

Mancala

Aim

The aim of our project Mancala is to apply and practice the system modeling skills acquired during the correspondent lecture, which we wanted to consequently implement.

Our interest was to employ these techniques to create a simple yet powerful Mancala application that would be fun to play. We are confident we reached this goal.

Team

Our team consists of Samet Erap, Gert Palok, Irma Sauga, and Dominik Schöler. We met each week on Thursday at 18pm. In order to let us work effectively in these meetings, we decided about what would be next to do and then prepared at home. Team roles are explained in the project guide document.

Samet Erap

In this project; we splitted the tasks as we have done in the previous homeworks. What I did are; some part of user stories and documentation of these in my part, some part of object diagrams and documentation of these in my part, some part of use cases and documentation of these in my part, creating the user manual and rules.

Gert Palok

I did a little bit of this and that, mostly modeling in Fujaba and programming. I learned that Fujaba in its current state is too unstable for any work, including educational purposes. Having your model's file size grow exponentially and corrupt regularly, combined with frequent crashes of either Eclipse or the operating system cause a lot of frustration.

Irma Sauga

In this project I mostly worked with documentation and did a bit of modelling in fujaba also. I contributed to diagrams, user stories and scenarios, use cases and project report. I have to agree with my fellow teammember, that Fujaba has got a lot of bugs, that disturb working with the program.

Dominik Schöler

I organized things and distributed the tasks so everyone had something to do. At the beginning I did some parts of the Fujaba modeling and I created the GUI. I did a lot of reviewing and optimizing to the user interface to make it nice and smooth. Finally, I helped on creating some of the static documentation.

I agree with Gert that Fujaba is not ready for use by anyone. The idea may be great, but in this implementation it is pure horror. The interruption of several other home works was not nice either. Nevertheless, I liked this project because it was not too large.

Tools

The main tool we used for developing Mancala is Fujaba. Additionally, Dia supported us in drawing diagrams and IntelliJ would let us create the graphical user interface. To keep us synchronized, we used a Skype group chat and to keep the data consistent we relied on git. In the final phase, pivotalTracker supported us in keeping track of tasks left to do.

Procedure

For the development of Mancala we followed the given guidelines. That way, we started creating scenarios with user stories, which we put into object diagrams. Building these, our class diagram already emerged because it made drawing object diagrams in Fujaba much easier.

We then sorted written user stories into use case diagrams to determine which processes players had to pass to use the application and/or play the game. While in this process, new ideas for user stories arose and were added. During modelling we also improved our class diagrams and object diagrams.

Then we played along the reseeding to define the method and looked what happened when each pit would be reseeded.

Finally, when more important parts of the modelling were done, implementation of the methods was started. Then we developed, applied design patterns and refined documents if necessary.

While in this process, new ideas for user stories arose and were added.

use object game or note-pad test to define methods, implement methods. Iterate, refactor, use/apply design patterns, refine documents from the previous steps.

Timeline

14. Oct

Initial meeting. Decide what has to be done, when it should be done and who would care for it.

Oct

plan more detailed

Oct

user interface draft

Nov

design and implement reseeding algorithm (not finished)

Nov

Dealing with state machine. Lots of trouble with ArgoUML

Nov

Deadline for state machine. This week, we had no time to work on the course project.

25. Nov

30. Nov

Deadline. Delivery of Mancala, all work is done.

Guide for the repository

1. The main class diagram is under the project.ctr – Diagrams – Class Diagrams – Main.

2. The object diagrams are under the `project.ctr` – Diagrams – Activity Diagrams – You will see the user stories and the object diagrams which are connected to user stories.
3. Our object diagrams are the same with story boards.
4. Under documents folder we have multiple files containing corresponding information – diagrams, report (this file), game rules, used terms, user scenarios, use cases, user manual and project guide (leading through some important 'code paths' and showing how we implemented MVC).
5. Folder `src` contains game source code.
6. Game application can be started from `Mancala.jar`
7. Game history is saved to `mancala_hist.txt` file
8. `Mancala.iml` contains design of the application (in IntelliJ IDEA)