Logical Operators:

Chinese Checkers

Iteration Two

February 5th, 2014

Curtis Smith

Peter Pobojewski

Saajid Mohammed

Chris Kellendonk

James Kostiuk

Kuba Subczynski

Ben Stitt

Taylor Whited

# Table of contents

[1 Table of contents 1](#_Toc381234682)

[2 Planning 2](#_Toc381234683)

[2.1 Team capacity calculations 2](#_Toc381234684)

[2.11 Release level plan 2](#_Toc381234685)

[2.12 Iteration level plan 2](#_Toc381234686)

[2.2 Meeting minutes 2](#_Toc381234687)

[2.21 February 6, 2014 2](#_Toc381234688)

[2.22 February 13, 2014 3](#_Toc381234689)

[2.23 February 20, 2014 5](#_Toc381234690)

[2.24 February 25, 2014 6](#_Toc381234691)

[2.3 Use cases 6](#_Toc381234692)

[2.31 Play offline game (IT-2-UC-3) 7](#_Toc381234693)

[2.32 Play against an AI (IT-2-UC-4) 9](#_Toc381234694)

[2.33 Show possible moves (IT-2-UC-5) 11](#_Toc381234695)

[2.34 Save a game (IT-2-UC-6) 13](#_Toc381234696)

[2.35 Add an AI player (IT-2-UC-7) 15](#_Toc381234697)

[2.36 Finish a game (IT-2-UC-8) 17](#_Toc381234698)

[2.37 Move a peg (IT-2-UC-9) 19](#_Toc381234699)

[2.38 View help pages (IT-2-UC-10) 21](#_Toc381234700)

[2.39 Read the EULA (IT-2-UC-11) 23](#_Toc381234701)

[2.4 Burn down 25](#_Toc381234702)

[2.41 Iteration two progress 25](#_Toc381234703)

[2.42 Iteration two versus iteration one 25](#_Toc381234704)

[2.5 Presentation 25](#_Toc381234705)

[2.6 Policies 26](#_Toc381234706)

[2.61 End-user license agreement (EULA) 26](#_Toc381234707)

[3 Design 30](#_Toc381234708)

[3.1 Class diagrams 30](#_Toc381234709)

[3.11 Game board engine 30](#_Toc381234710)

[3.2 User interface control elements 30](#_Toc381234711)

[3.21 Main activity 30](#_Toc381234712)

[3.22 Offline Configuration activity 30](#_Toc381234713)

[3.23 Offline Game activity 31](#_Toc381234714)

[3.24 Offline Game activity: resume dialog 31](#_Toc381234715)

[3.25 Offline Game activity: end of game dialog 32](#_Toc381234716)

[3.3 User interface control flow 32](#_Toc381234717)

[3.31 Overview 32](#_Toc381234718)

[3.32 Main Activity 32](#_Toc381234719)

[3.33 Offline game 32](#_Toc381234720)

[3.34 Offline Game Activity - Resume Dialog 32](#_Toc381234721)

[3.35 Offline Game Activity - End of Game Dialog 32](#_Toc381234722)

[3.36 UI Flow - Offline Configuration Activity 32](#_Toc381234723)

[3.4 Visual design 32](#_Toc381234724)

[3.41 Introduction 32](#_Toc381234725)

[3.42 Colour swatches 32](#_Toc381234726)

[3.43 Mock-ups 33](#_Toc381234727)

[3.5 AI design 35](#_Toc381234728)

[4 Testing 36](#_Toc381234729)

[4.1 Iteration two testing plan 36](#_Toc381234730)

[4.2 Integration testing 36](#_Toc381234731)

[4.3 Unit testing 36](#_Toc381234732)

[4.4 Acceptance testing 36](#_Toc381234733)

[4.5 Ease of use testing 36](#_Toc381234734)

[5 Appendices 37](#_Toc381234735)

[5.1 Appendix A 38](#_Toc381234736)

[5.2 Appendix B 39](#_Toc381234737)

[5.3 Appendix C 40](#_Toc381234738)

[5.4 Appendix D 41](#_Toc381234739)

[5.5 Appendix E 42](#_Toc381234740)

[5.6 Appendix F 43](#_Toc381234741)

[5.7 Appendix G 44](#_Toc381234742)

[6 References 45](#_Toc381234743)

# Planning

## Team capacity calculations

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Release level plan | |  | Iteration level plan | | |
| **Start** | 9-Jan |  |  | **# Weeks** | **Hours** |
| **Finish** | 27-Mar |  | **1st iteration** | 4 | 192 |
| **Last day of class** | 4-Apr |  | **2nd iteration** | 3 | 144 |
|  |  |  | **3rd iteration** | 3 | 144 |
| **Work days** | 77 |  | **4th iteration** | 1 | 48 |
| **Work weeks** | 11 |  |  |  |  |
| **# of people** | 8 |  | ***Total hours*** | | 528 |
| **Hours/Week/Person** | 6 |  |  |  |  |
| ***Total hours*** | 528 |  |  |  |  |

## Meeting minutes

### February 6, 2014

After some preparation, the group presented the results of iteration one to Vlad. This included introducing the team members, as well as answering questions about tools such as Android Studio, GitHub, and Travis.

After the presentation, the group conducted a regular meeting.

Firstly, the idea of a virtual suggestion box was put forward, and agreed upon.

Then, Taylor made suggestions regarding the format of documentation. Firstly, she volunteered to take over collecting documentation into binders for each iteration. She also suggested that the directories on GitHub should be organized by creator name and by iteration number. It was decided that the group would continue to use the master branch for documentation.

There was also a discussion on format. Documents should use references in APA format, and appendices should have clear and unique titles. Also, documents should be written in present tense, using no personal pronouns.

Next, issues with the application were discussed. Touch areas were not being scaled correctly. Some visual adjustments needed to be made as well, including vertically centering buttons, as well as adjusting the colours of text boxes and adjustments to spacing.

The group agreed to link to commits of related materials in GitHub tasks.

Peter expressed some technical issues involving Android emulation, and having trouble implementing a Java interface.

There were also issues with Travis. It wasn’t finding everything, and it was pulling from every branch.

Ben shared his thanks with the group, and resolved to use more tasks to communicate on GitHub.

The group agreed to try to complete tasks at a more even pace, and if a task is too large to complete at once, to break it into smaller tasks.

It was also agreed that the group needed to collaborate more on establishing the architecture of the application. This could be done partly by assigning brainstorming tasks, and they could be brought up at the next meeting to share with the group.

At this point, the group moved to another room to continue the discussion.

The group then discussed what goals or use cases to include in the second iteration. It was decided that an AI and save and continue features should be included. However, a display of pegs remaining to win was deemed an unnecessary frill, and left out.

It was decided that the next meeting be held the following Thursday.

The group was then assigned brainstorming tasks, and spoke briefly on what should occur when a game ends. The group then adjourned.

### February 13, 2014

The group discussed the perfect mark received for the first iteration, and wondered which percentage of the total was represented by this mark.

Then the topic turned to the use cases which had been posted to GitHub, with the intention of dividing these into individual tasks. If time allowed, the group would also speak about issues with Travis.

The first topic discussed was how to save a game. It was decided that it should be automatically saved when the game was closed as a serialized object.

Next, loading a game was discussed. The file should be loaded automatically, and prompt the user to continue the next time they select offline play.

It was decided that possible moves should be shown for a peg when it is tapped. There will be no option to toggle this feature in this iteration.

It was announced that unit tests for the game board are working well.

The discussion returned to saving and loading games. The assigned tasks included making UI mock-ups for the load dialog, updating existing flow diagrams, coding for UI flow control and for loading the board, and creating unit tests. The entire game state must be serialized, rather than just the game board.

Next, the discussion returned to how to show possible moves. The tasks assigned to this use case include creating a UI mock-up, creating a UML or flow diagram, and coding. This feature is difficult to test, and must be done manually.

It was agreed that Hot Seat mode should be renamed to Offline mode.

The topic of in-game help arose next. Tutorial-style overlays were considered, but will not be included in this iteration. Instead, the help section can reside in the settings screen native to Android. This help section should include an FAQ, and rules. Tasks assigned for this use case include creating a UI mock-up, new activity classes in UML, and coding, including a parent class for pages. This can be tested automatically.

Next, it was decided that a game should be saved using the onPause event, and loaded using the onCreate event. The loading dialog box will only be displayed when entering an offline game from the main menu. Loading will be performed while the dialog box is being displayed, so there is no delay after selecting to continue.

Next, the option to play against an AI was discussed. The tasks assigned include the UML and activity diagrams, and an interface should be coded before implementation. It was also decided that the AI should hold its own temporary copy of the game state.

Next, the group discussed how to toggle AI players in the configuration activity. It was decided that a button should appear next to each player’s name entry box. Once activated, the applicable text box changes to three radio buttons, indicating the difficulty level. Two UI mock-ups are to be created for this: one for a single button toggle, and one for a switch-style toggle. These changes require modification of the existing offline play flow control diagram as well.

To test AI players, one may set up games in which AIs play against each other. Foremost, an AI player must not break the rules. After this, their ability to win against each other should be proportional to their respective difficulty levels.

The next group meeting was scheduled for the following Thursday, during reading week. Absent group members may attend remotely via Google Hangouts or a similar system.

The group then discussed what would happen when a game is finished. It was decided that the game screen should be blurred, as if behind frosted glass, and a dialog announcing the winner should be displayed, along with one or two buttons. For offline play, it should display a button to return to the main menu. For online play, it should display two buttons: one to return to the main menu, one to spectate for the remainder of the game.

Lastly, Taylor volunteered to reorganize some folders on GitHub.

### February 20, 2014

Note: Due to online meeting, these minutes may be incomplete.

The group spoke over the Internet via Google hangouts.

The meeting began with a review of the iteration 2 burn-down. It seems that the group is opening too many tasks without closing them.

Peter expressed concern over how to implement unit tests. Thus far, he had been testing in JCreator, but will test further in Android Studio.

Next, the group covered each use case in turn.

Taylor showed some test mock-ups for the endgame dialog. The group decided that the game screen should be dimmed, but not blurred.

Kuba completed load dialogs, and Saajid opened a separate issue for testing this.

Next, the group reviewed a mock-up for showing possible moves, and tried to decide on a colour scheme for unoccupied home spaces and for selected pegs. The group agreed that unoccupied home spaces should be lighter, and selected pegs should be darker.

The group was presented with a mock-up for a help screen, and agreed that it was good.

In order to implement an AI player, James needed an interface or guidelines to complete the code.

The group was shown mock-ups for AI selection on the configuration screen, and decided it should start by displaying all robots except for the first player.

For the proper endgame mock-up, Taylor needs a screenshot from Kuba. It was decided that the win condition can be determined from the game board alone.

New tasks were discussed, including the phase 2 testing plan, UI tables for activities, game board animation, acceptance testing, activity and integration tests, the network API, touch detection, and the EULA.

Saajid needs some time for testing, so the group agreed to try to get some modules completed ahead of time.

The group agreed that most or all use cases could be completed by the deadline, but likely cannot all be tested.

The group agreed that play should proceed in a clockwise direction. Variables need to be assigned to determine player home positions on the game board, and a variable needs to be assigned to show that a peg has jumped another.

It was also agreed that only one level of AI needs to be completed for this iteration.

See for breakdown of progress of iteration.

### February 25, 2014

Note: These minutes may be incomplete due to the note taker arriving late.

The group discussed what to include in the iteration 2 binder. The iteration 1 presentation will be included.

The group will try to have coding done by Wednesday night.

For acceptance testing, the user must be able to start a game for up to six players. Scripts should be made regarding tasks and expected consequences. The write-up and testing should be done by early Thursday afternoon. A reference should be included for the template used.

The group discussed whether it was worth it to continue using Travis. It was decided that since it does no harm, Travis will be kept in place for the time being.

The group members are to sort their binder directories into subdirectories to designate groupings of files.

The EULA is to be kept as a class rather than an activity, and should use a custom dialog similar to the OfflineGameResumeDialog.

For AI documentation, a flow diagram should be provided. Currently, the AI is selectable, but will not run as intended.

Use cases can be placed under the planning section in the binder.

The burn down looked bad, but was improving. This was further improved by closing completed issue.

## Use cases

Use case templates ("Use Case Templates", 2014) provided by TechnoSolutions.

### Play offline game (IT-2-UC-3)

#### Revision history

|  |  |  |
| --- | --- | --- |
| Date | Author | Description of change |
| 2014/02/11 | Ben Stitt | Authored use case. |
| 2014/02/16 | Ben Stitt | Revised to meet documentation specifications and name change. |

#### Use case

Play offline game

#### Id

IT-2-UC-3

#### Description

The user would like to play a complete local game with their friends. This must be made possible without communicating with an external server, or employing an AI.

#### Level

User goal

#### Primary actor

The user

#### Supporting actors

Friends of the user who are present

#### Stakeholders and interests

The user and their friends

#### Pre-conditions

The user must have the application installed and running, and have completed the Offline Configuration Activity.

#### Post conditions

##### Success end condition

The appropriate game end screen is displayed. It informs the players which of them has won.

##### Failure end condition

The user is unable to launch the Offline Game Activity.

##### Minimal guarantee

The application does not crash, and data on the device remains unaffected.

#### Trigger

The user completes the Hot Seat Configuration Activity.

#### Main success scenario

1. The user clicks the Start Game button on the Offline Configuration screen.
2. Each player makes their moves in turn.
3. The Offline Game Activity ends with the appropriate victory screen.

#### Extensions

Alternatively, the user could include one or more AI players.

#### Frequency

This use case should occur each time the user starts an offline game. As such, it could occur weekly to hourly.

#### Assumptions

The user has at least one friend present.

#### Special requirements

The device must run Android and have the necessary space to install the game.

#### Issues

1. How does the local server coordinate player turns?

#### To do

1. Return to main menu.

### Play against an AI (IT-2-UC-4)

#### Revision history

|  |  |  |
| --- | --- | --- |
| Date | Author | Description of change |
| 2014/02/12 | Ben Stitt | Authored use case. |

#### Use case

Play against an AI

#### Id

IT-2-UC-4

#### Description

The user has completed the applicable Configuration Activity, and has opted to include one or more AI players. They then wish to play through a game with these AI players.

#### Level

User Goal

#### Primary actor

The user

#### Stakeholders and interests

The user wishes to play a game of Chinese Checkers, but does not have the appropriate number of players present.

#### Pre-conditions

The user must have the application installed and running, and have completed the appropriate Configuration Activity.

#### Post conditions

##### Success end condition

The appropriate win or loss screen is displayed.

##### Failure end condition

The AI player fails to launch, or fails to take its turns.

##### Minimal guarantee

The application does not crash, and data on the device remains unaffected.

#### Trigger

The user completes the appropriate Configuration Activity, and has included one or more AI players in the game.

#### Main success scenario

1. The user clicks the Start Game button on the appropriate Configuration screen.
2. The user and the AI players make their moves in turn.
3. The Game Activity ends with the appropriate win or loss screen.

#### Extensions

An AI player could be included in either a local game or a network game.

#### Variations

An AI can be run either locally, or from a remote server.

##### Frequency

This use case should occur each time the user starts game that includes an AI player. As such, it could occur weekly to hourly.

##### Assumptions

The user wishes to play with a number of players different than the number of players otherwise available.

#### Special requirements

The device must run Android and have the necessary space to install the game. If the AI player is run from a remote server, the user must be able to connect to the Internet.

#### Issues

1. Is the AI process prepared to run?
2. Is the local and/or network server process prepared to interact with the AI process?

#### To do

1. Return to main menu.

### Show possible moves (IT-2-UC-5)

#### Revision history

|  |  |  |
| --- | --- | --- |
| Date | Author | Description of change |
| 2014/02/12 | Ben Stitt | Authored use case. |
| 2014/02/13 | Peter Pobojewski | Edited use case to more accurately portray the process. |
| 2014/02/16 | Ben Stitt | Revised to meet documentation syntax. |

#### Use case

Show possible moves

#### Id

IT-2-UC-5

#### Description

The user has a game in progress, and it is their turn. They wish to see which moves they can make using a certain peg.

#### Level

User goal

#### Primary actor

The user

#### Stakeholders and interests

The may be unfamiliar with the rules of Chinese Checkers, or may not be able to see which moves can be made.

#### Pre-conditions

The user must have the application installed and running. The user must have a game active, and it must be their turn. They must also be currently selecting a piece they control.

#### Post conditions

##### Success end condition

The user sees their possible moves, and is able to make any one of them.

##### Failure end condition

The user is unable to see their possible moves, or game play is interrupted.

##### Minimal guarantee

The application does not crash, and data on the device remains unaffected.

#### Trigger

User touches a piece they wish to move.

#### Main success scenario

1. The user interacts with the board by touching a piece.
2. The game responds by somehow animating or illuminating places this piece can move to.
3. The player continues their turn as usual.

#### Frequency

This function is activated whenever a user interacts with a piece they control.

#### Assumptions

If there is a toggle for this feature, the user has enabled the feature.

#### Special requirements

The device must run Android and have the necessary space to install the game. The user must also have a game active, and it must be their turn.

#### Issues

1. Can this feature be disabled manually by the player?
2. How do we distinguish possible moves for a piece?

#### To do

1. The user may take their turn.

### Save a game (IT-2-UC-6)

#### Revision history

|  |  |  |
| --- | --- | --- |
| Date | Author | Description of change |
| 2014/02/13 | Ben Stitt | Authored use case |
| 2014/02/13 | Peter Pobojewski | Edited use case to be more accurate. |
| 2014/02/16 | Ben Stitt | Revised to meet documentation syntax and reflect decisions made in most recent meeting. |

#### Use case

Save a game

#### Id

IT-2-UC-6

#### Description

The user wishes to continue an active local game at a later time.

#### Level

User goal

#### Primary actor

The user

#### Stakeholders and interests

The user has more pressing issues at hand and may want to continue their game later. It is important that the user is given the ability to continue a game at their next earliest convenience.

#### Pre-conditions

The user must have the application installed and running, and be in the midst of an active offline game.

#### Post conditions

##### Success end condition

The current game is saved in some file on their device.

##### Failure end condition

The current game is not saved, the user is notified.

##### Minimal guarantee

The application does not crash, and data on the device remains unaffected.

#### Trigger

User exits game and is prompted with a “Would you like to save this game?”

#### Main success scenario

1. The user triggers the save in a certain way
2. The current game is saved to a local file
3. The application returns to the main menu screen.

#### Variations

Network games will automatically be saved to a remote server.

##### Frequency

This use case will be triggered each time a user leaves an offline game without completing it. As such, it could be triggered daily to monthly.

##### Assumptions

The user wishes to complete the game at a later time, and their device has enough free space to save a game.

#### Special requirements

The device must run Android and have the necessary space to install the application, and to save the current game. If the game is being played over the network, the device must be connected to the Internet.

#### Issues

1. Where is the game being saved?

#### To do

1. Be prepared to load the game.

### Add an AI player (IT-2-UC-7)

#### Revision history

|  |  |  |
| --- | --- | --- |
| Date | Author | Description of change |
| 2014/02/15 | Ben Stitt | Authored use case. |
| 2014/02/16 | Ben Stitt | Revised to meet documentation syntax. |

#### Use case

Add an AI player

#### Id

IT-2-UC-7

#### Description

The user has entered the Offline Configuration Activity, and has selected the desired number of players. They wish to exchange slots for human players with those for AI players.

#### Level

User goal

#### Primary actor

The user

#### Stakeholders and interests

The user wishes to play a game of Chinese Checkers, but does not have the appropriate number of players present.

#### Pre-conditions

The user must have the application installed and running, and have entered the Offline Configuration Activity.

#### Post conditions

##### Success end condition

The user is able to add an AI player to the game, taking the place of a potential human player.

##### Failure end condition

The AI player is not added, leaving the human player in place.

##### Minimal guarantee

The application does not crash, and data on the device remains unaffected.

#### Trigger

The user selects the AI button next to the player name text entry box.

#### Main success scenario

1. The user taps the AI button.
2. The text entry box is replaced by three radio buttons, indicating the desired AI difficulty level.
3. The user taps the desired difficulty button.

#### Variations

In a network game, AI players may be added automatically when a human player leaves.

##### Frequency

The use case could occur multiple times during the configuration of a game. As such, time between triggers could range from seconds to weeks.

##### Assumptions

The user wishes to play with a number of players different than the number of players otherwise available.

#### Special requirements

The device must run Android and have the necessary space to install the game. If the AI player is run from a remote server, the user must be able to connect to the Internet.

#### Issues

1. How are AI players generated and passed in?

#### To do

1. Generate and pass in AI players at game start.

### Finish a game (IT-2-UC-8)

#### Revision history

|  |  |  |
| --- | --- | --- |
| Date | Author | Description of change |
| 2014/02/15 | Ben Stitt | Authored use case. |
| 2014/02/16 | Ben Stitt | Revised to meet documentation syntax. |

#### Use case

Finish a game

#### Id

IT-2-UC-8

#### Description

A player has triggered a win condition, and the players must be informed of which of them has won, and end the game if necessary.

#### Level

User goal

#### Primary actor

Any player in the active game

#### Supporting actors

Other players in the active game

#### Stakeholders and interests

All players in the active game wish to know who has won the game, and if necessary, end it.

#### Pre-conditions

A player has made a move resulting in all their pegs placed in the opposite triangle.

#### Post conditions

##### Success end condition

The user is presented with a dialog box, allowing them to return to the main menu.

##### Failure end condition

The game fails to end.

##### Minimal guarantee

The application does not crash, and data on the device remains unaffected. The user is able to force the application to close if necessary.

#### Trigger

A player’s pegs are all moved into the opposite triangle.

#### Main success scenario

1. The win condition is triggered.
2. The user is presented with a dialog box announcing the winner.
3. The user clicks the button on the dialog box, and is returned to the main menu.

#### Extensions

In a network game, step 2 is only presented to the winner, and the dialog contains two buttons. If the user does not tap the button from Main success scenario 3 then the user remains in the game as a spectator.

#### Frequency

This use case occurs at the end of each game. As such, it could occur hourly to weekly.

#### Assumptions

The user has not quit the active game.

#### Special requirements

The device must run Android and have the necessary space to install the game. To trigger this use case from a network game, the user must have an active Internet connection.

#### To do

1. Clear the applicable game save data.

### Move a peg (IT-2-UC-9)

#### Revision history

|  |  |  |
| --- | --- | --- |
| Date | Author | Description of change |
| 2014/02/17 | Ben Stitt | Authored use case. |

#### Use case

Move a peg

#### Id

IT-2-UC-9

#### Description

The user drags their peg from its current location to a new one.

#### Level

User goal

#### Primary actor

The user

#### Stakeholders and interests

The user wishes to complete their turn by moving a game peg across the board.

#### Pre-conditions

The user must have the application installed and running. The user must have a game active, and it must be their turn.

#### Post conditions

##### Success end condition

The peg is moved to the new location on the game board. This change is reflected in the stored game board state, and displayed to the user.

##### Failure end condition

The peg remains in its current location. No change is made to the game board state, or the change is not displayed.

##### Minimal Guarantee

The application does not crash, and data on the device remains unaffected. The state of the game board is not altered in any other way.

#### Trigger

The user drags the peg the wish to move to a valid position.

#### Main success scenario

1. The user drags the peg.
2. The peg’s location is updated on a temporary copy of the game board.
3. The display is updated to show the peg in its new position.
4. The user clicks the “done” button.
5. The permanent game board is updated.

#### Extensions

In the case that the peg can jump other pegs, steps 1 through 3 can be repeated any number of times.

#### Variations

In the case that an AI player moves a peg, the process occurs as follows.

1. The AI player reports its move to the game state.
2. The permanent game board is updated.
3. The display is updated to show the peg in its new position.

##### Frequency

This use case is triggered whenever a player moves a game peg. As such, it could occur seconds or hours apart.

##### Assumptions

The user is prepared to complete their turn.

#### Special requirements

The device must run Android and have the necessary space to install the game. The user must also have a game active, and it must be their turn.

#### Issues

1. Does pixel-position conversion need to be considered?

#### To do

1. Proceed to the next player’s turn.

### View help pages (IT-2-UC-10)

#### Revision history

|  |  |  |
| --- | --- | --- |
| Date | Author | Description of change |
| 2014/02/18 | Ben Stitt | Authored use case. |

#### Use case

View help pages

#### Id

IT-2-UC-10

#### Description

The user wishes to open the in-game help.

#### Level

User goal

#### Primary actor

The user requires additional knowledge.

#### Stakeholders and interests

The user wishes to play the game in a knowledgeable and efficient manner.

#### Pre-conditions

The user must have the application installed and running.

#### Post conditions

##### Success end condition

The user is able to read through and close the help pages.

##### Failure end condition

The help pages fail to open.

##### Minimal Guarantee

The application does not crash, and data on the device remains unaffected.

#### Trigger

The user taps the icon for the Android-native help and settings menu.

#### Main success scenario

1. The user taps the appropriate menu item.
2. The user reads the help page.
3. The user taps the close button.

#### Extensions

This process may be applied to other types of pages.

#### Variations

These pages may also be viewed from an external web site.

##### Frequency

This use case is only executed when the user requires knowledge of the game. As such, it may only be minutes between uses. Later, the user may never use them.

##### Assumptions

The user is not aware of, or does not know all of the information stored in these pages.

#### Special requirements

The device must run Android and have the necessary space to install the application. The device must give the user access to the help and settings menu.

#### Issues

1. How does Android handle pages of this type?

#### To do

1. Return to normal execution of the application.

### Read the EULA (IT-2-UC-11)

#### Revision history

|  |  |  |
| --- | --- | --- |
| Date | Author | Description of change |
| 2014/02/23 | Ben Stitt | Authored use case. |

#### Use case

Read the EULA

#### Id

IT-2-UC-11

#### Description

The user opens the latest version of the application for the first time. Before they are able to continue, they must agree to the End User License Agreement.

#### Level

User Goal

#### Primary actor

The user

#### Stakeholders and interests

The user wishes to play Chinese Checkers, and Logical Operators would like to protect their intellectual property.

#### Pre-conditions

The user must have the application installed and it must be the first time running it since the last update.

#### Post conditions

##### Success end condition

The user accepts the EULA, and is able to use the application.

##### Failure end condition

The user is unable to see or agree to the EULA, or cannot use the application.

##### Minimal Guarantee

The application does not crash, and data on the device remains unaffected.

#### Trigger

The user starts the application for the first time since the latest update.

#### Main success scenario

1. The user is presented with the EULA, and is prompted to either accept or refuse it.
2. The user taps the “Accept” button.
3. The main menu is displayed.

#### Extensions

1. The user taps the “Refuse” button.
2. The application closes.

#### Variations

The EULA may also be displayed on a web page for the application.

##### Frequency

This use case only occurs the first time a user runs a new version of the application. As such, it may be months or years between uses, or it may only be used once.

##### Assumptions

The user is able to make an informed choice regarding the EULA.

#### Special requirements

The device must run Android and have the necessary space to install the game.

#### Issues

1. Will displaying the EULA affect testing in any way?

#### To do

1. Ensure the EULA is kept up to date for each version released.

## Burn down

### Iteration two progress

Figure .1 Iteration two progress

### Iteration two versus iteration one

Figure .2 Iteration one versus iteration two

## Presentation

The following presentation will be presented during the retrospective meeting on February 27, 2014.

Order follows: left to right, top to bottom.

























## Policies

### End-user license agreement (EULA)

By downloading any application from Logical Operators or Google™ (hereafter referred to as "Google"), installing or using this application or any portion thereof ("Application"), you agree to the following terms and conditions (the "Terms and Conditions").

1. Use of application

a. Logical Operators grants you the non-exclusive, non-transferrable, limited right and license to install and use this Application solely and exclusively for your personal use.

b. You may not use the Application in any manner that could damage, disable, overburden, or impair the Application (or servers or networks connected to the Application), nor may you use the Application in any manner that could interfere with any other party’s use and enjoyment of the Application (or servers or networks connected to the Application).

c. You agree that you are solely responsible for (and that Logical Operators has no responsibility to you or to any third party for) your use of the Application, any breach of your obligations under the Terms and Conditions, and for the consequences (including any loss or damage which Logical Operators may suffer) of any such breach.

2. Proprietary rights

You acknowledge that (a) the Application contains proprietary and confidential information that is protected by applicable intellectual property and other laws, and (b) Logical Operators and/or third parties own all right, title and interest in and to the Application and content, excluding content provided by you, that may be presented or accessed through the Application, including without limitation all Intellectual Property Rights therein and thereto. “Intellectual Property Rights” means any and all rights existing from time to time under patent law, copyright law, trade secret law, trademark law, unfair competition law, and any and all other proprietary rights, and any and all applications, renewals, extensions and restorations thereof, now or hereafter in force and effect worldwide. You agree that you will not, and will not allow any third party to, (i) copy, sell, license, distribute, transfer, modify, adapt, translate, prepare derivative works from, decompile, reverse engineer, disassemble or otherwise attempt to derive source code from the Application or content that may be presented or accessed through the Application for any purpose, unless otherwise permitted, (ii) take any action to circumvent or defeat the security or content usage rules provided, deployed or enforced by any functionality (including without limitation digital rights management functionality) contained in the Application, (iii) use the Application to access, copy, transfer, transcode or retransmit content in violation of any law or third party rights, or (iv) remove, obscure, or alter Logical Operators', or any third party’s copyright notices, trademarks, or other proprietary rights notices affixed to or contained within or accessed in conjunction with or through the Application.

3. Google terms of service and privacy policy

a. Google’s Privacy Policy (located at http://www.google.com/privacypolicy.html) explains how Google treats your information and protects your privacy when you use the Application. You agree to the use of your data in accordance with The Google’s privacy policies.

b. The Application may contain features that are used in conjunction with Google’s search and other services. Accordingly, your use of such features of the Application is also governed by Google’s Terms of Service located at http://www.google.com/terms\_of\_service.html, Google’s Privacy Policy located at http://www.google.com/privacypolicy.html, as well as any applicable Google Service-specific Terms of Service and Privacy Policy, which may be updated from time to time and without notice.

4. Logical Operators terms of service and privacy policy

a. Logical Operators may collect information through the Application in order to build an online profile and establish rankings. Logical Operators will not use this information for any other purpose, and will not share it with any third party.

b. Logical Operators will abide by Google's Privacy Policy as outlined in section 3. In addition, Logical Operators will require you to follow Google's Terms of Service as outlined in section 3.

5. Termination

These Terms and Conditions will continue to apply until terminated by either you or Logical Operators as set forth below. You may terminate these Terms and Conditions at any time by permanently deleting the Application from your mobile device in its entirety. Your rights automatically and immediately terminate without notice from Logical Operators, or any Third Party if you fail to comply with any provision of these Terms and Conditions. In such event, you must immediately delete the Application

6. Indemnity

To the maximum extent permitted by law, you agree to defend, indemnify and hold harmless Logical Operators, their affiliates and their respective directors, officers, employees and agents from and against any and all claims, actions, suits or proceedings, as well as any and all losses, liabilities, damages, costs and expenses (including reasonable attorneys fees) arising out of or accruing from your use of the Application, including your downloading, installation, or use of the Application, or your violation of these Terms and Conditions.

7. Disclaimer of warranties

a. You expressly understand and agree that your use of the Application is at your sole discretion and risk, and that the Application is provided as is and as available, without warranty of any kind.

b. You are solely responsible for any damage to your mobile device, or other device, or loss of data that results from such use.

c. Logical Operators further expressly disclaim all warranties and conditions of any kind, whether express or implied, including, but not limited to the implied warranties and conditions of merchantability, fitness for a particular purpose and non-infringement, with respect to the Application.

d. The Application is not intended for use in the operation of nuclear facilities, life support systems, emergency communications, aircraft navigation or communication systems, air traffic control systems, or any other activities in which the failure of the Application could lead to death, personal injury, severe injury, or environmental damage.

8. Limitation of liability

You expressly understand and agree that Logical Operators, their subsidiaries and affiliates, and their licensors are not liable to you under any theory of liability for and direct, indirect, incidental, special consequential, or exemplary damages that may be incurred by you through your use of the Application, including any loss of data or damage to your mobile device, whether or not Logical Operators, or their representatives have been advised of or should be aware of the possibility of any such loss arising.

9. Miscellaneous

a. These Terms and Conditions constitute the entire Agreement between you and Logical Operators relating to the Application and govern your use of the Application, and completely replace any prior or contemporaneous agreements between you and Logical Operators regarding the Application.

b. The failure of Logical Operators to exercise or enforce any right or provision of these Terms and Conditions does not constitute a waiver of such right or provision, which will still be available to Logical Operators.

c. If any court of law, having the jurisdiction to decide on this matter, rules that any provision of these Terms and Conditions is invalid, then that provision will be removed from the Terms and Conditions without affecting the rest of the Terms and Conditions. The remaining provisions of these Terms and Conditions will continue to be valid and enforceable.

d. The rights granted in these Terms and Conditions may not be assigned or transferred by either you or Logical Operators without the prior written approval of any of these parties. Neither you nor Logical Operators are permitted to delegate their responsibilities or obligations under these Terms and Conditions without the prior written approval of the other party.

e. These Terms and Conditions and your relationship with Logical Operators under these Terms and Conditions will be governed by the laws of the Province of Ontario, Canada without regard to its conflict of laws provisions. You and Logical Operators agree to submit to the exclusive jurisdiction of the courts of the Province of Ontario to resolve any legal matter arising from these Terms and Conditions. Notwithstanding this, you agree that Logical Operators will still be allowed to apply for injunctive remedies (or an equivalent type of urgent legal relief) in any jurisdiction.

# Design

## Class diagrams

### Game board engine

See Appendix F

## User interface control elements

### Main activity

|  |  |  |
| --- | --- | --- |
| Component ID | Control Type | Effect |
| offlineActivityConfigurationButton | Button | Starts the OfflineConfigurationActivity OR OfflineGameActivity (with resume dialog) if game is saved |

### Offline Configuration activity

|  |  |  |
| --- | --- | --- |
| Component ID | Control Type | Effect |
| offlineTwoPlayerButton | ToggleButton | Displays two input fields:   * offlineRedPlayerNameEditText * offlineBluePlayerNameEditText |
| offlineThreePlayerButton | ToggleButton | Displays three input fields:   * offlineRedPlayerNameEditText * offlinePurplePlayerNameEditText * offlineYellowPlayerNameEditText |
| offlineFourPlayerButton | ToggleButton | Displays four input fields:   * offlineRedPlayerNameEditText * offlineGreenPlayerNameEditText * offlineBluePlayerNameEditText * offlineOrangePlayerNameEditText |
| offlineSixPlayerButton | ToggleButton | Displays all six input fields |
| offlineRedPlayerTypeButton | ToggleButton | Human by default; toggles to AI |
| offlineGreenPlayerTypeButton | ToggleButton | AI by default; toggles to Human |
| offlinePurplePlayerTypeButton | ToggleButton |
| offlineBluePlayerTypeButton | ToggleButton |
| offlineYellowPlayerTypeButton | ToggleButton |
| offlineOrangePlayerTypeButton | ToggleButton |
| Offline<COLOUR>PlayerEasyButton | ToggleButton | Note: replace <COLOUR> with the 6 player colours. This has been done to be concise.  Easy difficulty is the default  Selecting another difficulty deselects the current one |
| Offline<COLOUR>PlayerMediumButton | ToggleButton |
| Offline<COLOUR>PlayerHardButton | ToggleButton |
| offlineRedPlayerNameEditText | EditText | Allows for textual entry  Validation:   * Can’t be blank depending on number of human players selected |
| offlineGreenPlayerNameEditText | EditText |
| offlinePurplePlayerNameEditText | EditText |
| offlineBluePlayerNameEditText | EditText |
| offlineYellowPlayerNameEditText | EditText |
| offlineOrangePlayerNameEditText | EditText |
| offlineGameActivityButton | Button | Finishes this activity and starts OfflineGameActivity |

### Offline Game activity

|  |  |  |
| --- | --- | --- |
| Component ID | Control Type | Effect |
| offlineMoveResetButton | Button | Undoes any peg movements since start of current turn |
| offlineMoveDoneButton | Button | Updates game with player’s move  Rotates the board  Changes the current player label at the top of the screen |

### Offline Game activity: resume dialog

|  |  |  |
| --- | --- | --- |
| Component ID | Control Type | Effect |
| offlineAcceptContinuationButton | Button | Dialog disappears |
| offlineDeclineContinuationButton | Button | This activity finishes and OfflineConfigurationActivity starts  Saved game is deleted |

### Offline Game activity: end of game dialog

|  |  |  |
| --- | --- | --- |
| Component ID | Control Type | Effect |
| offlineGameEndToHomeButton | Button | This activity finishes and MainActivity starts |
| offlineGameEndToNewButton | Button | This activity finishes and OfflineConfigurationActivity starts |

## User interface control flow

### Overview

See Appendix G

### Main activity

See Appendix H

### Offline game

See Appendix I

### Offline game activity: resume dialog

See Appendix J

### Offline game activity: end of game dialog

See Appendix K

### Offline configuration activity

See Appendix L

## Visual design

### Introduction

The purpose of this document is to analyse and describe why and how Logical Operators styled the Chinese Checkers interface for iteration two of the application. All visual designs and styles were created after an agreed upon user interface was created for the application. The iteration two visual design process is an extension of the iteration one visual design process with more modifications than new features.

### Colour swatches

#### Light Primary Palette

rgb(255, 173, 176)

rgb(255, 210, 173)

rgb(255, 246, 173)

rgb(173, 255, 203)

rgb(173, 225, 255)

rgb(231, 173, 225)

#### Dark Primary Palette

rgb(197, 4, 9)

rgb(208, 106, 8)

rgb(208, 192, 0)

rgb(0, 128, 40)

rgb(0, 88, 153)

rgb(94, 0, 135)

#### Chosen swatches

The Primary Palette and Monochrome Palette from iteration one are still contain the most prominent swatches used throughout the application. The Light Primary Palette contains the swatches for a lighter shade of a colour from the Primary Palette. The lighter colours will be predominantly used to signify an empty peg space within a player’s home. The Dark Primary Palette contains swatches for a darker shade of colour from the Primary Palette. The darker colours show which peg is currently selected by the user.

#### Rejected swatches

Rejected swatches may be revisited in future iterations, but are not significant enough to this iteration to include in this design document.

### Mock-ups

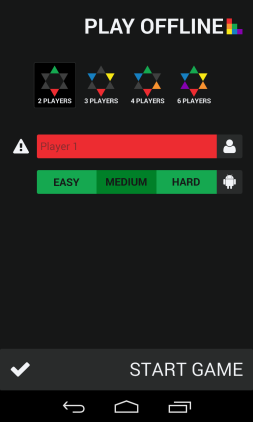


Figure .1 Play offline configuration

See Appendix B for full-scale mock-up image



Figure .2 Play offline game

See Appendix C for full-scale mock-up image



Figure .3 Application dialog

See Appendix D for full-scale mock-up image

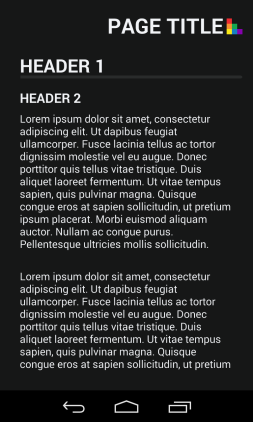


Figure .4 Static page

See Appendix E for full-scale mock-up image

Taking into account the design requirements specified in this document, four mock-up images have been produced that display the layouts of the application for iteration one as accurately as possible:

* The Play offline configuration mock-up has been modified since iteration one to show how a user can select an AI to play against as well. The title of the activity has also been changed to “PLAY OFFLINE” to reflect the change of renaming “hotseat” to “offline”.
* Playing an offline game now shows which peg is selected, which pegs are missing from their home and the possible moves which are outline in a light grey from the Monochrome Palette.
* Application dialogs will be used within the application to give appropriate feedback to the user and allow them to navigate appropriately between activities. The design strictly follows the Monochrome Palette by default, but can be customized with buttons and text flavoured to the other palettes.
* A static page activity has been designed for pages such as Help and the EULA. It has been designed to be bare-bones basic to make way for generalization.

## AI design

# Testing

## Iteration two testing plan

## Integration testing

## Unit testing

## Acceptance testing

## Ease of use testing

# Appendices

## Appendix A

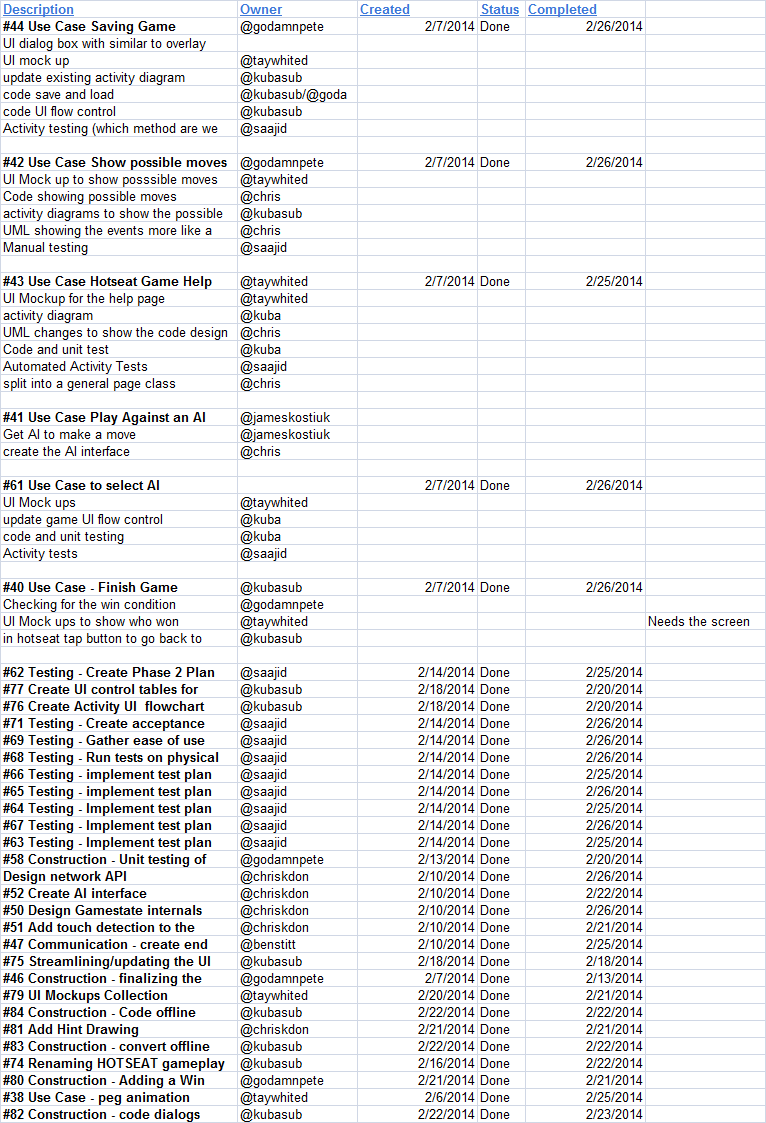


Figure .1 Iteration Two progress from February 20, 2014 meeting

## Appendix B

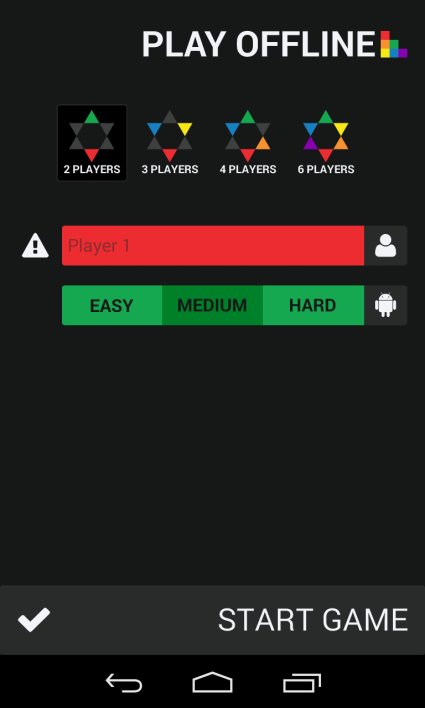


Figure .2 Play offline configuration mock-up

## Appendix C



Figure .3 Play offline game mock-up

## Appendix D



Figure .4 Application dialog mock-up

## Appendix E

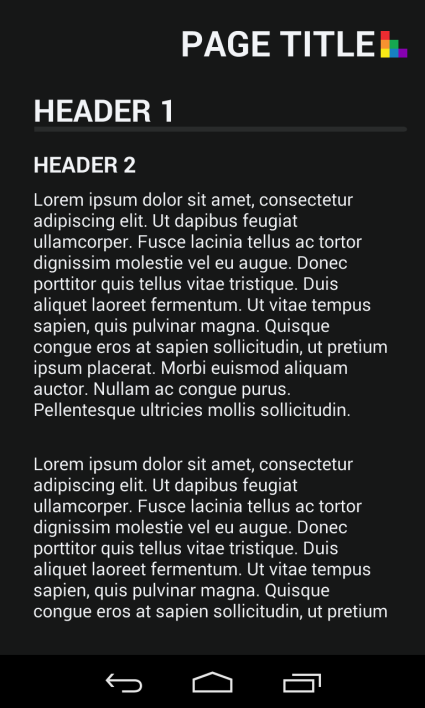


Figure .5 Static page mock-up

## Appendix F

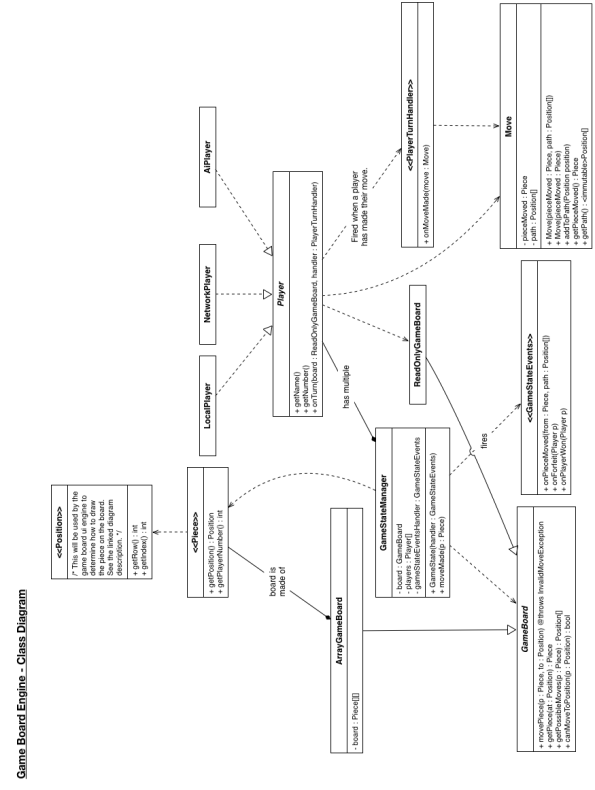


Figure .6 Game board engine class diagram

## Appendix G

# References

Use Case Templates. (n.d.). *TechnoSolutions*. Retrieved January 18, 2014, from http://www.technosolutions.com/use\_case\_template.html