

Institiúid Teicneolaíochta Cheatharlach



INSTITUTE *of*
TECHNOLOGY
CARLOW

At the Heart of South Leinster

Viking Chess Using MCTS

Functional Specification

Declan Murphy – C00106936
Supervisor: Joseph Kehoe
2015

Contents

1. Introduction	2
1.2. Objective	2
1.3. Features	2
2. Project Phases	2
3. Supported Platforms	2
4. Tools & Languages	2
4.1. Languages.....	2
4.2. Tools	2
5. Brief Use Cases	3
5.1. Play New Game	3
5.1.1. <i>Description:</i>	3
5.1.2. <i>Process:</i>	3
5.2. Display Rules Page.....	3
5.2.1. <i>Description:</i>	3
5.2.2. <i>Process:</i>	3
5.3. Display ‘About Page’	3
5.3.1. <i>Description:</i>	3
5.3.2. <i>Process:</i>	3
6. Extended Use Case	4
6.1. Play New Game (VS CPU)	4
6.1.1. <i>Description:</i>	4
6.1.2. <i>Primary Actor</i>	4
6.1.3. <i>Process</i>	4
6.1.4. <i>Alternatives:</i>	4
6.1.5. <i>Use Case Diagram</i>	5
7. Screen Specification	6
7.1. Screen Flowchart.....	6
7.2. Main Page.....	7
7.3. Rules Page	8
7.4. Settings Page.....	9
7.5. Game Page	10
7.6. About Page.....	11

1. Introduction

This document outlines the planned functionality for the Viking Chess application and the planned user interface for the application. The development phases are also outlined. In addition, the supported platforms are also detailed along with the tools and languages used for development. Use cases are specified in brief format with an extended use case for the main 'Play Game' use case.

1.2. Objective

The objective of this project is to implement the ancient game of Viking Chess as a Windows Store application using the Monte Carlo Tree Search algorithm to create a competent A.I player.

1.3. Features

- The application will feature five variants of the Viking Chess (Tafl) board game.
- The application will allow the user to choose to play against another user locally on the same system or against the CPU
- The application will allow the user to view the rules of the game.
- The application will not feature networked play between two players.
- Performance will be a secondary concern to developing a working game and MCTS algorithm.
- The application should pass certification and be releasable onto the Windows Store.

2. Project Phases

The project is split into four phases.

Phase 1: Initial design and development of the application.

Phase 2: Refinement of the application code and UI, preparation for the implementation of the MCTS algorithm

Phase 3: Development begins on the MCTS algorithm, further refinements on existing code and UI.

Phase 4: Finalization of MCTS algorithm and application, testing and debugging.

3. Supported Platforms

The application will be built as a Universal Windows Platform (UWP) application for Windows 10 Store. Efforts will be made to make the application available across the extended family of Windows 10 devices. However, development will be focused upon supporting Windows 10 Desktop at a resolution of 1920x1080 unless time allows.

4. Tools & Languages

The following tools and languages will be used for the development of the application:

4.1. Languages

- C#
- XAML

4.2. Tools

- Microsoft Visual Studio 2015 – Primary development IDE
- Microsoft Blend for Visual Studio 2015 – XAML Design IDE
- Sandcastle Help File Builder – Code Documentation Generator
- Adobe Photoshop CC – Image Creation
- Audacity – Sound Edition

5. Brief Use Cases

5.1. Play New Game

5.1.1. Description:

The Play New Game use case begins when a user presses the new game button from the main menu.

5.1.2. Process:

1. The system loads the 'Game Settings' screen.
2. The user decides if they want to play against a human locally or the CPU.
3. The user decides if they want to play as the attackers or the defenders.
4. The user selects the game mode they want to play from the selection of game modes.
5. The user enters the name for the attacker and the defender.
6. The system loads the specified game page with the selected settings.
7. The game begins with the attacking player moving first.
8. Play rotates between players until one player wins.
9. The use case ends when the game is over.
10. The system returns the user to the main menu.

5.2. Display Rules Page

5.2.1. Description:

The 'Display Rules Page' use case begins when a user presses the 'Rules' button from the main menu.

5.2.2. Process:

1. The system loads the 'Rules' page.
2. The user presses the button for the rule they would like to view.
3. The system loads the animation and associated text for the selected rule.
4. The user can continue to view the selection of rules.
5. The use case ends when the user presses the Back button.
6. The system returns the user to the main menu.

5.3. Display 'About Page'

5.3.1. Description:

The Display 'About Page' use case begins when a user presses the 'About' button from the main menu.

5.3.2. Process:

1. The system loads the 'About page.
2. The use case ends when the user presses the Back button.
3. The system returns the user to the main menu.

6. Extended Use Case

6.1. Play New Game (VS CPU)

6.1.1. Description:

The Play New Game use case begins when a user presses the new game button from the main menu.

6.1.2. Primary Actor

User

6.1.3. Process

1. The user presses the 'New Game' button from the Main Menu.
2. The system loads the 'Settings' page.
3. The user selects to play against the CPU.
4. The user selects to play as the attacker.
5. The user enters the name for the attacker and the defender.
6. The user selects a game mode.
7. The system loads the specified game page with the selected settings.
8. The system starts the game timer and the attacker's turn timer.
9. The user selects one of the attacker's pieces.
10. The system highlights all the squares that the selected piece can move to.
11. The user selects one of the highlighted squares.
12. The system moves the piece to the highlighted square.
13. The system checks if any piece has been captured.
14. The system checks if the game has been won.
15. The system updates the move notation on the UI.
16. The system stops the attacker's timer.
17. The system starts the defender's timer.
18. The system starts the CPU move.
19. The system begins the MCTS algorithm on a new thread for the CPU player.
20. Once the MCTS algorithm completes, the system processes the returned move for the CPU player.
21. The system checks if any piece has been captured.
22. The system checks if the game has been won.
23. The system updates the move notation on the UI.
24. The system stops the defender's timer.
25. The system starts the attacker's timer.
26. Steps 9-22 repeat until the game has been won.

6.1.4. Alternatives:

13a: A piece has been captured:

1. The system removes the piece from the game board.

14a: The game has been won:

1. The system displays a message to the user that the game has been won.
2. The system returns the user to the Main Menu.

6.1.5. Use Case Diagram

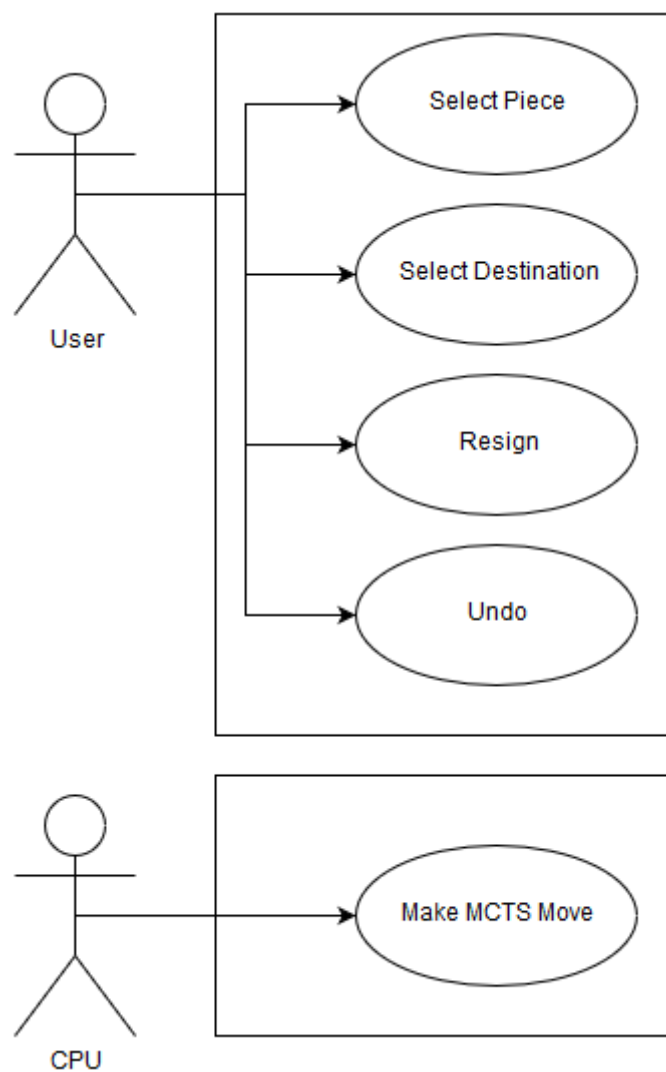


Figure 1 - Simplified Use Case Diagram for Play New Game (VS CPU)

7. Screen Specification

All Screens will be created using XAML (Extensible Application Markup Language), a proprietary XML-based markup language developed by Microsoft. Each screen has a back button that returns the user to the previous screen and a fullscreen button to enter/exit fullscreen with the application.

7.1. Screen Flowchart

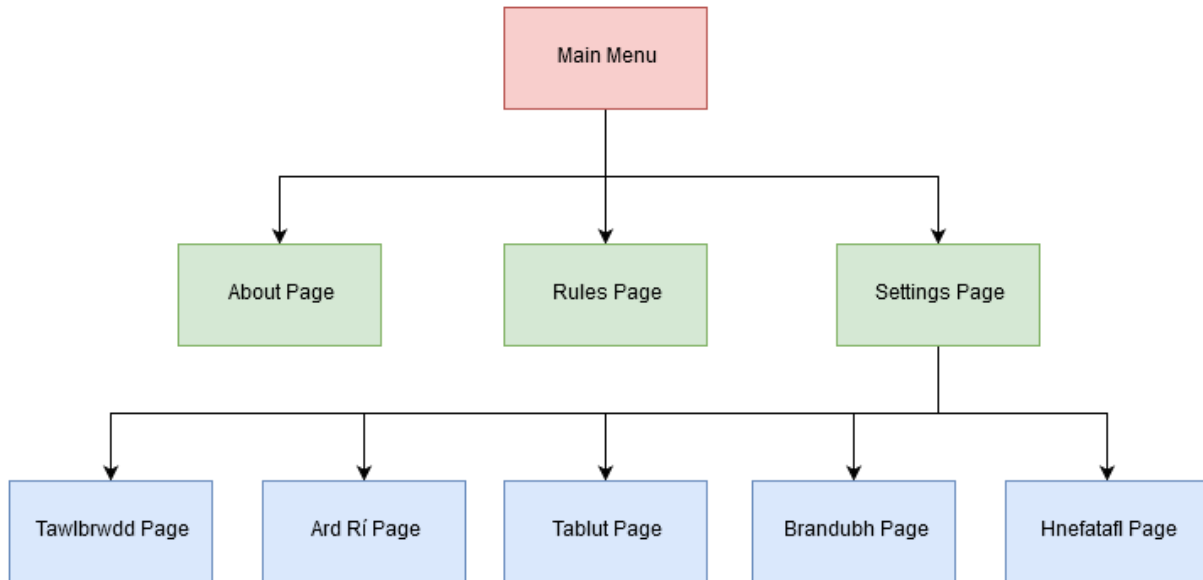


Figure 2 - Screen Layout. All pages can be returned to the previous page.

7.2. Main Page



Figure 3 - Main Menu Mock-Up

The Main Menu contains the game logo and three buttons:

- **New Game** – Starts a new game
- **Rules** – Displays the rules of the game
- **About** – Displays details about the application and the developer.

7.3. Rules Page

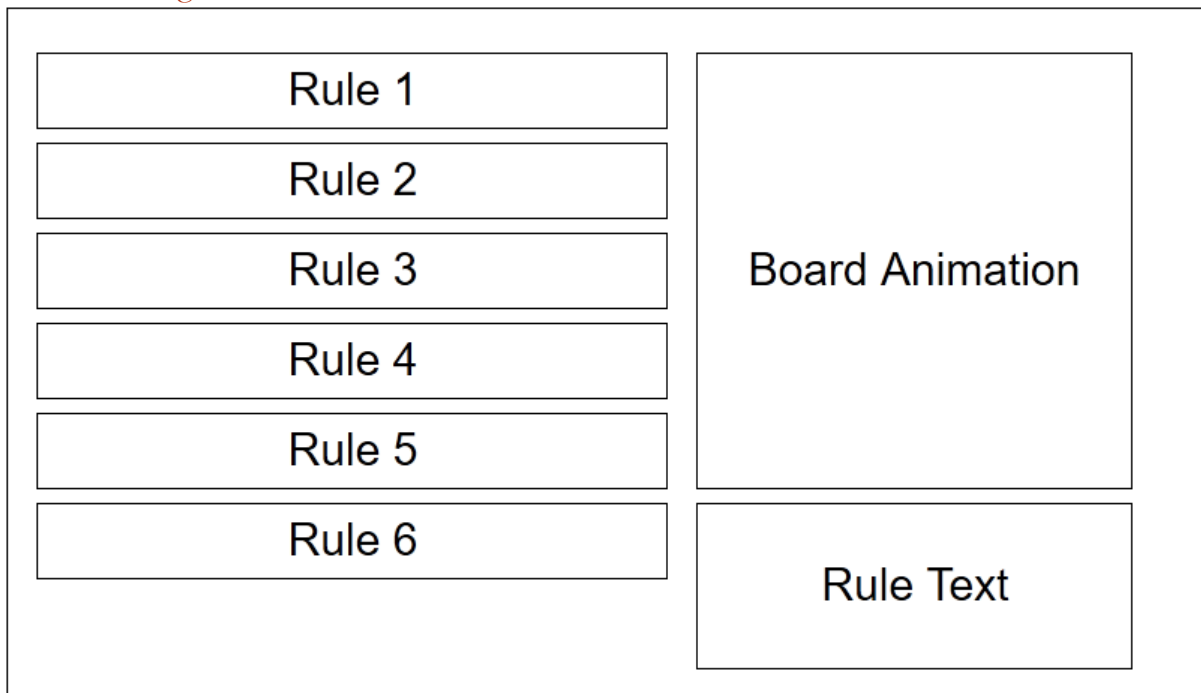


Figure 4 - Rules Page Mock-Up

The Rules Page contains several buttons for the different rules of the game. When a button is pressed, an animation will begin in the **Board Animation** window visualizing the selected rule. The contents of the **Rule Text box** will also be updated.

7.4. Settings Page

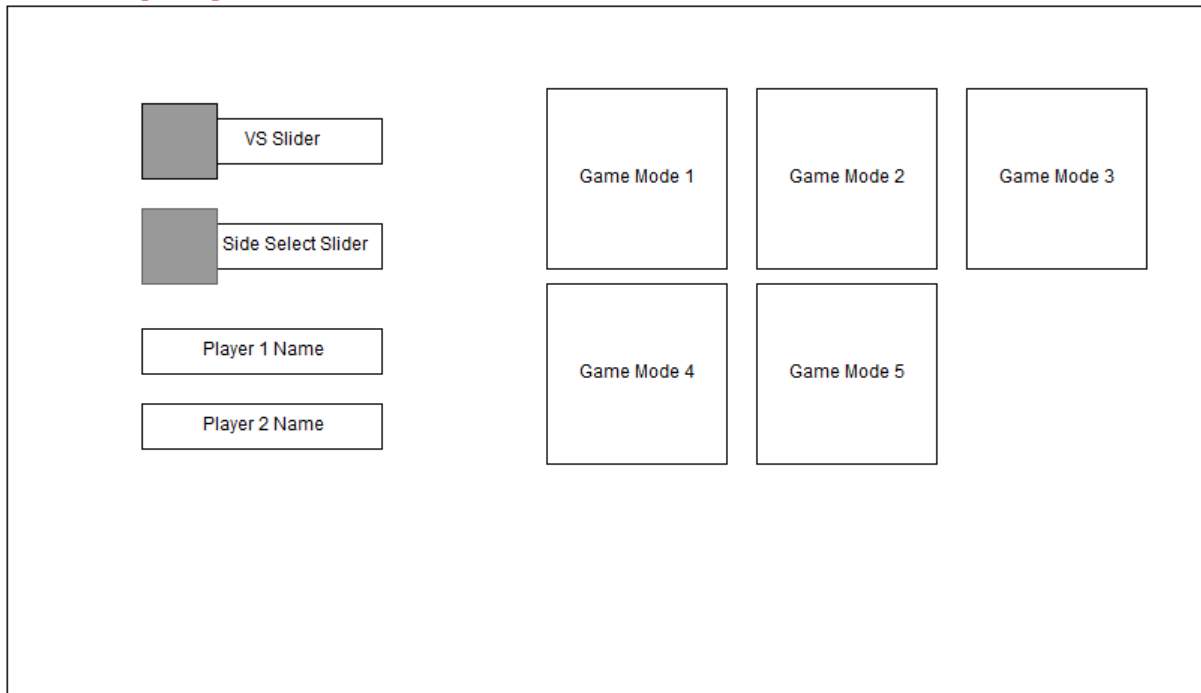


Figure 5 - Settings Page Mock-Up

The Settings Page contains:

- **VS Slider** – Sets whether to play against another human or the CPU.
- **Side Select Slider** – Sets which side the human player controls (only relevant when playing against the CPU).
- **Text Boxes** - Sets the display name for the Attacker and the Defender.
- **Game Mode Buttons** - Starts a game of the specified game mode with the selected settings.

7.5. Game Page

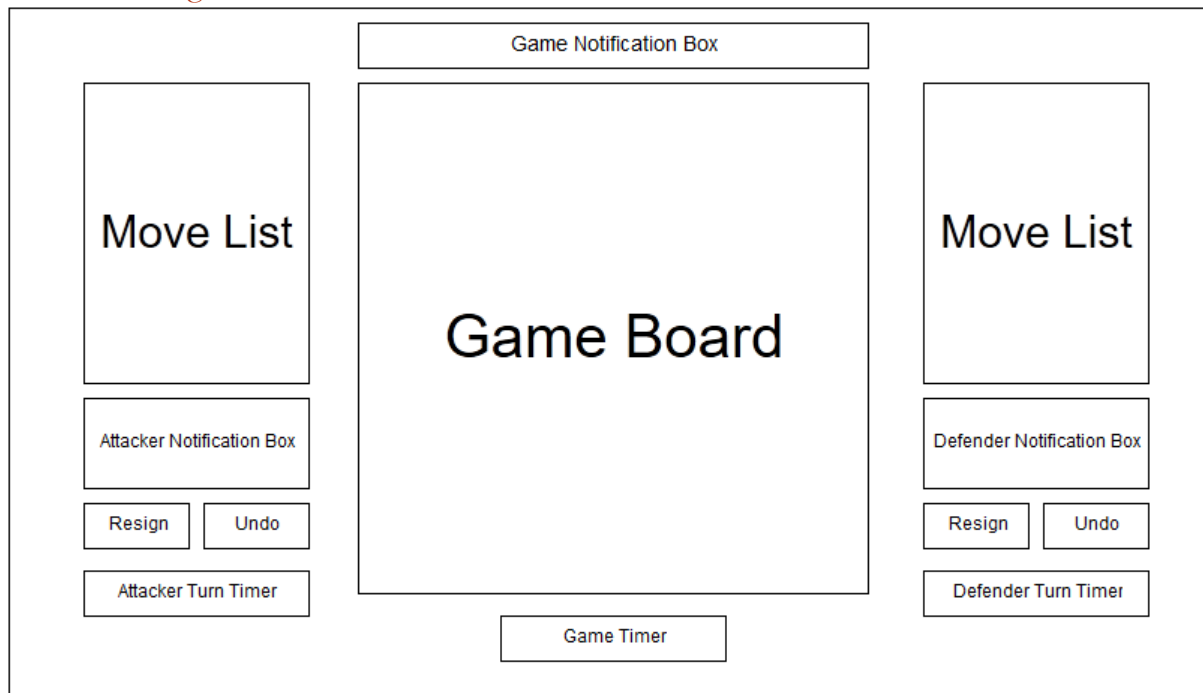


Figure 6 - Game Page Mock-Up

Each Game Page regardless of the Game Mode has the same layout.

- **Game Board** – Displays the game board and pieces for the selected game mode.
- **Move List** – Displays the moves made by each player in text notation e.g. C4-E4.
- **Notification Boxes** – Displays the name for each player and notifies each one of their turn.
- **Resign Buttons** – Forfeits the game for a player.
- **Undo Buttons** – Undoes the last move made.
- **Turn Timers** – Displays the time elapsed for each player per turn.
- **Game Timer** – Displays the total time elapsed over the course of the game.
- **Game Notification Box** – Displays Game notifications such as when the King is in Check or the CPU is processing their move.

7.6. About Page



Figure 7 - About Page Mock-Up

The About Page displays information about the game and the developer.