

Chess App Spec

By: Anthony Vessa, Brandon Yuan, Yotam Freund

Table of Contents

1) Project Plan	3
1.1) Purpose	3
1.2) Timeline	3
2) Project Description	4
2.1) Project Goals	4
2.2) Constraints and Restrictions	4
3) Project Requirements	4
3.1) Functional Requirements	4
3.2) Wire Frames	4
3.3) Flowchart for Game	7
4) Architecture	8
5) Tech Stack	10

1) Project Plan

1.1) Purpose

The purpose of this project is to deliver a cross platform mobile application capable of providing mobile device users the ability to learn and play chess in an uncomplicated environment.

1.2) Timeline

Date	Goal	Result
	Complete the Main Page and navigation	Successful completion of the Main Page class
	Complete the Model	The Piece and Board classes were completed
	Create and add the logo to the project	The logo was successfully created and displayed in the project
	Create images for all pieces and empty squares	The images were made in Adobe Photoshop
	Finish implementing the view and view model for the game	A first draft was made, but there were many bugs
	Finish debugging the Piece and Board Models	The Piece and Board models were successfully debugged and function
	Finish debugging the Game View and Game View Model Classes	The game was debugged and Chess Master Guru Warrior Legend is functional

2) Project Description

ChessWarrior is a mobile application written in C# and XAML that will allow users to play chess against one another on a single device.

2.1) Project Goals

ChessWarrior will be successful if two players can successfully play against each other in a chess game and store provided personal information and preferences in the settings.

2.2) Constraints and Restrictions

CONSTRAINT	ISSUE/SOLUTION
Delivering Check.	Incorporating a method to determine check in the model.
Copying a 2d array.	Using an imported method from a library.
Moving a pawn forward 2 spaces.	Incorporating a variable determining if the pawn has moved.

3) Project Requirements

3.1) Functional Requirements

Chess Master Guru Warrior Legend must be capable of the following

- Navigating from the main page to Settings and Game pages.
- Storing optionally provided user information in the Settings and displaying it in the Game page.
- Providing functionality for a locally played chess match.

3.2) Wire Frames

- Main Page

Chess Master Guru Warrior Legend

NEW GAME

SETTINGS



- Settings

Profile

Username:

ExampleName01'

Email:

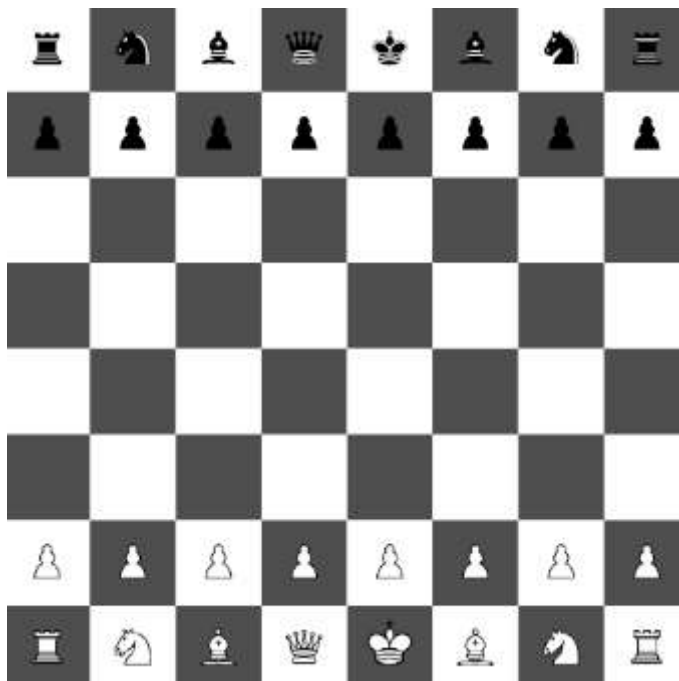
Example@email.com'

Game Settings

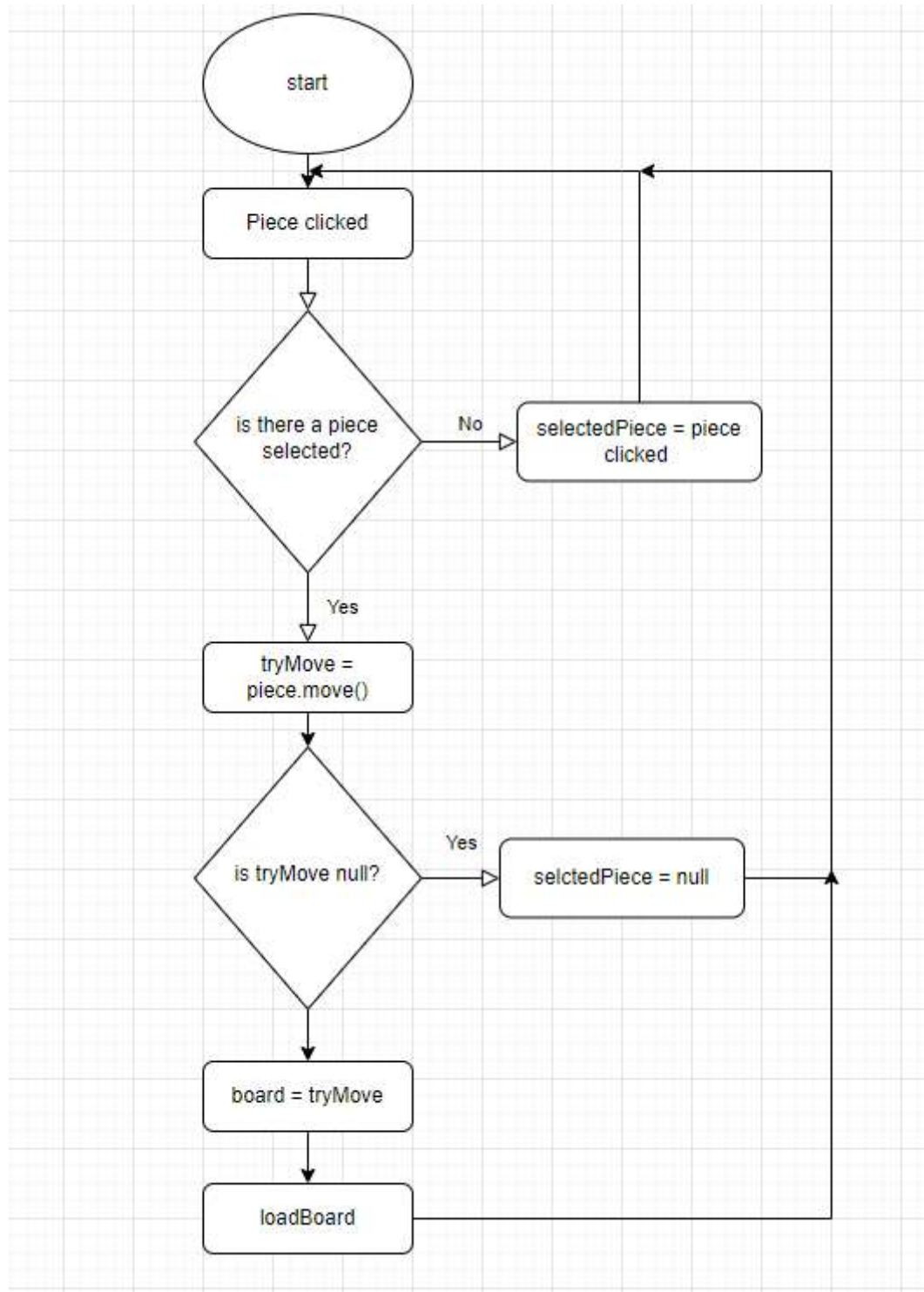
Enable Auto-Queen:

☐

- Game



3.3) Flowchart for Game



4) Architecture

ViewViewModel

1 Main Page

This will be the main/home page for the application and will contain navigation to the start page and settings. It also contains our logo.

1.1 Main Page View Model

This page connects the main page to the model and binds the navigation to the Main Page View.

1.2 Main Page.xaml.cs

This page initiates the class and binding functionality.

2 Game

This page will be the screen where the board and pieces are displayed, and chess games will be played.

2.1 Game View Model

This page connects the main page to the model where all of the functionality for the pieces and board is.

2.2 Game.xaml.cs

This page initiates the class and binding functionality.

3 Settings

Optional information can be stored here and certain settings options are available.

3.1 Settings View Model

This page connects the main page to the model.

3.2 Settings.xaml.cs

This page initiates the class and binding functionality.

4 Base View Model

This class contains the base functionality for much of our code.

Model

1 Piece

This class will contain abstract functionality for a chess piece and all pieces will inherit from it.

1.1 Pawn

This class will contain the functionality for Pawn movement and capturing

1.2 Rook

This class will contain the functionality for Rook movement and capturing.

1.3 Bishop

This class will contain the functionality for Bishop movement and capturing.

1.4 Queen

This class will contain the functionality for the Queen movement and capturing.

1.5 King

This class will contain the functionality for the King movement and capturing.

1.6 Knight

This class will contain the functionality for Knight movement and capturing.

1.7 Empty Square

This class will contain the functionality for determining the color of empty squares.

2 Board

This class will contain the functionality for determining check and instantiating the board.

3 Image

This folder contains all images used for the pieces and the logo.

4 Settings Database

This folder contains the functionality for saving the information stored in the settings page.

5) Tech Stack

Frontend: XAML

Backend: C#