TDT4240

Software Architecture

Spring 2012

Group A17

Android SDK

Steinar Haram

Emil Andreas Mork

Stian Sørebø

Ole Jørgen Rishoff

Implementation Document

Primary focus:

Maintainability

Secondary focus:

Usability

Contents

List of Tables

List of Figures

1 Introduction

This document contains implementation details for our developed version of the classical *Nine Men's Morris* game. The game is developed as a native Android application. The applications primary attribute is modifiability, while its secondary attribute is usability. The second chapter will highlight the design and implementation details. The following chapter contains a user manual, while chapter four contains a brief description of the testing of functional and quality requirements. The relation between the implementation and the planned architecture will be reviewed in chapter five. Chapter six highlights encountered problems and gained experience.

2 Design and implementation details

2.1 Skiller multiplayer framework

Due to the desire of developing a fully functional multiplayer game, the *Skiller multiplayer framework* [?] has been used. This is a third party COTS software, and its usage has sped up the development process. Registration was needed in order to gain access to the Skiller SDK. When the registration was done, a new game could be created, and an application ID, an application key, and an application secret was supplied. These are used in the code to identify the specific application.

This framework supplies a server solution for turn based games, and it has been implemented in the network class. When playing a network game, the GameController class tells the Game model to network class sends event messages to the server, and the server delivers it to the opponent.

2.2 Activities

Figure 1 shows an overview of the application's different activities, and how the user interactions can change them. The user can navigates back to the menuActivity by pressing the back button.

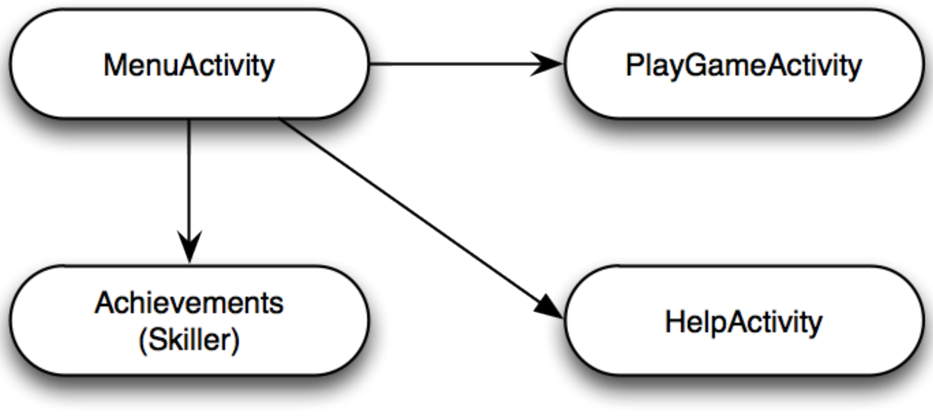


Figure 1: Application activities

The MenuActivity shows a menu consisting of five items, allowing the user to create or join a multiplayer game, start a local game, check achievements, or check the game rules. The PlayGameActivity is responsible for creating or joining multiplayer games, and starting local games. The HelpActivity is responsible for showing the game rules. The achievement screen is supplied by *Skiller*.

Screencaps of the different activities, and the achievement screen, are shown in section 3.5.

2.3 MVC

Our implementation of the MVC structure is based interface communication. In regular Java it is more common to create a MVC structure where the view has model and the controller contains view and model. Android has its own MVC structure where an activity represent controller and view, which makes it difficult to build ut the MVC in a regular way. The most favorable way of createing MVC in android is to use interfaces to update the view when there are changes in the model. Figure 2 shows an overview of the MVC structure, with BoardView (View), GameController (Controller) and Game (Model). When the user interact with the view the controller handles the user action and tells the model what to do. When the model is updated, the view receive a message via the interface, and updates the screen.

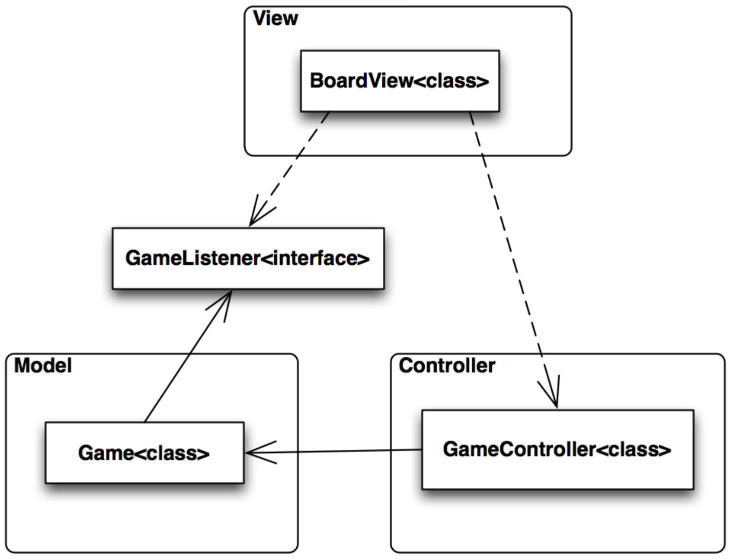


Figure 2: MVC structure

2.4 States

The Game class is implemented with associated states. The states change on runtime depending on player interaction. All states have their own way of controlling user possibilities, and telling the user what to do while playing. Figure 3 shows the relation between the Game class, the State interface, and the context classes implementing the State interface.

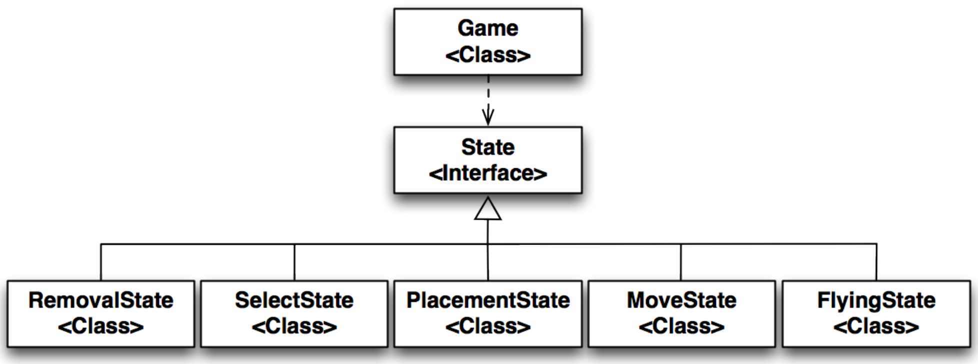


Figure 3: State structure

3 User manual

3.1 Functional requirements

- Requires the user to have an Android device with Android OS v2.2 or newer.
- Requires internet connection to be able to play online.

3.2 Running the application

The project files are supplied with the delivery. The project can be run in an emulator or on an Android device. In both scenarios it is recommended to open the project in Eclipse. It can also be installed directly on an Android device with the supplied *.apk file.

3.2.1 Opening project in Eclipse

Opening the project in Eclipse is done as follows:

1. Choose *File*
2. Choose *Open project*

3. Choose *Existing source code*
4. Navigate to the download path, and open the project

3.2.2 Running

To run the project in an emulator, the user needs to install an AVD, and then choose to run the project with this AVD.

3.2.3 Running on device

The user needs to connect the Android device to the computer, and run the project with the device set as target.

3.2.4 Installing APK-file

The user needs to connect the Android device to the computer, and transfer the APK-file to the SD card. The settings on the device must be set to accept installing applications from unknown sources. The next step is to open the file browser on the device, and click the application file on the SD card. It will then prompt the user to install the application.

3.3 Game rules

The game is implemented with the same set of rules as the classic board game *Nine Men's Morris* [?]. The goal of the game is to either block any opponent moves, or to reduce your opponent's piece number to less than three. If you get three pieces in a row, you enter a morris state, and are allowed to remove one of your opponent's pieces. Pieces that are in a morris state, i.e. forms three in a row either horizontally or vertically, are not removable.

3.4 Creating Skiller account

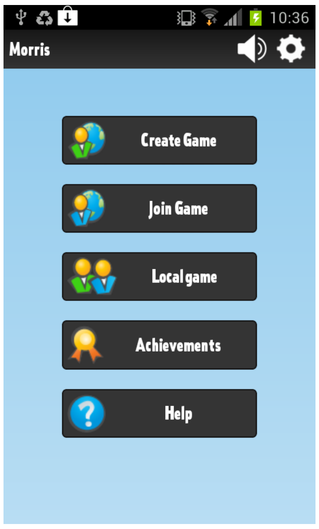
When running the application, the user is prompted to create a Skiller user account. The registration only requires a username and a password. If the user already has an account, he can log in with the existing user information.

The account automatically gets 50 coins that are used for playing online games in the future.

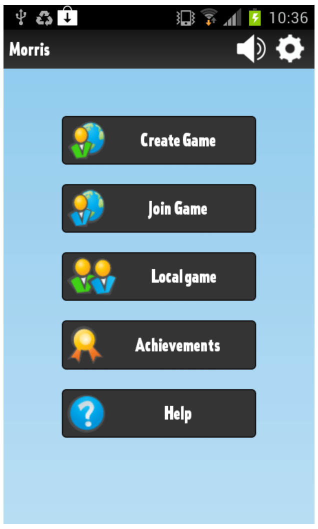
3.5 How to play

3.5.1 Choosing game mode

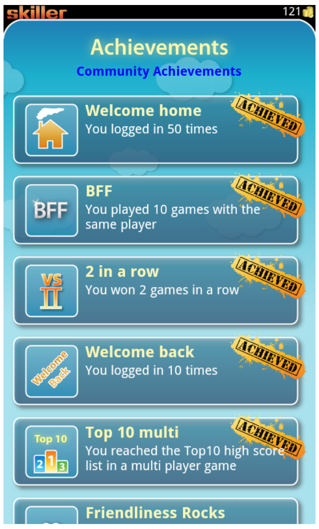
A user can choose between online mode or hotseat mode. Clicking "Crate Game" or "Join Game" will start a game in online mode. Clicking "Hotseat" will start a game in hotseat mode.



(a) Available game modes



(b) Achievements



skiller 121

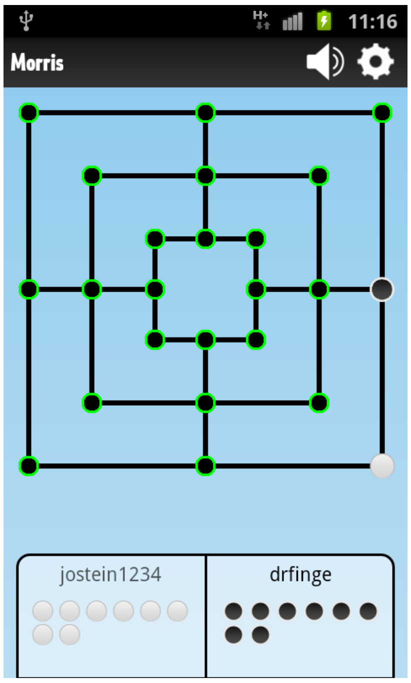
Achievements
Community Achievements

Welcome home — You logged in 50 times — ACHIEVED
BFF — You played 10 games with the same player — ACHIEVED
2 in a row — You won 2 games in a row — ACHIEVED
Welcome back — You logged in 10 times — ACHIEVED
Top 10 multi — You reached the Top10 high score list in a multi player game — ACHIEVED
Friendliness Rocks

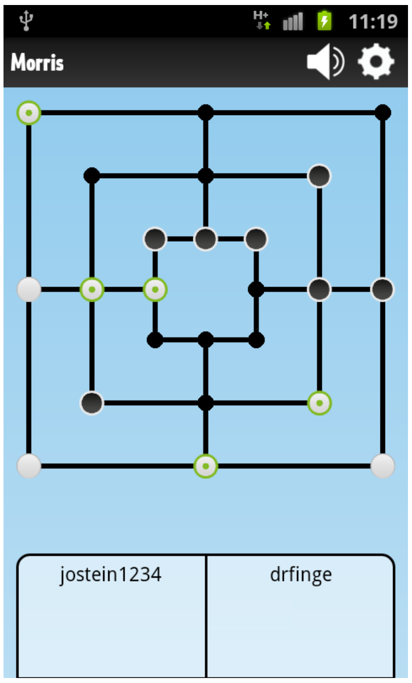
(c) Help explaining the games rules

3.5.2 Placing, selecting, moving, and removing pieces

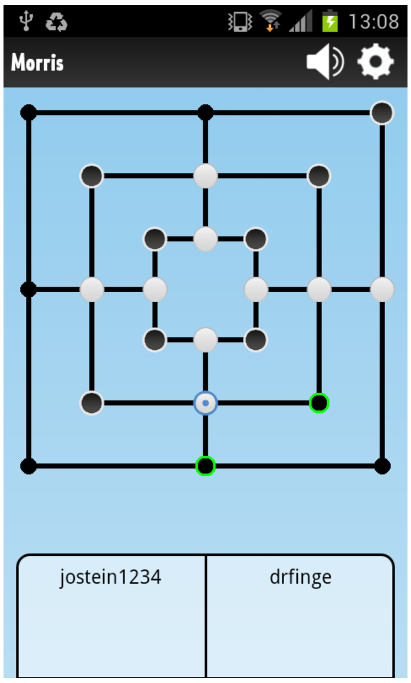
When it is your turn to move, either the board or your pieces will be highlighted. In addition, the name of the current player will be blinking as the game progresses.



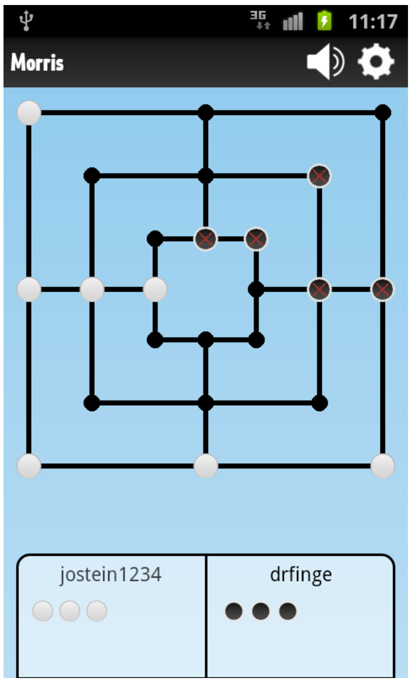
(d) Green indicator shows where you can place a piece



(e) Highlights selectable pieces



(f) Highlight selected piece, green indicator on possible moves



(g) Highlights removable pieces with a red cross

3.5.3 Hotseat mode

If you start a local game as described in section 3.5.1, you can control both players from the same device.

3.5.4 Online mode

If you start an online game as described in section 3.5.1, you are taken to the board screen, and need to wait for another player to join your game. The guest, i.e. the one who joins the game, will get the initial move. Your own pieces will always be white.

4 Relations to the architecture

This section should list the inconsistencies between your architecture and the implementation. Give the reasons for these inconsistencies. Discuss whether they could have been discovered at an earlier point, for instance during the ATAM evaluation.

5 Issues

It turned out that the Skiller framework was poorly documented and it gave some unreadable exception messages. This slowed down the testing quite a bit. In addition, because we normally have only had two Android devices at our disposal, much of the testing have been done in the emulator. This is of course not very effective. Also, when running the application in the emulator and creating games, players would automatically join without user interaction. The Skiller team was unable to give us any good answers to why we experienced this problem.

We underestimated the importance of groupwide understanding of the framework, and the implementation should have been done collectively. As it was, we assigned one person to this task, and the rest of the group were unable to do anything with the framework in his absence. We also should have a done a more thourogh research regarding the use of the framework and its documentation.

5.1 Gained experieces

In future projects we will spend more time researching possibilities when choosing to work with a third party framework. We did not look into the documentation of the framework before implementing it, and if we had, we would probably have chosen a different framework. We will also perform a quick search for problems related to it before making a final decision.

Some members of the group went into this project with more experience than others, developing native Android applications. Due to good communication we have been able to use this experience to our advantage.

We have gotten a better understanding of the patterns we chose to implement. The experience of implementing a MVC pattern in a native Android application has been very useful. The state pattern and its usage was unknown for all of us, but we now have a good understanding of how it can be used. We could perhaps made this pattern a bit more clear, and assign more responsibilities to the different states.