****

**Object Oriented Software Engineering Project**

**CS319 Project: Quantum Chess**

**Design Report**

**Group 2A**

**Rubin Daija**

**Serhat Bezmez**

**Kaan Aktürk**

**Mastan Abdulkhaligli**

**Content**

[1.Introduction 3](#_Toc496341732)

[1.1. Purpose of the System 3](#_Toc496341733)

[1.2. Design Goals 3](#_Toc496341734)

[**1.2.1. Adaptability** 3](#_Toc496341735)

[**1.2.2. Efficiency** 4](#_Toc496341736)

[**1.2.3. Reliability** 4](#_Toc496341737)

[**1.2.4. Usability** 4](#_Toc496341738)

[**1.2.5. Extensibility** 5](#_Toc496341739)

[**1.2.6. Trade-Offs** 5](#_Toc496341740)

[1.3. Definition, Acronyms and Abbreviations 6](#_Toc496341743)

[1.4. References 6](#_Toc496341744)

[1.5. Overview 6](#_Toc496341745)

# 1.Introduction

## 1.1. Purpose of the System

Quantum Chess is sufficient to satisfy needs of the players. Graphics are shown as 2D. The interface of the game is not complicated. Thus, users will avoid confusions and will not waste their time to understand the game interface. It was desired players to gain new experiences and demonstrate them different and surprising dimensions of chess. So, we decided to design developed version of the classic chess game by shaping the rules of chess. It is also believed that the game will attract chess lovers.

## 1.2. Design Goals

In design goals part, it is demonstrated that what we expect and aim from our application. The essential factors that we paid attention during the development of the program are explained below.

### **1.2.1. Adaptability**

The game is developed on Java environment. Even if Java requires extra effort to run GUI packages, it can be playable on major operating systems that contain JVM [1] such as Windows, Linux and Mac OS. We wanted to utilize this advantage of Java. So that, we can prevent people from dealing with operating system compatibility issues and let our game be reached by many people.

### **1.2.2. Efficiency**

The state of being playable is one of the most essential matters for users. Therefore, we direct our attention to efficiency of the game. Power-ups and movements of the objects are designed with considering their level of smoothness. Thus, it is offered users to relish the game. To boost the performance of the game, we optimized our code. It is tried to find most suitable algorithms for our scenario. It is avoided finding tricky ways to implement the code in order not to sacrifice the performance of our program. So that, the burden of the code is lightened.

### **1.2.3. Reliability**

For the unproblematic game pleasure, it is attempted to detect all the blanks in the code. To succeed in finding faults, during the process of developing the program, all of the possible reasons of crash of program are evaluated. Borders of algorithm are investigated and tried to find solutions to get rid of them.

### **1.2.4. Usability**

All the phases of the project, it is aimed to create a friendly interface. Since, we placed the complication into the play game screen not into the user interface. The interface of the application is simple and comprehensible for all users. It contains few buttons. So that, starting and discovering the interface do not require any knowledge and experience. In addition, there is also tutorial that people who are unfamiliar with the game can gain knowledge.

### **1.2.5. Extensibility**

Sometimes, it is important to expand the work done. During the process, new functions, classes, components and API’s [2] are required in order to increase the quality of the algorithms. So, extensibility of Java is utilized.

### **1.2.6. Trade-Offs**

#### **Performance and Memory**

As it is mentioned before, our main purpose is to offer users pure entertainment. To boost the performance, we made concessions to memory. When players select one of the power-up options, our system will require more performance. In addition, the power-ups in the game are most crucial matter in the game. So that, we gave priority to the animation of power-ups and let our memory to increase.

#### **Usability and Functionality**

Generally, players give importance to friendly usability of applications. Since people do not cope with the complications. They desire system with simplest interface. We could not ignore their demand. So, we reduced the functionality of our menu interface.

#### **Efficiency and Reusability**

Our main idea is to create unique chess game. We did not decide to expand our game to another game. We confined our attention to the efficiency of the game. So, we did not consider the reusability of our project.

## 1.3. Definition, Acronyms and Abbreviations

JVM [1] : JAVA Virtual Machine which is an engine that can run JAVA programs. So that, operation systems that contain JVM can start JAVA applications.

API [2] : Application Programming Interface is a set of classes that developed in Java.

## 1.4. References

[1] “Java Virtual Machine (JVM),” *W3schools*, 20-Sep-2017. [Online]. Available: https://www.w3schools.in/java-tutorial/java-virtual-machine/. [Accessed: 18-Oct-2017].

[2] T. Sintes, “Just what is the Java API anyway?,” *JavaWorld*, 13-Jul-2001. [Online]. Available: https://www.javaworld.com/article/2077392/java-se/just-what-is-the-java-api-anyway.html. [Accessed: 18-Oct-2017].

Object-Oriented Software Engineering, Using UML, Patterns, and Java, 3rd Edition*, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2010*

## 1.5. Overview

Our main purpose is to broaden people’s experiences in chess. In this design report, our prospects about the game are displayed. We desire that almost every person can utilize from our game. So, the project is developed on Java Environment. As much as possible, the faults of program are determined. To meet the demands of users, a friendly user interface is created. In addition, there are also some conflicts that we have to deal with. To increase the performance, we extend our limitations of memory. To provide friendly user interface, the functionality of menu page is diminished.