Guide through the repository

Mancala project repository consists of three folders: docs, model and src.
Docs folder includes all the documentation.
Model folder includes mancala game class diagram, game scenario, user stories, test cases and object_diagrams folder. The last one has object_diagrams file, which explains why the diagrams are the way they are, and two folders dia and png, where all the diagrams to the user stories are (in .dia and .png format respectively).
Src folder includes source code of the game.

Management information

Introduction

The purpose of this document is to give an overview of progress made in the development of course project Mancala, along with notes of problems encountered and conclusions made. The project was started on the 25.October and finished on the 30.November.

Goals and results

Our main task was to design and implementation the game Mancala by using object first design method. The biggest part was done but some user stories weren't written, because we haven't got enough ideas for that (Mancala has a lack of possible game variations and only a few strategies for playing). Because of that not all user stories are from scenario. Code and documentation are done. Test cases are written. Working Mancala game presented.

Detailed results in the large scope

At the start of the project we decided to use Google docs for the textual part (scenarios/user stories/use cases) but later we decided to use git because its more flexible and powerful tool.
First 5 use cases we took from exercise 1 with Ulrich comments, modified use cases accordingly and divided them into more use cases. During our meetings we discussed about different possibilities:
- regular application (standalone or web);
- command line application;
- nanomancala (holes and stones are not visible, players know initial number of stones in each hole and take turns naming the hole they play, keeping the whole board state in mind);

We explored different Mancala game strategies, for example:
http://fritzdooley.com/mancala/5_online_guide_mancala_strategy.html
Then we selected programming environment, developed GUI and choose standalone application. We decided to use e-mail instead of bug tracker because of small project size, and because it will take more time to choose and install bug tracker (github solution is not good for us).
Also we created Google group to make our communication much more easier. To write scenario we used UNIX Mancala command-line application where we played a complete two-player game without AI, trying to reproduce all possible significantly different game situations, wrote down all turns.
Mancala game is much more interesting when playing against another person than AI and thus it is

more people oriented. For that reason we decided to avoid AI implementation.

We made class diagram in Fujaba. For test cases we preferred to use other way because:
- in Fujaba it will be necessary to draw a lot, which is a bit complicated. It is easier to modify a bit or just insert object diagram made with dia.
- Development was started while modeling wasn't finished.

Most documentation was done by Natalja Timofejeva and Anastassia Soikonen and code development by Olga Agen and Anna Agafonova. Anastassia used gmail group to commit files because git uploading didn't work (so most of the files, that Anastassia wrote, Natalja uploaded to git for her).

The table shows what tasks and when they were done:

Task	Start Date	Author	Comments
Scenario, user stories, object diagrams	5. November	Natalja Timofejeva,Anastassia Soikonen	Scenario, 20 user stories, 32 object diagrams in .dia and .png format done
Class diagram	9. November	Anna Agafonova	Class diagram in Fujaba done
Test cases	15. November	Anna Agafonova, Natalja Timofejeva	Not in Fujaba, 16 test cases
Code with documentation	20. November	Olga Agen	Code and Javadoc created
User manual	24. November	Natalja Timofejeva, Anastassia Soikonen	
Project presentation	28. November	Anna Agafonova, Olga Agen	
Project guideline	30. Novemer	All team members	Project presented

Risks

- Crew scheduling
  Since a large number of other substances, homework deadlines and midterm periods overlapped with the project, it was hard to find enough time to finish all goals on time. All team members have jobs and that also takes a lot of time.
- Conflicts in the team
  There wasn't any conflicts in our team. All team members met in different places to discuss our project and to put new tasks for everyone. Also we used Skype and Google group messaging for solving problems. Google group messaging is very helpful, when you need everyone to be informed about some project aspect.
- Hard to find ideas for user stories
  Mancala has a lack of possible game variations and only a few strategies for playing. Because of that not all user stories are from scenario.

- Little experience in writing test cases in Fujaba
  Therefore we used another tool

Summary

The most part of course project is done. There are certain flaws we are aware of, but all in all we believe we were able to cover all the required modeling, coding and documentation despite shortages in time and experience.