# ****Test Plan****

Test Plan Identifier: WPDraughts-TPv1

### Introduction

Since this is project is being created by a single developer all testing must be conducted by the said developer. The testing will be carried out in three phases.

* Automated Testing
* Manual Testing
* UX Testing

### Software Risk Issues

* Unreliable connection to the X-Drive on the college network.
* Outdated Windows Phone SDK on college computers.
* Windows Phone project type does not support unit testing.

### Features To Be Tested

The following list describes the features to be tested.

* All In game menus
* The ability to select a game piece.
* The ability to see possible moves.
* The ability to make a move.
* The ability to jump and remove an opponent’s game piece
* The ability to jump and remove multiple opponent pieces in one turn.
* The ability to change a pawn piece to a king piece.
* The ability to reach an end game state.

## TEST POLICY

### Mission Statement

It is the responsibility that the developer will test all software to ensure that it meets all the objectives and requirements required.

### Testing Approach

Testing will be done at all stages of the development process at the discretion of the developer.

Both manual and automated testing techniques will be used, appropriately.

The final project submission must include a comprehensive suite of both manual and automated tests.

## Test Strategy

The test strategy for this project will consist of Unit testing, Manual testing and UX testing. As this is an academic project and not a commercial product with a project owner to satisfy acceptance testing will come in the form of grading by the project supervisors.

### Unit testing

The classes that comprise the draught game model will be accompanied by a bank of automated tests. Since these classes represent the core of the application it is imperative that the behaviour of the methods in these classes matches the expected behaviour. Since these classes are designed to be reused maintaining a library of unit tests is worth the time and effort involved. Unit tests should be rerun after any code changes and new tests should be created if any new functionality is added to the game model.

### Manual Testing

This will be undertaken by the developer and will be concerned with testing the frontend of the application. By laying out steps to be taken and the expected results I can ascertain if the application is preforming correctly. These tests may also indicate errors in the game model and highlight a need for additional unit testing. These tests may also highlight extra functionality needed to implement the application. If this is the case the required code and relevant unit test should be added to the game model.

### UX Testing

This will be accomplished by having test users fill out a pre-test questionnaire, use the application and then fill out a post-test questionnaire. Given the straight forward interactions required between user and application I do not believe that recording of sessions is necessary. The UX test must take place on an actual windows 7 phone. The emulator software that is supplied with Visual Studio 2010 is not acceptable.

### Acceptance testing

Since this project is being created as part of an academic exercise it is the project supervisors that will be responsible for acceptance testing. This is will be performed at the end of the project development. The acceptance test results will be in the form of my overall grade and will take into account a number of factors other than the functionality of the project i.e. algorithm complexity’s, coding quality, use of testing techniques, use of design patterns and associated documentation including this test plan.

## Test Tools

* All editing, compiling and debugging will be done in Visual Studio 2010.
* The unit testing features of Visual Studio will used for unit testing.
* The Windows Phone emulator may be used for manual testing.
* New testing tools may be considered as they are encountered during the academic year.

## Meetings

A meeting with a project supervisor will take place each Tuesday from 4 – 6 pm. The objective of these meetings is to ensure continuous progress on the project. The supervisor may request to see evidence of testing and its effect on the development process.

## Item Pass/Fail Criteria

The test process will be finished once all development has ceased and all unit tests of critical functions have been to completed with 100% success and manual tests have been completed to an acceptable level(to be determined by developer).

## Suspension Criteria and Resumption Requirements

If unit testing of critical functions fails then a serious effort should be made to correct or find a suitable work around of these errors. It is at the discretion of the developer how much time should be given over to this and when to continue further work on the project. It may be decided that the best course of action is to proceed with development and produce an imperfect application as opposed to an unfinished one. Testing may resume if errors are corrected or suitable work around are implemented. Any decisions made in this regard must be made with the following in mind “what course of action should guarantee the best final grade”.

## Environmental Needs

The following elements are required to support the overall testing effort.

* Access to a computer.
* Access to Visual Studio 2010.
* Access to Windows Phone 7.5 (Mango) SDK.
* Access to a smart phone running Windows Phone 7.5 OS.
* Access to a suitable word editor for testing documentation.

## Training needs

The tester must have knowledge of C#.

The tester must know how to create unit tests and understand there reports.

The tester must have basic word processing skills.

The tester must be competent in the use of smart phones preferably a Windows 7 smartphone.

## Planning Risks and Contingencies

Considering that risk identification and management are the main concerns in every software project, the types of planning risks and contingencies which the project could encounter could be.

### Schedule Risk

• Wrong time estimation within the development team.

• Failure to identify complex functionalities of the project and time required to develop these functionalities.

• Unexpected project scope expansions.

### Technical risks

* Loss of internet access.
* Hardware failure (phone, computer)

# Unit Test Cases

Unit test will use either the Initial Board configuration fig.1 or a mid-game Test Board laid out as shown fig.2

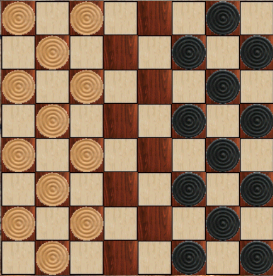
 

Fig 1

Fig 2

**Test Name: Evaluate Board for White Player Using Initial Board**

**Description**

* Test Board = Initial Board
* Player Colour =White

**Expected Result**

* Type = int
* Value = 12

**Test Name: Evaluate Board for Black Player Using Initial Board**

**Description**

* Test Board = Initial Board
* Player Colour = Black

**Expected Result**

* Type = int
* Value = 12

**Test Name: Evaluate Board for White Player Using Test Board**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White

**Expected Result**

* Type = int
* Value = 16

**Test Name: Evaluate Board for Black Player Using Test Board**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White

**Expected Result**

* Type = int
* Value = 18

**Test Name: Get All Possible Jumps for Normal White Piece Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 7

**Expected Result**

* Type = int (number of moves returned)
* Value = 2

**Test Name: Get All Possible Jumps for Normal Black Piece Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 8

**Expected Result**

* Type = int (number of moves returned)
* Value = 2

**Test Name: Get All Possible Jumps for King Piece Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 1

**Expected Result**

* Type = int (number of moves returned)
* Value = 2

**Test Name: Get All Possible Moves for White Piece Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 3

**Expected Result**

* Type = int (number of moves returned)
* Value = 2

**Test Name: Get All Possible Moves for Black Piece Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 9

**Expected Result**

* Type = int (number of moves returned)
* Value = 2

**Test Name: Get All Possible Moves for King Piece Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 5

**Expected Result**

* Type = int (number of moves returned)
* Value = 4

**Test Name: Get All Possible Moves for White Player Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White

**Expected Result**

* Type = int (number of moves returned)
* Value = 12

**Test Name: Get All Possible Moves for Black Player Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White

**Expected Result**

* Type = int (number of moves returned)
* Value = 14

**Test Name: Get All Possible Moves for White Piece with Forced Jumps Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 6

**Expected Result**

* Type = int (number of moves returned)
* Value = 1

**Test Name: Get All Possible Moves for Black Piece with Forced Jumps Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 9

**Expected Result**

* Type = int (number of moves returned)
* Value = 1

**Test Name: Get All Possible Moves for King Piece with Forced Jumps Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 2

**Expected Result**

* Type = int (number of moves returned)
* Value = 1

**Test Name: Get All Possible Moves for White Player with Forced Jumps Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White

**Expected Result**

* Type = int (number of moves returned)
* Value = 3

**Test Name: Get All Possible Moves for Black Player with Forced Jumps Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black

**Expected Result**

* Type = int (number of moves returned)
* Value = 6

**Test Name: Get Opponent Colour for White Piece Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 3

**Expected Result**

* Type = PlayerColour
* Value = Black

**Test Name: Get Opponent Colour for Black Piece Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 6

**Expected Result**

* Type = PlayerColour
* Value = White

**Test Name: Get Possible Boards for White Player Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White

**Expected Result**

* Type = int (number of boards returned)
* Value = 12

**Test Name: Get Possible Boards for Black Player Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black

**Expected Result**

* Type = int (number of boards returned)
* Value = 14

**Test Name: Check That Movable Piece Is Selected When It Should Be**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 7

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Check That Movable Piece Is Selected When It Should Not Be**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 3

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Check That Movable Piece Is Selected When It Should Be No Forced Jumps**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 3

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Game over When White Player Should Have Won Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Game over When Black Player Should Have Won Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Move Legal for White Piece Moving To the Top Right Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 3
* Draught Piece move from square B2 to C1

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Move Legal for White Piece Moving To the Bottom Right Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 3
* Draught Piece move from square B2 to C3

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Move Legal for Black Piece Moving To the Top Left Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 10
* Draught Piece move from square E7 to D6

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Move Legal for Black Piece Moving To the Bottom Left Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 9
* Draught Piece move from square E5 to D6

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Move Legal for White Piece Jumping To the Top Right Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 7
* Draught Piece move from square D4 to F2

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Move Legal for White Piece Jumping To the Bottom Right Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 7
* Draught Piece move from square D4 to F6

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Move Legal for Black Piece Jumping To the Top Left Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 8
* Draught Piece move from square E3 to C1

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Move Legal for Black Piece Jumping To the Bottom Left Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 8
* Draught Piece move from square E3 to C5

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Move Illegal for White Piece Moving To the Top Left Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 6
* Draught Piece move from square D2 to C1

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Is Move Illegal for White Piece Moving To the Bottom Left Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 6
* Draught Piece move from square D2 to C3

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Is Move Illegal for Black Piece Moving To the Top Right Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 8
* Draught Piece move from square E3 to F2

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Is Move Illegal for Black Piece Moving To the Bottom Right Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 8
* Draught Piece move from square E3 to F4

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Is New Location an Empty Square When an Empty Square Is Selected Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 3
* Draught Piece move from square B2 to C1

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is New Location an Empty Square When an Empty Square Is Not Selected Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 6
* Draught Piece move from square D2 to E3

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Is New Location Legal for Normal White Piece When It Should Be Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 3
* Draught Piece move from square B2 to C1

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is New Location Legal for Normal Black Piece When It Should Be Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 9
* Draught Piece move from square E5 to D6

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is New Location Legal for King Piece When It Should Be Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 11
* Draught Piece move from square H4 to G3

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is New Location Legal for Normal White Piece When It Should Not Be Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 3
* Draught Piece move from square B2 to A1

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Is New Location Legal for Normal Black Piece When It Should Not Be Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 8
* Draught Piece move from square E3 to F1

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Is New Location Legal for King Piece When It Should Not Be Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 1
* Draught Piece move from square A3 to B3

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Is Opponent Piece in Square to the Top Right Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 1
* Draught Piece move from square A3 to B2

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Opponent Piece in Square to the Bottom Right Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 1
* Draught Piece move from square A3 to B4

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Opponent Piece in Square to the Top Left Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 8
* Draught Piece move from square E3 to D2

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Opponent Piece in Square to the Bottom Left Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 8
* Draught Piece move from square E3 to D4

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Opponent Piece Not In Square to the Top Right Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 3
* Draught Piece move from square B2 to C1

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Is Opponent Piece Not In Square to the Bottom Right Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 3
* Draught Piece move from square B2 to C3

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Is Opponent Piece Not In Square to the Top Left Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 11
* Draught Piece move from square H4 to G3

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Is Opponent Piece Not In Square to the Bottom Left Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 11
* Draught Piece move from square H4 to G5

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Is Piece in This Square When There Is a Piece Test**

**Description**

* Test Board = Mid Game Test Board
* Test Square = B2

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is Piece in This Square When There Is No Piece Test**

**Description**

* Test Board = Mid Game Test Board
* Test Square = A1

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Is Black Piece to Be Crowned When It Should Be Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 1

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is White Piece to Be Crowned When It Should Be Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 11

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is White Piece to Be Crowned When It Should Not Be Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = White
* Draught Piece = test piece 3

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Is Black Piece to Be Crowned When It Should Not Be Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = black
* Draught Piece = test piece 8

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Is There a Second Jump When There Is Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 9
* Draught Piece move from square E5 to C3

**Expected Result**

* Type = Boolean
* Value = true

**Test Name: Is There a Second Jump When There Is Not Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 1
* Draught Piece move from square A3 to C1

**Expected Result**

* Type = Boolean
* Value = false

**Test Name: Remove Jumped Piece Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 1
* Draught Piece move from square A3 to C1

**Expected Result**

* Type = DraughtPiece
* Value = null

**Test Name: Replace Jumped Piece Test**

**Description**

* Test Board = Mid Game Test Board
* Player Colour = Black
* Draught Piece = test piece 1
* Draught Piece move from square A3 to C1

**Expected Result**

* Type = DraughtPiece
* Value = White piece in B2

**Test Name: Set the Game Board Test**

**Description**

* Test Board = Initial Board

**Expected Result**

* Type = string
* Value = "\nw\_w\_w\_w\_\n\_w\_w\_w\_w\nw\_w\_w\_w\_\n\_\_\_\_\_\_\_\_\n\_\_\_\_\_\_\_\_\n\_b\_b\_b\_b\nb\_b\_b\_b\_\n\_b\_b\_b\_b"

**Test Name: Apply Normal White Piece Move Test**

**Description**

* Test Board = Mid Game Test Board
* Draught Piece move from square B2 to C1

**Expected Result**

* Type = string
* Value = "\n\_\_B\_B\_\_\_\n\_\_\_w\_\_\_\_\nw\_\_\_\_\_B\_\n\_w\_w\_\_\_\_\n\_\_b\_b\_b\_\n\_\_\_\_\_\_\_\_\n\_\_\_\_\_\_\_\_\n\_\_\_W\_W\_\_"

**Test Name: Apply Normal Black Piece Move Test**

**Description**

* Test Board = Mid Game Test Board
* Draught Piece move from square A5 to B6

**Expected Result**

* Type = string
* Value = "\n\_\_B\_\_\_\_\_\n\_w\_w\_B\_\_\n\_\_\_\_\_\_B\_\n\_w\_w\_\_\_\_\n\_\_b\_b\_b\_\n\_\_\_\_\_\_\_\_\n\_\_\_\_\_\_\_\_\n\_\_\_W\_W\_\_"

**Test Name: Undo Move Test**

**Description**

* Test Board = Mid Game Test Board
* Draught Piece move from square B2 to C1

**Expected Result**

* Type = string
* Value = "\n\_\_B\_B\_\_\_\n\_w\_w\_\_\_\_\n\_\_\_\_\_\_B\_\n\_w\_w\_\_\_\_\n\_\_b\_b\_b\_\n\_\_\_\_\_\_\_\_\n\_\_\_\_\_\_\_\_\n\_\_\_W\_W\_\_"

**Test Name: Is Depth Reached When It Has Been Test**

**Description**

* Player = AI player
* Number of Moves Ahead = 4
* Current depth = 4

**Expected Result**

* Type = bool
* Value = true

**Test Name: Is Depth Reached When It Has Not Been Test**

**Description**

* Player = AI player
* Number of Moves Ahead = 4
* Current depth = 2

**Expected Result**

* Type = bool
* Value = false

**Test Name: Set Game Difficulty to Easy Test**

**Description**

* Player = AI player
* Difficulty = easy

**Expected Result**

* Type = int (number of moves Ahead AI will look)
* Value = 2

**Test Name: Set Game Difficulty to Normal Test**

**Description**

* Player = AI player
* Difficulty = normal

**Expected Result**

* Type = int (number of moves Ahead AI will look)
* Value = 4

**Test Name: Set Game Difficulty to Hard Test**

**Description**

* Player = AI player
* Difficulty = hard

**Expected Result**

* Type = int (number of moves Ahead AI will look)
* Value = 8

**Test Name: Min Max Test**

**Description**

* Player = AI player
* Test Board = Mid Game Test Board
* Number of Moves Ahead = 2
* Current depth = 4

**Expected Result**

* Type = string
* Value = "\nB\_B\_B\_\_\_\n\_\_\_w\_\_\_\_\n\_\_\_\_\_\_B\_\n\_w\_\_\_\_\_\_\n\_\_b\_\_\_b\_\n\_\_\_\_\_\_\_\_\n\_\_\_\_\_\_\_\_\n\_\_\_W\_W\_\_"

### Detailed Walkthrough of the Min Max Test

Test Board Possible moves for AI (MAX Player)

1. Piece 1 jump to C1
2. Piece 1 jump to C5
3. Piece 2 double jump to A1
4. Piece 2 double jump to E1
5. Piece 2 move to B6
6. Piece 5 move to B6
7. Piece 5 move to B8
8. Piece 5 move to D6
9. Piece 5 move to D8
10. Piece 8 jump to C1
11. Piece 8 jump to C5
12. Piece 9 double jump to A1
13. Piece 9 move to D6

White counter move (MIN Player) to black player move (MAX player) 1 (Piece 1 jump to C1)

1. Piece 4 move to C3 => new Game Board 1 => Evaluated Value = 19
2. Piece 4 move to C5 => new Game Board 2 => Evaluated Value = 19
3. Piece 6 move to E1 => new Game Board 3 => Evaluated Value = 19
4. Piece 7 jump to F2 => new Game Board 4 => Evaluated Value = 18
5. Piece 7 jump to F6 => new Game Board 5 => Evaluated Value = 18
6. Piece 11 move to G3 => new Game Board 6 => Evaluated Value = 19
7. Piece 11 move to G5 => new Game Board 7 => Evaluated Value = 19
8. Piece 12 move to G5 => new Game Board 8 => Evaluated Value = 19
9. Piece 12 move to G7 => new Game Board 9 => Evaluated Value = 19

White counter move (MIN Player) to black player move (MAX player) 2 (Piece 1 jump to C5)

1. Piece 3 move to C1 => new Game Board 10 => Evaluated Value = 19
2. Piece 3 move to C3 => new Game Board 11 => Evaluated Value = 19
3. Piece 6 move to E1 => new Game Board 12 => Evaluated Value = 19
4. Piece 7 jump to F2 => new Game Board 13 => Evaluated Value = 18
5. Piece 7 jump to F6 => new Game Board 14 => Evaluated Value = 18
6. Piece 11 move to G3 => new Game Board 15 => Evaluated Value = 19
7. Piece 11 move to G5 => new Game Board 16 => Evaluated Value = 19
8. Piece 12 move to G5 => new Game Board 17 => Evaluated Value = 19
9. Piece 12 move to G7 => new Game Board 18 => Evaluated Value = 19

White counter move (MIN Player) to black player move (MAX player) 3 (Piece 2 double jumps to A1)

1. Piece 6 move to E1 => new Game Board 19 => Evaluated Value = 20
2. Piece 7 jump to F2 => new Game Board 20 => Evaluated Value = 19
3. Piece 7 jump to F6 => new Game Board 21 => Evaluated Value = 19
4. Piece 11 move to G3 => new Game Board 22 => Evaluated Value = 20
5. Piece 11 move to G5 => new Game Board 23 => Evaluated Value = 20
6. Piece 12 move to G5 => new Game Board 24 => Evaluated Value = 20
7. Piece 12 move to G7 => new Game Board 25 => Evaluated Value = 20

White counter move (MIN Player) to black player move (MAX player) 4 (Piece 2 double jumps to E1)

1. Piece 3 move to C1 => new Game Board 26 => Evaluated Value = 20
2. Piece 3 move to C3 => new Game Board 27 => Evaluated Value = 19
3. Piece 7 jump to F2 => new Game Board 28 => Evaluated Value = 19
4. Piece 7 jump to F6 => new Game Board 29 => Evaluated Value = 19
5. Piece 11 move to G3 => new Game Board 30 => Evaluated Value = 20
6. Piece 11 move to G5 => new Game Board 31 => Evaluated Value = 20
7. Piece 12 move to G5 => new Game Board 32 => Evaluated Value = 20
8. Piece 12 move to G7 => new Game Board 33 => Evaluated Value = 20

White counter move (MIN Player) to black player move (MAX player) 5 (Piece 2 moves to B6)

1. Piece 3 move to C1 => new Game Board 34 => Evaluated Value = 18
2. Piece 3 move to C3 => new Game Board 35 => Evaluated Value = 18
3. Piece 4 move to C3 => new Game Board 36 => Evaluated Value = 18
4. Piece 4 move to C5 => new Game Board 37 => Evaluated Value = 18
5. Piece 6 move to E1 => new Game Board 38 => Evaluated Value = 18
6. Piece 7 jump to F2 => new Game Board 39 => Evaluated Value = 17
7. Piece 7 jump to F6 => new Game Board 40 => Evaluated Value = 17
8. Piece 11 move to G3 => new Game Board 41 => Evaluated Value = 18
9. Piece 11 move to G5 => new Game Board 42 => Evaluated Value = 18
10. Piece 12 move to G5 => new Game Board 43 => Evaluated Value = 18
11. Piece 12 move to G7 => new Game Board 44 => Evaluated Value = 18

White counter move (MIN Player) to black player move (MAX player) 6 (Piece 5 moves to B6)

1. Piece 3 move to C1 => new Game Board 45 => Evaluated Value = 18
2. Piece 3 move to C3 => new Game Board 46 => Evaluated Value = 18
3. Piece 4 move to C3 => new Game Board 47 => Evaluated Value = 18
4. Piece 4 move to C5 => new Game Board 48 => Evaluated Value = 18
5. Piece 6 move to E1 => new Game Board 49 => Evaluated Value = 18
6. Piece 7 jump to F2 => new Game Board 50 => Evaluated Value = 17
7. Piece 7 jump to F6 => new Game Board 51 => Evaluated Value = 17
8. Piece 11 move to G3 => new Game Board 52 => Evaluated Value = 18
9. Piece 11 move to G5 => new Game Board 53 => Evaluated Value = 18
10. Piece 12 move to G5 => new Game Board 54 => Evaluated Value = 18
11. Piece 12 move to G7 => new Game Board 55 => Evaluated Value = 18

White counter move (MIN Player) to black player move (MAX player) 7 (Piece 5 moves to B8)

1. Piece 3 move to C1 => new Game Board 56 => Evaluated Value = 18
2. Piece 3 move to C3 => new Game Board 57 => Evaluated Value = 18
3. Piece 4 move to C3 => new Game Board 58 => Evaluated Value = 18
4. Piece 4 move to C5 => new Game Board 59 => Evaluated Value = 18
5. Piece 6 move to E1 => new Game Board 60 => Evaluated Value = 18
6. Piece 7 jump to F2 => new Game Board 61 => Evaluated Value = 17
7. Piece 7 jump to F6 => new Game Board 62 => Evaluated Value = 17
8. Piece 11 move to G3 => new Game Board 63 => Evaluated Value = 18
9. Piece 11 move to G5 => new Game Board 64 => Evaluated Value = 18
10. Piece 12 move to G5 => new Game Board 65 => Evaluated Value = 18
11. Piece 12 move to G7 => new Game Board 66 => Evaluated Value = 18

White counter move (MIN Player) to black player move (MAX player) 8 (Piece 5 moves to D6)

1. Piece 3 move to C1 => new Game Board 67 => Evaluated Value = 18
2. Piece 3 move to C3 => new Game Board 68 => Evaluated Value = 18
3. Piece 4 move to C3 => new Game Board 69 => Evaluated Value = 18
4. Piece 4 move to C5 => new Game Board 70 => Evaluated Value = 18
5. Piece 6 move to E1 => new Game Board 71 => Evaluated Value = 18
6. Piece 7 jump to F2 => new Game Board 72 => Evaluated Value = 17
7. Piece 7 jump to F6 => new Game Board 73 => Evaluated Value = 17
8. Piece 11 move to G3 => new Game Board 74 => Evaluated Value = 18
9. Piece 11 move to G5 => new Game Board 75 => Evaluated Value = 18
10. Piece 12 move to G5 => new Game Board 76 => Evaluated Value = 18
11. Piece 12 move to G7 => new Game Board 77 => Evaluated Value = 18

White counter move (MIN Player) to black player move (MAX player) 9 (Piece 5 moves to D8)

1. Piece 3 move to C1 => new Game Board 78 => Evaluated Value = 18
2. Piece 3 move to C3 => new Game Board 79 => Evaluated Value = 18
3. Piece 4 move to C3 => new Game Board 80 => Evaluated Value = 18
4. Piece 4 move to C5 => new Game Board 81 => Evaluated Value = 18
5. Piece 6 move to E1 => new Game Board 82 => Evaluated Value = 18
6. Piece 7 jump to F2 => new Game Board 83 => Evaluated Value = 17
7. Piece 7 jump to F6 => new Game Board 84 => Evaluated Value = 17
8. Piece 11 move to G3 => new Game Board 85 => Evaluated Value = 18
9. Piece 11 move to G5 => new Game Board 86 => Evaluated Value = 18
10. Piece 12 move to G5 => new Game Board 87 => Evaluated Value = 18
11. Piece 12 move to G7 => new Game Board 88 => Evaluated Value = 18

White counter move (MIN Player) to black player move (MAX player) 10 (Piece 8 jumps to C1)

1. Piece 3 move to C1 => new Game Board 89 => Evaluated Value = 19
2. Piece 3 move to C3 => new Game Board 90 => Evaluated Value = 19
3. Piece 4 move to C3 => new Game Board 91 => Evaluated Value = 19
4. Piece 4 move to C5 => new Game Board 92 => Evaluated Value = 19
5. Piece 7 move to E3 => new Game Board 93 => Evaluated Value = 19
6. Piece 7 jump to F6 => new Game Board 94 => Evaluated Value = 18
7. Piece 11 move to G3 => new Game Board 95 => Evaluated Value = 19
8. Piece 11 move to G5 => new Game Board 96 => Evaluated Value = 19
9. Piece 12 move to G5 => new Game Board 97 => Evaluated Value = 19
10. Piece 12 move to G7 => new Game Board 98 => Evaluated Value = 19

White counter move (MIN Player) to black player move (MAX player) 11(Piece 8 jumps to C5)

1. Piece 3 move to C1 => new Game Board 99 => Evaluated Value = 19
2. Piece 3 move to C3 => new Game Board 100 => Evaluated Value = 19
3. Piece 4 move to C3 => new Game Board 101 => Evaluated Value = 19
4. Piece 4 jump to D6 => new Game Board 102 => Evaluated Value = 18
5. Piece 6 move to E1 => new Game Board 103 => Evaluated Value = 19
6. Piece 6 move to E3 => new Game Board 104 => Evaluated Value = 19
7. Piece 11 move to G3 => new Game Board 105 => Evaluated Value = 19
8. Piece 11 move to G5 => new Game Board 106 => Evaluated Value = 19
9. Piece 12 move to G5 => new Game Board 107 => Evaluated Value = 19
10. Piece 12 move to G7 => new Game Board 108 => Evaluated Value = 19

White counter move (MIN Player) to black player move (MAX player) 12 (Piece 9 double jumps to A1)

1. Piece 4 move to C3 => new Game Board 109 => Evaluated Value = 22
2. Piece 4 move to C5 => new Game Board 110 => Evaluated Value = 22
3. Piece 6 move to E1 => new Game Board 111 => Evaluated Value = 22
4. Piece 11 move to G3 => new Game Board 112 => Evaluated Value = 22
5. Piece 11 move to G5 => new Game Board 113 => Evaluated Value = 22
6. Piece 12 move to G5 => new Game Board 114 => Evaluated Value = 22
7. Piece 12 move to G7 => new Game Board 115 => Evaluated Value = 22

White counter move (MIN Player) to black player move (MAX player) 13 (Piece 9 moves to D6)

1. Piece 3 move to C1 => new Game Board 116 => Evaluated Value = 18
2. Piece 3 move to C3 => new Game Board 117 => Evaluated Value = 18
3. Piece 4 move to C3 => new Game Board 118 => Evaluated Value = 18
4. Piece 4 move to C5 => new Game Board 119 => Evaluated Value = 18
5. Piece 6 move to E1 => new Game Board 120 => Evaluated Value = 18
6. Piece 7 jump to F2 => new Game Board 121 => Evaluated Value = 17
7. Piece 7 move to E5 => new Game Board 122 => Evaluated Value = 18
8. Piece 11 move to G3 => new Game Board 123 => Evaluated Value = 18
9. Piece 11 move to G5 => new Game Board 124 => Evaluated Value = 18
10. Piece 12 move to G5 => new Game Board 125 => Evaluated Value = 18
11. Piece 12 move to G7 => new Game Board 126 => Evaluated Value = 18

AI Counter Moves (MAX Player)

3rd step no move made boards are evaluated.

Max

|

Min

|

Max

At the lowest level of the tree where the boards are evaluated then passed up to the previous level which is a Min level so the lowest value is taken as best value for each branch.

This gives use the following values for the 13 branches at this level.

1 2 3 4 5 6 7 8 9 10 11 12 13

{18, 19, 19, 19, 17, 17, 17, 17, 17, 18, 18, 22, 17}

These values are then passed up to the root of the tree which is a MAX level so the greatest value is taken. In This case 22 which corresponds to move 12 (Piece 9 double jumps to A1)

Given us the following board



ToString () representation = "\nB\_B\_B\_\_\_\n\_\_\_w\_\_\_\_\n\_\_\_\_\_\_B\_\n\_w\_\_\_\_\_\_\n\_\_b\_\_\_b\_\n\_\_\_\_\_\_\_\_\n\_\_\_\_\_\_\_\_\n\_\_\_W\_W\_\_"

# Manual Test Cases

Manual test will be conducted by both visually monitoring and using a debugging tool to ascertain that the application is preforming as expected.

**Test Case: Start a single player game.**

**Step 1:** Launch Application

**Expected Result** Title Screen is displayed with labels Draught Game and 1 Player and 2 Player and forced moves check box

**Step 2:** Tap 1 Player text

**Expected Result** Choose difficulty screen is displayed with options Easy Normal Hard

**Step 3:** Choose any difficult

**Expected Result** The game board with all game pieces in their starting positions and the text “Your Turn” and “White player” should be visible

**Test Case: Start a two player game.**

**Step 1:** Launch Application

**Expected Result** Title Screen is displayed with labels Draught Game and 1 Player and 2 Player and forced moves check box

**Step 2:** Tap 2 Player text

**Expected Result** The game board with all game pieces in their starting positions and the text “Your Turn Player 1” and “White Player” should be visible

**Test Case: Make a normal move in single player game.**

**Prerequisite** 1 Player game started

**Step 1:** Tap a white draught piece

**Expected Result** If this piece can move the movable to squares should be highlighted if not no screen change

**Step 2:** Tap one of the highlighted squares

**Expected Result** The chosen piece should have been moved to the selected square, the text “phone is thinking” should be visible until AI has completed a turn

**Test Case: Make a normal move in two player game for player 1.**

**Prerequisite** 2 Player game started

**Step 1:** Tap a white draught piece

**Expected Result** If this piece can move the movable to squares should be highlighted if not no screen change

**Step 2:** Tap one of the highlighted squares

**Expected Result** The chosen piece should have been moved to the selected square, the text “Your Turn Player 2” and “Black Player” should be visible

**Test Case: Make a normal move in two player game for player 2.**

**Prerequisite** 2 Player game started, player 1 has just completed a move.

**Step 1:** Tap a black draught piece

**Expected Result** If this piece can move the movable to squares should be highlighted if not no screen change

**Step 2:** Tap one of the highlighted squares

**Expected Result** The chosen piece should have been moved to the selected square, the text “Your Turn Player 1” and “White Player” should be visible

**Test Case: Make a jump move in either game type.**

**Prerequisite** 1 or 2 Player game started. Must be a human players turn, an opponent piece must be in a jumpable position

**Step 1:** Tap a draught piece that is in a position to jump an opponent piece

**Expected Result** If this piece can jump the movable to squares should be highlighted with an opponent’s piece between the selected piece and the highlighted square, if not no screen change

**Step 2:** Tap the highlighted square that represents a jump move.

**Expected Result** The chosen piece should have been moved to the selected square. The jumped opponent piece should be removed, the text “Your Turn Player 2” and “Black Player” should be visible or “phone is thinking” in a single player game.

**Test Case: Make a multiple jump move in either game type.**

**Prerequisite** 1 or 2 Player game started. Must be a human players turn, opponent must have pieces in a position that allows multiple jumps.

**Step 1:** Tap a draught piece that is in a position to jump more than one opponent piece in a row

**Expected Result** If this piece can jump the movable to squares should be highlighted with an opponent’s piece between the selected piece and the highlighted square, if not no screen change

**Step 2:** Tap the highlighted square that represents a jump move.

**Expected Result** The chosen piece should have been moved to the selected square, the jumped opponent piece should be removed, the square that represents the next jump move should now be highlighted.

**Step 3:** Tap the next highlighted square.

**Expected Result** The chosen piece should have been moved to the selected square, the jumped opponent piece should be removed. This should continue till no more jumps are available to selected piece this turn. At which time the text “Your Turn Player 2” and “Black Player” should be visible or “phone is thinking” in a single player game.

**Test Case: Win a game as player one.**

**Prerequisite** 1 or 2 Player game started. Must be player 1’s turn, player must be in a position to jump opponents last piece with next move.

**Step 1:** Tap a draught piece that is in a position to jump the opponent’s last piece

**Expected Result** If this piece can jump the movable to squares should be highlighted with an opponent’s piece between the selected piece and the highlighted square, if not no screen change

**Step 2:** Tap the highlighted square that represents a jump move.

**Expected Result** An image of a crown and the text “White Wins” should be displayed

**Test Case: Win a game as player two.**

**Prerequisite** 2 Player game started. Must be player 2’s turn, player must be in a position to jump opponents last piece with next move.

**Step 1:** Tap a draught piece that is in a position to jump the opponent’s last piece

**Expected Result** If this piece can jump the movable to squares should be highlighted with an opponent’s piece between the selected piece and the highlighted square, if not no screen change

**Step 2:** Tap the highlighted square that represents a jump move.

**Expected Result** An image of a crown and the text “Black Wins” should be displayed

**Test Case: Select easy difficulty**

**Prerequisite** 1 Player game must be selected from title screen. A break point must be set at the beginning of the MinMax method in the AI class. A watch must be added to the variable numberOfMovesAhead.

**Step 1:** Complete a move

**Expected Result** The code execution should pause at the breakpoint. The watch should show the variable value to be 2

**Test Case: Select normal difficulty**

**Prerequisite** 1 Player game must be selected from title screen. A break point must be set at the beginning of the MinMax method in the AI class. A watch must be added to the variable numberOfMovesAhead.

**Step 1:** Complete a move

**Expected Result** The code execution should pause at the breakpoint. The watch should show the variable value to be 4

**Test Case: Select hard difficulty**

**Prerequisite** 1 Player game must be selected from title screen. A break point must be set at the beginning of the MinMax method in the AI class. A watch must be added to the variable numberOfMovesAhead.

**Step 1:** Complete a move

**Expected Result** The code execution should pause at the breakpoint. The watch should show the variable value to be 8

# UX Test Documentation

## Informed Consent.

Kevin Smith a student of the Institute of Technology Tralee as part of he’s 3rd Year Computing project requests your participation in a usability study of WPDraughts a draughts game deployed on the Windows 7 Phone platform. You will be required to complete a background questionnaire at the beginning of the session then you will be allowed to use the application while being asked to perform a simple few tasks. After the session you will be asked to fill in a post session questionnaire. Any personal information given such as name or contact details will be keep confidential by Kevin Smith. Questionnaire answers, comments made by you in relation to the application and testers observations of your session may be shared with members of the faculty of the Institute of Technology Tralee. Kevin Smith does not accept responsibility for further distribution of said information by the Institute of Technology nor does he accept responsibility for the institute’s use of the information. No recordings either visual or audio will be created of your session. You participation is completely voluntary and you should feel under no obligation to do so. If you agree to these terms and are willing to participate in this study please sign below.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature Date

## Back Ground Questionnaire

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Email: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Contact Number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Profession: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Age: \_\_\_\_\_\_\_

Do you own a smart phone? Yes/No

If so is it a Windows Phone? Yes/No

If not what Type? IPhone Android

With the exception of the texting and phone applications of a phone what else do you use your phone for? Tick any that apply.

Social Networking Photography Social Networking

Gaming Business Internet

News Weather E-mail

TV/Movies Music Shopping

GPS

If you use your phone for gaming what types of games do you play? Tick all that apply.

Action/Adventure Card/Board Educational

Family Music Platformer

Puzzle/Trivia Racing/Flying Role Playing

Shooter Sports/Recreation Strategy/Simulation

Have you played Draughts before? Yes/No

How familiar are you with the rules of Draughts? Not at All / A Little / Reasonably well / Very well

Have you used Windows 8? Yes/No

## Post Session Questionnaire

Did you find the game menus easy to navigate? Yes/No

Additional Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Did you understand all of the menu options? Yes/No

Additional Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Are there any other options that you believe should be present? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Did you find in game labels or text easy to read and understand? Yes/No

Additional Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Did you like the overall appearance of the game? Yes/No

Additional Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Did you find the draught pieces easy to see on the game board? Yes/No

Additional Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Did you find it easy to recognise the highlighted move squares? Yes/No

Additional Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Did you find that the game offered little challenge at easy level? Yes/No

Additional Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Did you find the game offered a greater challenge at hard level? Yes/No

Additional Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Did you find it easy to select your intended points on the board? Yes/No

Additional Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please another any further comments on any aspect of you session experience here \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Sessions Tasks

**To be filled used and filled out by tester during session.**

**Once the participant has signed the consent form and filled in the background questionnaire and you have retrieved these documents the session may begin.**

**Hand participant the phone once it is turned on and unlocked and the main tile menu screen is visible**

Please launch the application from the main phone tile screen.

Completed: Yes/No Time Taken: Prompting required: Yes/No

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please select a 1 player game.

Completed: Yes/No Time Taken: Prompting required: Yes/No

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please select the easy difficulty option.

Completed: Yes/No Time Taken: Prompting required: Yes/No

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please in your own time play the game till completion.

Completed: Yes/No Time Taken: Prompting required: Yes/No

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Once participant has completed a play thru**

Please tap on the game over screen.

Completed: Yes/No Time Taken: Prompting required: Yes/No

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Once again please select a 1 player game.

Completed: Yes/No Time Taken: Prompting required: Yes/No

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please select the hard difficulty option.

Completed: Yes/No Time Taken: Prompting required: Yes/No

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Please in your own time play the game till completion.

Completed: Yes/No Time Taken: Prompting required: Yes/No

Comments: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Thank the participant for their efforts. Retrieve the phone and then present them with the post session questionnaire.**