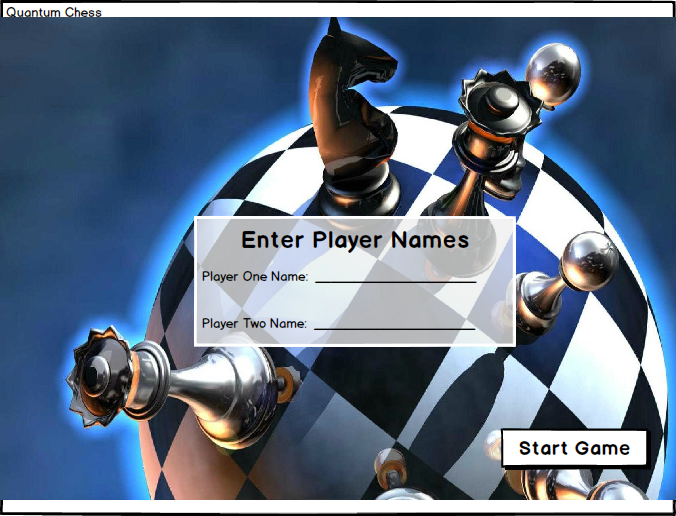


When the program is started, the home page is shown at first. The users have three options that are Start Game, Help and Credits. By pressing Start Game, it is desired from players to write their names. Then, after pressing Start Game button Quantum Chess starts. If they press Help button, it represents 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.

****

****

****

****

