

2D Chess Game Requirements

By: Nicholas Heath and Caleb Gordon



Table of Contents

[Preface 2](#_Toc494053920)

[Document History 2](#_Toc494053921)

[Introduction 3](#_Toc494053922)

[Glossary 4](#_Toc494053923)

[Requirements 5](#_Toc494053924)

[Functional Requirements 5](#_Toc494053925)

[Non-functional Requirements 6](#_Toc494053926)

# Preface

## Document History

|  |  |  |
| --- | --- | --- |
| Version | Date | Summary of Changes |
| 1 | September 25, 2017 | Initial document |
|  |  |  |
|  |  |  |

# Introduction

The purpose of this project is to provide entertainment to people in the form of a customizable game of chess. The chess game will run using a client-server approach and interacts with a database.

# Glossary

* Application – A software program that runs on a computer
* Array – A group of items stored in the computer’s memory
* Byte – A unit of computer information that represents a piece of data
* Client – A piece of hardware or software that accesses a service made available by a server
* Database – A collection of information that is organized so that it can be easily accessed, managed, and updated
* Foreign key – A field in a database table that uniquely identifies a row of another table or the same table
* Functional requirement – A software requirement that states the services that the software should provide and how it behaves based on particular situations and inputs
* Hashing – The transformation of a string of characters into a fixed-length value that represents the original string and cannot be reversed back into the original string
* Kilobyte – A unit of computer information equal to 1024 bytes
* Non-functional requirement – A software requirement that describes the constraints on the services offered by the software
* Relational database – A database structured to recognize relations among stored items of information
* Requirement – A high-level abstract statement of a software or a detailed functional specification
* Resolution – The dimensions of a computer image in pixels
* Server – A piece of hardware or software that provides functionality to its clients
* Session – An interactive information interchange between multiple clients and a server

# Requirements

## Functional Requirements

* The application must be able to support two player online chess game sessions.
  + A “guest” player can connect to a “host” player’s session.
  + All available player sessions hosted by others will be visible in a lobby.
* The chessboard and chess pieces in each game must be customizable.
  + The chessboard’s alternating squares should not be the same color.
  + The chess piece images must be able to be replaced with user custom images stored in the application’s database.
* The application must have an achievement system.
  + Achievements will have different difficulty levels.
  + If time allows, achievements accomplished by one user will be available to others.
* The standard chess game must implement all standard chess rules.
  + A player must be able to capture their opponent’s pieces and remove them from the board.
  + Each type of piece should be able to move across the board in accordance to the rules of chess. A knight should move in an L shape, a bishop can move diagonally, etc.
  + A player’s king piece should be able to be placed in check, checkmate, or stalemate.
    - The game should automatically end in a stalemate under the following conditions:
      * One king vs one king
      * One king and bishop vs one king
      * One king and knight vs one king
      * One king and bishop vs one king and bishop with the bishops on the same color chessboard square
  + The following special moves should be able to be performed:
    - Castling: A player’s king may move two spaces towards one of their rook, and then the rook moves to the square through which the king passed.
    - En passant: A player may capture their opponent’s pawn that moved two spaces forward as if it moved only one space forward.
    - Pawn promotion: When a pawn reaches the other side of the chessboard, the user will be able to change it to a queen, bishop, rook, or knight.
* The application must be able to support chess games with time restrictions.
  + A player will lose the game if a certain amount of time (the total time they have spent on turns) has passed for that player in a timed game.
  + A player will lose the game if they do not make a move in a certain amount of time in a timed-moved game.
  + Any time restrictions are set by the host player.

## Non-functional Requirements

* The application must use a client-server approach.
  + The client side must be created in C# and run using the Windows operating system.
  + The server side must be created in Go and run using the Linux operating system.
  + The application must use a database which is managed by the server side.
  + The server must be accessible by clients via a network connection.
* Users shall be authenticated with a user-created username and password.
  + Passwords must be hashed using the bcrypt password hashing function.
* All user data (achievements, custom chessboard settings, etc.) must be linked to the user via a foreign key in a relational database.
* The custom user images for the chess pieces must be stored in the database as a byte array.
  + The maximum allowable image size must be 500 kilobytes.
  + The resolution for the image must be 60 x 60.
* Timed chess games will use timer objects on the server-side.
  + The maximum time a timed chess game will be limited to is 60 minutes.
  + The minimum allowable time setting for a timed chess game will be 2 minutes.
* Timed-move chess games will use timer objects on the client-side.
  + The maximum time a move in a timed-move chess will be limited to is 60 seconds.
  + The minimum allowable time setting for a move in a timed-move chess game will be 3 seconds.
* There must be a minimum of 10 achievements that can be earned in the achievement system.