

User Manual

Mancala

For Systems Modeling



1. Introduction

Mancala game is developed using Java Swing and Fujaba Modeling.

2. Features

- **Start New Game**

The user start to play a new game.

Player 1 will be asked for the name.

Player 2 will be asked for the name.

Player randomly gets a first move in the game.

- **Reset Game**

- **The user start to play a new game.**

- **Player 1 will be asked for the name.**

- **Player 2 will be asked for the name.**

- **Player randomly gets a first move in the game.**

- **Instructions**

Player can refer to the Instructions any time while the game is in progress or after the game is finished.

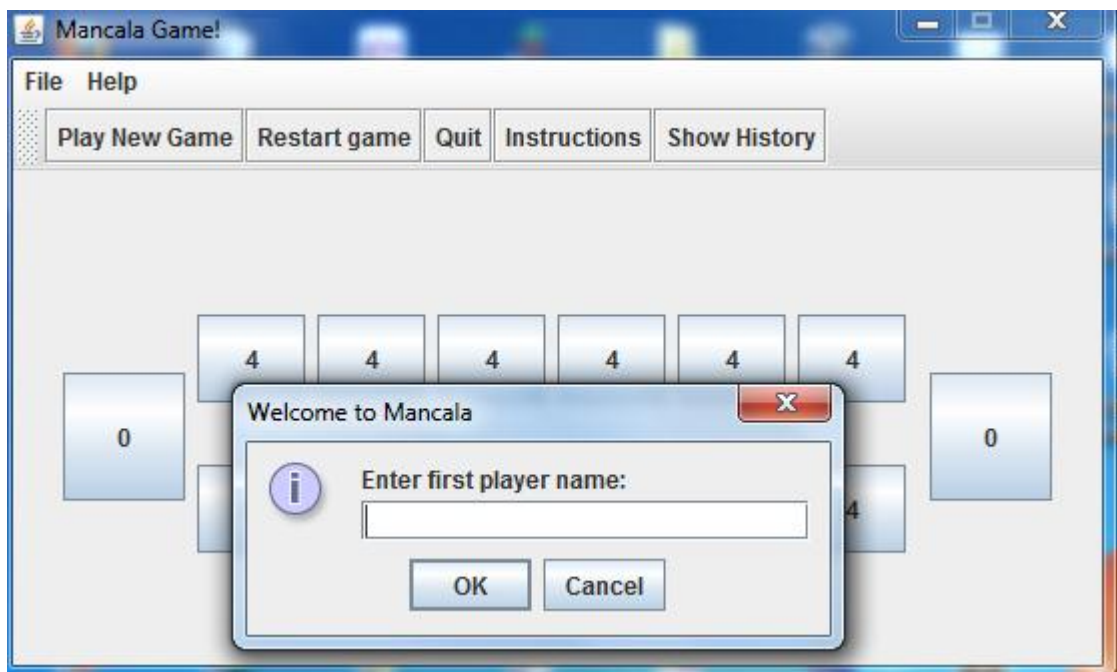
- **Show History**

Player can see the history at any time while the game is in progress or after the game is finished.

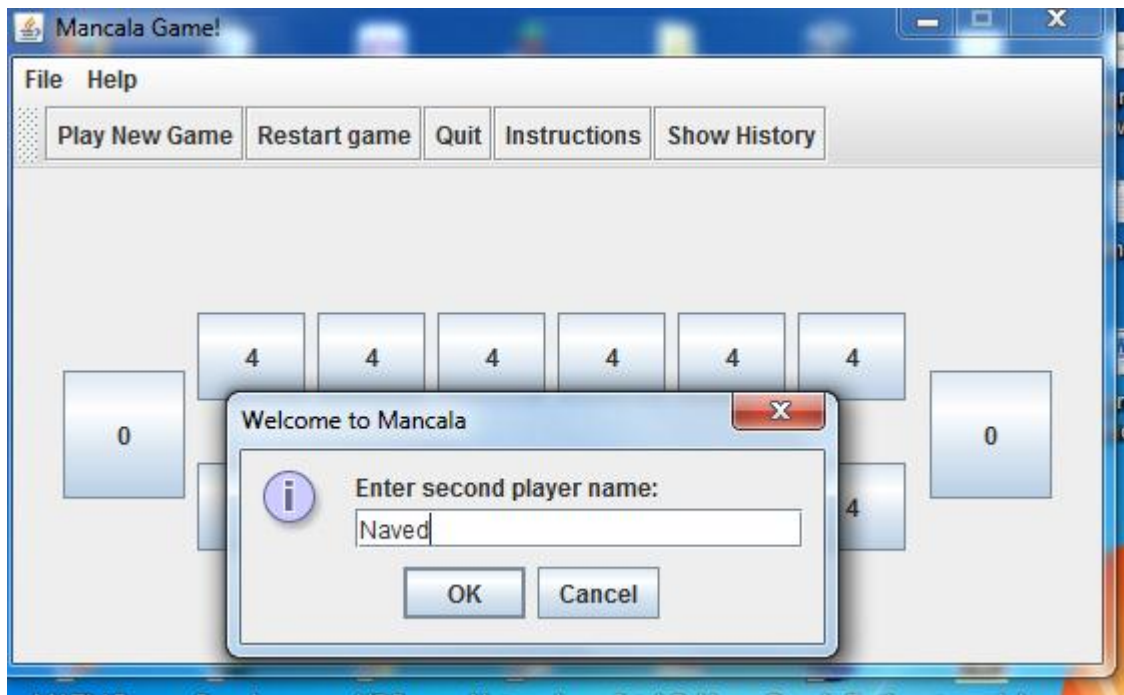
3. Setting up the game

User just need to start the game and the game will ask for the player names and then the Mancala Board will be displayed in front of the Player and they can start playing the game.

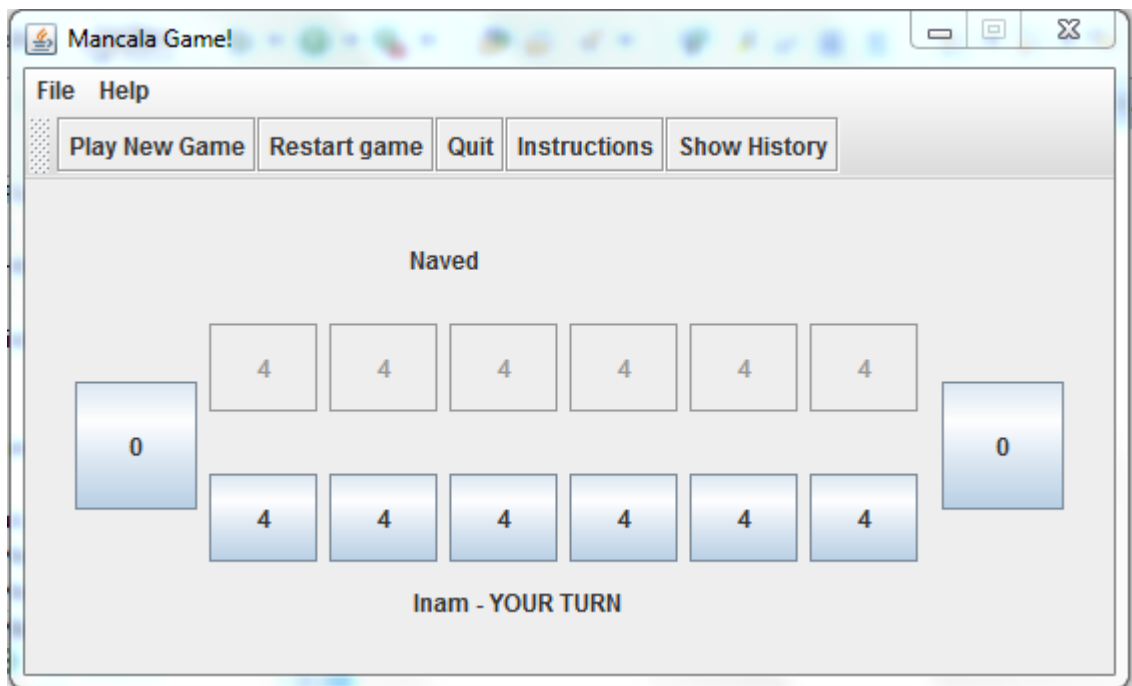
4. Game start up and figures



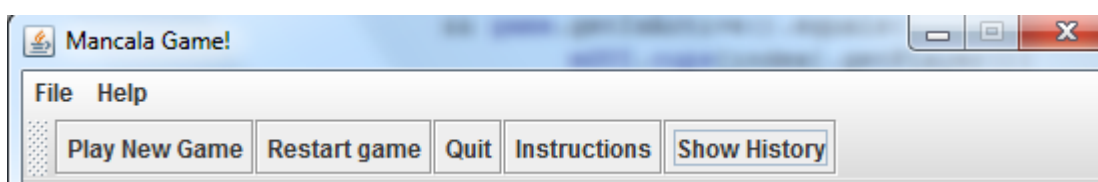
Figure(1) asking first player name



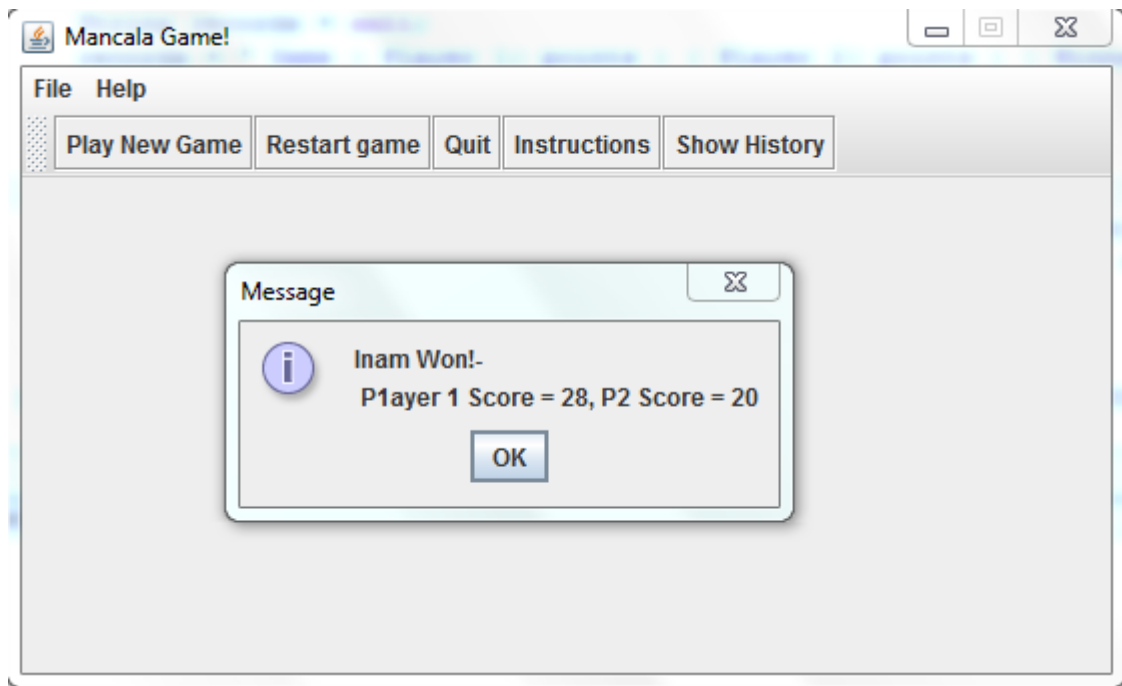
Figure(2) asking second player name



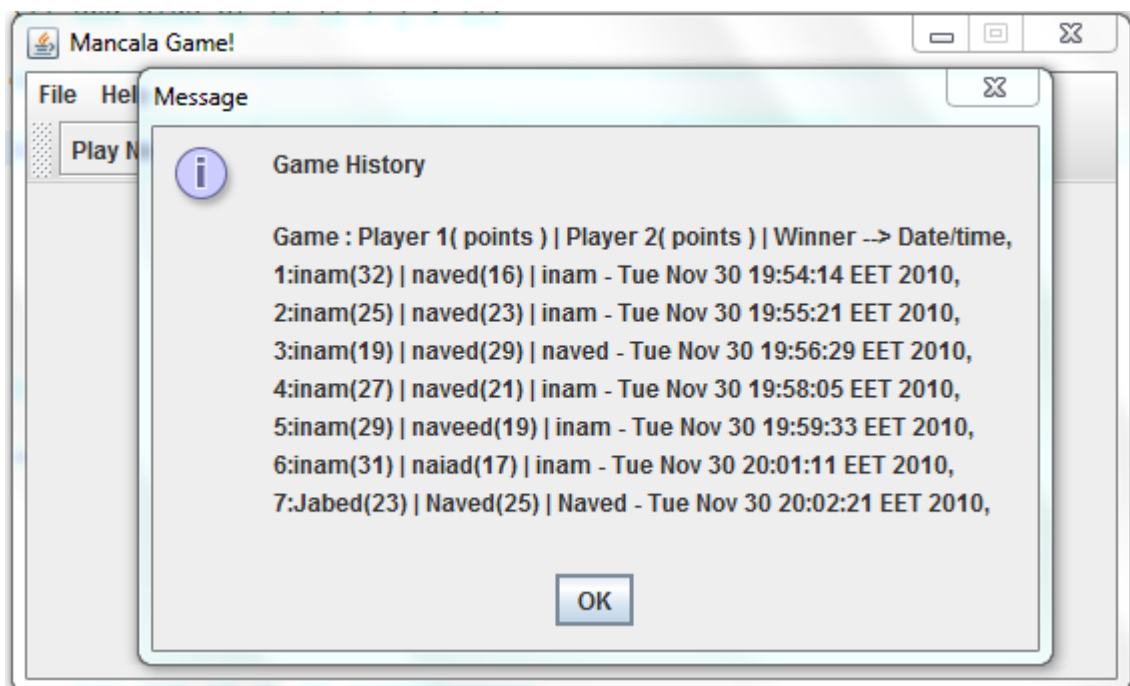
Figure(3) Main User Interface of the game showing the mancala board



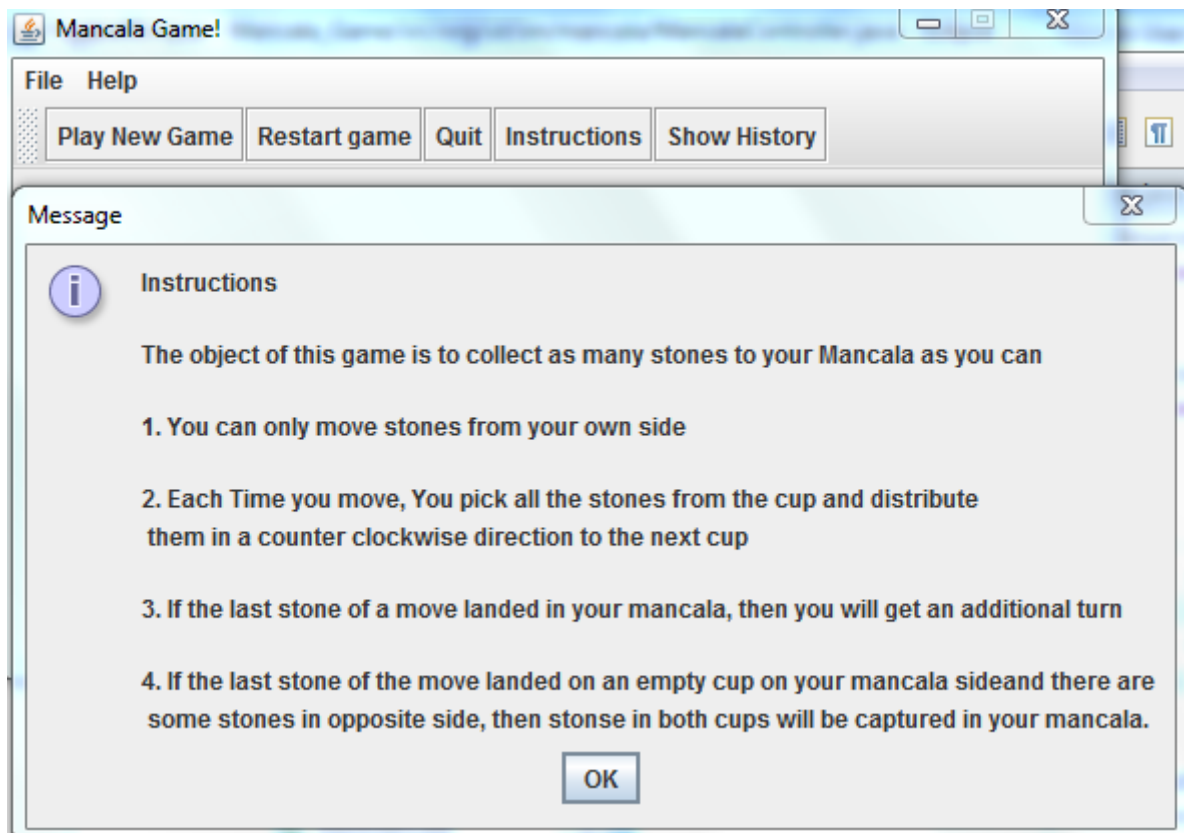
Figure(4) Mancala game toolbar features



Figure(5) Mancala Game winner



Figure(6) showing the game history



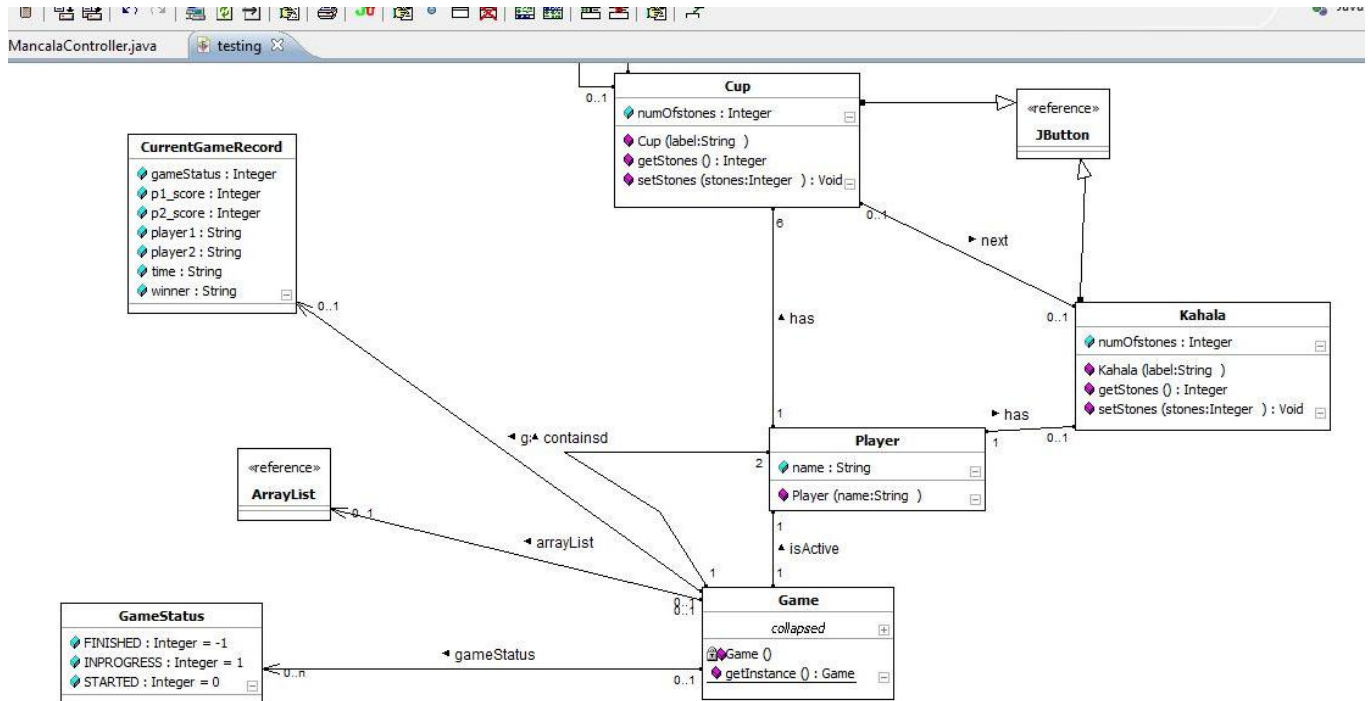
Figure(7) showing the instructions

5. Contact Us

- Naved navedz@gmail.com
- Inam Soomro inam@ut.ee
- Jabed Morshed Choudhry jabedmorshed@gmail.com
- Naiad Khan eyeslave@gmail.com

Code documentation

In our Mancala project we have implemented MVC architecture. For Model we have used Fujaba to generate the java classes and our classes diagram using Fujaba is shown below:



As you can see in the diagram we have 8 classes:

Game: Game is the main class and we have implemented as singleton which contains 2 players. Game class also has an ArrayList this array will maintain the history of the game session played and will display by pressing the show history button. This ArrayList contains the object of CurrentGameRecord class which will save object for each game session played. As shown in the figure this will have information about two players their and their scores additionally it also contains the name of winner and the time when the game is finished. We also tried to implement GameStatus as enum but couldn't do it in fujaba.

We didn't make the Turn separate object because we didn't need it and represented this by making an isActive association with player which represents the turn of the player and it will be changed from controller of our project.

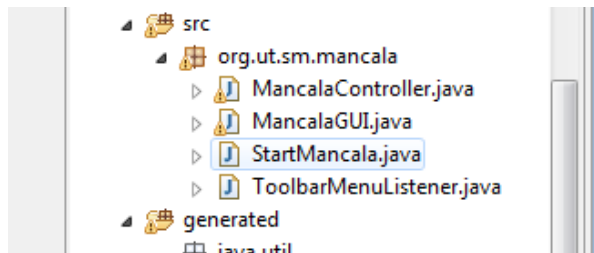
Player: Player has one Kahala and 6 cups this is represented by the association with Kahala and Cups. Each player will have 6 cups and 1 Kahala this is represented by the cardinalities as shown in the figure. Additionally player has the name attribute to represent its name.

Kahala: We have 2 kahalas one for each player Kahala. It is derived from swing JButton class where we will use this for the purpose of button. This class has number of stones and methods to convert the

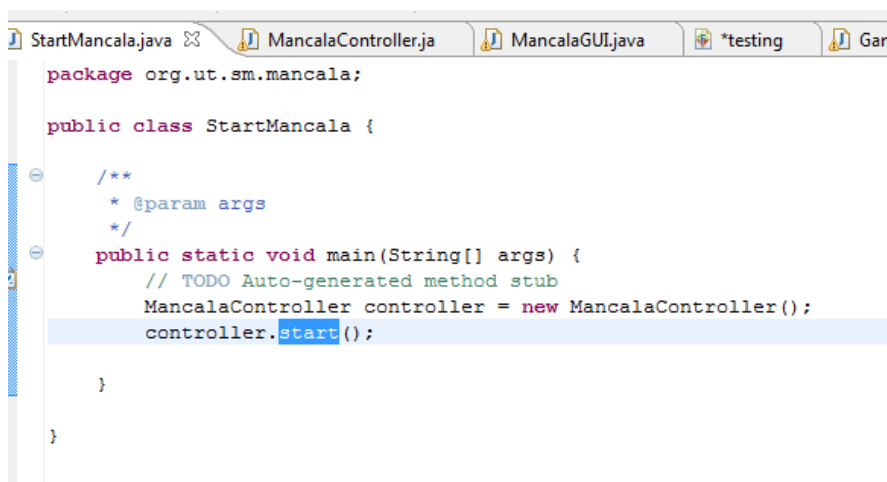
numeric value to text string because to display on button we need to do it. Kahala is possessed by the player and will be next to his cup which represented by the next association between Cup and Kahala.

Cup: Cup is also derived from JButton class of java swing and will be used same as Kahala. Each player will have 6 cups and initially each will have 4 stones when the game is setup for players. Cup will be possessed by player each player has 6 cups. one cup will be next to other cup which represented by self association next.

Other than Fujaba generated Model classes we have implemented controller and view in plain java classes as shown in the figure below:



Start Mancala is the main file to start the game code can be seen in the below screen shot:



This class contains main method to start the game by calling the start method of MancalaController. MancalaGUI is the view it create the necessary GUI for playing game by calling executing its initComponents() method. It will set the Cups, Kahalas, and MenuItem and several buttons.

MancalaController is the controller of our game where we have implemented more or less complete logic for our Mancala game. When two players join the game randomly first turn will be assign to one of the player. After setting cup for each player we have added action listener to the components. The actionPerformed method of this class is heart of our project which contains all the functionality of our Mancala game and other listeners are implemented in ToolbarMenuListener.