Developers Manual

Welcome On Board.

Table of Content

Getting Started

In order to get started with the code you need to have both GIT and a BitBucket account. Of course you also need the Java Development Kit 1.6 and a suggestion would be to use the IDE Eclipse.

GIT

To get started and install GIT the right way you should follow this guide:
http://git-scm.com/book/en/Getting-Started
When you reach the GIT Basics you are ready to create an account on BitBucket and once your account has been added to our project the following command will get you the project:

```
git clone https://USERNAME@bitbucket.org/Shortleft/chessfeud.git
```

Once you have the project you should now learn the basics of GIT (Read the guide if you haven't used it before.)

Java Development Kit

To download the JDK you simply go over to the site: http://www.oracle.com/technetwork/java/javase/downloads/index.html
and download some 1.6 version. Follow the instructions to install the JDK.

Eclipse (suggested)

We suggest that you use Eclipse when programming in our project, but of course you can use the IDE or Text Editor of your choice, though we can't make sure we can support you.
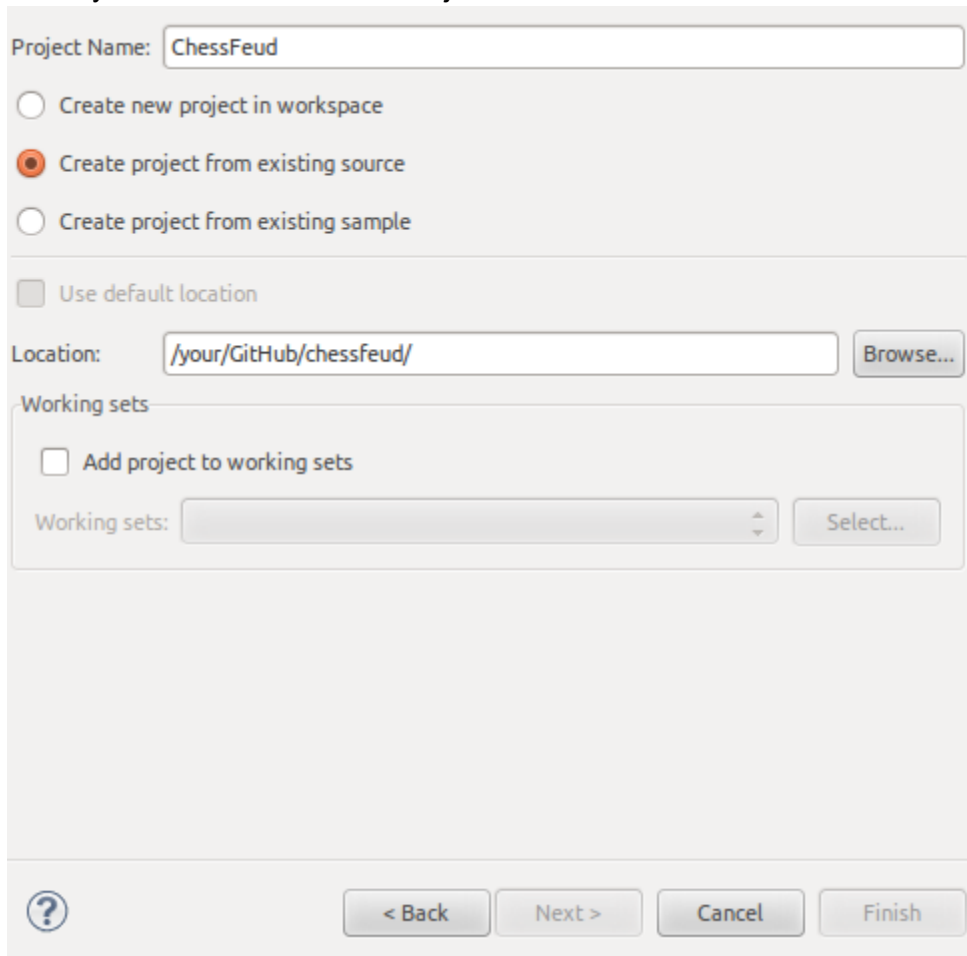
Installing Eclipse:

Simply go to: http://www.eclipse.org/downloads/ and download the latest version of Eclipse. Start the installer and follow the instructions. Once Eclipse is setup you are ready to install the ADT Plugin.

Installing ADT Plugin (Android) for Eclipse:
Follow this guide: http://developer.android.com/sdk/installing/installing-adt.html

To set up the project in Eclipse you need to press File -> New -> Project..
There you will select "Android Project" and fill in the fields like this:

Click Next.
Choose the Android 2.2 version (API Level 8) and click Finish.

Now you should have all the necessary parts and can start getting accustomed to the code.

The Code:

The Big Picture:

The chess game included in the applications follows the MVC-pattern while there still are other part like the DbHandler which is a utility that helps set up the model.

Model:

The model represents a session of chess. It is built up by the core classes ChessModel, Rules and ChessBoard as well as the abstract class Piece. The ChessModel runs the flow of the program and uses the other classes to make program work correctly (following the rules of chess).

ChessModel:
The chessmodel runs the flow of the model and is also what the view reads when making the graphical representation of the model.

To change the model the clicks method is used (which represents a click on the board) and calls the other classes/methods depending of the click.
If it is a click when no piece is selected it selects the piece (if there is a piece of the right team there) and reports the possible moves. If a piece is already selected it will (if possible) move there or otherwise select the piece at the new position (if same team) or just deselct the piece.

ChessBoard:
The chessboard is basically a 8x8 matrix with Objects of the class Piece. It has functions to get the Piece at a given Position and get width/height for easier loop-throughs. It also contain the move-method since the pieces are moved on the gameboard.

Rules:
The Rules class are a class with static methods, it works like a util-class for the game model to help determine what is going to happen. The most important methods are isDraw, isCheck, isCheckmate which returns booleans depending on the state of the chessboard and also getPossibleMoves which is the core method for determining where a piece is able to move.

Piece:
Piece is an abstract class which basically holds the team and id for the different pieces. All chess pieces are subclasses of Piece and the only changeable thing between them is the algorithm for how they move.

Views:

The different views of the program are built up with xml, as is standard in Android. Every view has an activity that binds the different buttons etc to the code.
Here are the current view:
Game Overview - menu_main.xml - MainActivity.java
        The first view to get launched is currently what is called the Game Overview. Here a list of all current and recently finished games will pop up along with invites etc. The user is able to start a new game and reach the settings/about from here.
The Chess View - activity_play.xml - PlayActivity.java

The view of a current game is called The chess view; here you will see the actual game and its state along with some information about the game (who you are playing against, taken pieces etc). The chess game is called GameView and is implemented in the activity_play code.
Settings - activity_settings.xml - SettingsActivity.java
This is the view for the settings. Here you can see and change your game settings.
About - activity_about.xml - AboutActivity.java
Here you can read about the author and contributors. If you are good enough, maybe your name will be here one day.
Statistics - activity_profile.xml - ProfileActivity.java
Here you can see your statistics.

Utils:

In order for the application to work online a server is needed, this makes a DbHandler in the application needed as well. The DbHandler handles all the sending and receiving to and from the server. It is the only thing that is allowed to connect to the server and it doesn't know anything about the rest of the program to get rid off circular dependencies.
The DbHandler is used by the Game Overview to get all the games and it is also used when logging in/registering to make sure the user is valid.

The End