Requirements and Analysis Document for the Challenge Accepted project, (RAD)

- A digital version of 'Upp till Bevis'.

Contents

Version: 2.0
Date: 2013-05-21

Authors:

Isabelle Frölich	900831-2846
Madeleine Appert	891110-4845
Johan Gustavsson	871024-7134

This version overrides all previous versions.

1. Introduction

This is a java application that in real life, is a board game called 'Upp Till Bevis'. During this project, we have created a digital version of that game. The players accomplish missions in order to move forward on the game board. The game ends when a team reaches the goal tile.

1.1 Purpose of application

With our application we aim to construct a computer based version of a game called '*Upp till bevis',* which is a board game requiring the teams playing to complete missions in order to continue forward on the board, towards the finish line.

1.2 General characteristics of application

The application will be a desktop, standalone (non-networked), multi-player application with a graphical user interface.

The application will be turn-based, and the turn switches as soon as the results from a mission has been registered. The next player is chosen by the application based on the order that the teams registered to be a part of the game.
Although there are no time restrictions for how long a turn might be, there are time restrictions regarding how long a team has to finish a mission. The application will end, only if it is manually shut down. If the game is cancelled, nobody wins. The game relies on the users to be truthful, as it cannot determine the success of missions. There is a time limit of 30 seconds to finish a mission.

1.3 Scope of application

The application requires 2-8 teams. Each team must consist of at least 2 people, but there is no upper boundary. There will be no computer-player available. Application will not keep track of high score etc. but it does show whose turn it is.

1.4 Objectives and success criteria of the project

1. The graphical user interface is to be of such quality that the user at all times knows who's turn it is, what stage is present and what he/she is supposed to do.
2. This digital version of a real life board game, should implement the 'challenge tiles' that the real game offers.
3. There are to be the same 4 categories that the real life version offers

1.5 Definitions, acronyms and abbreviations

1. GUI, graphical user interface.
2. Java, platform independent programming language.
3. Turn, all teams will have their turns. A turn begins when a team is either able to make a bet, or choose an opponent for a challenge. It is over when all missions connected to that turn have been finished, and the results have been registered.
4. Mission, assignments that the players need to complete in order to move forward.
5. Category, defines what kind of Mission the player(s) are to accomplish.
6. Challenge, the case where a player landed on a Challenge-Tile at the end of his last Turn. The Team is then required to challenge another paying team to a challenge. The team which is the most successful during their Mission, wins. In the case of a draw, the challenged team is the winner.

2. Requirements

2.1 Functional requirements

The players/teams should be able to:

1. Access the game rules, from both the main menu and the board.
2. Start a new game.
   a) Select how many teams will take part in the turn.
   b) Assign names to the different teams
3. Do a normal turn. During a turn the team will be able to:
   a) Choose how many steps they want to bet.
   b) Change that bet before the actual mission starts.
   c) View the assigned cards.
   d) Choose to see the next card within the time limit.
   e) At the end of a team's turn the other playing team(s) will be able to approve/disapprove the mission.
4. Do a challenge turn. During the turn the team will be able to:
   a) Choose which other team they would like to compete against.
   b) Change the opponent team before the challenge actually starts.
   c) View the assigned cards.
   d) Choose to see the next cards until the deque for that mission has been gone through.
   e) Report their success.
   f) Opposing team can view the assigned cards for their mission.
   g) Opposing team can choose to see the next cards until the deque for that mission has been gone through.
   h) Opposing team can report their success.
5. Exit the application. Will end turn, round and game.

2.2 Non-functional requirements

2.2.1 Usability

An arbitrary player is to understand how the game works without being forced to seek information elsewhere than in the application itself. Therefore the rules will be available to find the application. The language will be in English, hence the game requires all participants to speak English

2.2.2 Reliability

N/A

2.2.3 Performance

Any actions initiated by a player should not exceed a 2 sec response time.

2.2.4 Supportability

There are no supportability requirements regarding a modifiable GUI to suit other platforms.

2.2.5 Implementation

To achieve platform independence the application will use the Java environment. All hosts must have the JRE installed and configured. The application needs to be installed on all hosts where it will run.

2.2.6 Packaging and installation

The application will be delivered as a jar file. To be able to run the program, the user will simply have to have a Java 1.6 program installed on his/her system.

2.2.7 Legal

There could be legal issues regarding rights to the "Upp Till Bevis" game and trade mark. This is not covered here.

2.3 Application models

2.3.1 Use case model

See APPENDIX for UCM diagram and textual descriptions.

2.3.2 Use case priority

1. NewGame (high)
2. DoNormalTurn (high)
3. DoMission (high)
4. Bet (high)
5. DoChallenge (mid)
6. ShowNextCard (mid)
7. MoveForward (mid)
8. MoveBackward(low)
9. ShowRules (low)
10. EndGame(low)

2.3.3 Domain Model

See APPENDIX.

2.3.4 User Interface

Application will use a fixed GUI (non-themeable, non-resizable).

2.4 References

General explanation of the game (in Swedish).
http://www.braspel.com/?id=317

Thorough explanation of game rules (in Swedish).
http://www.braspel.com/filearchive/1/1917/rules_UTB%20new.pdf

3. Possible Future Directions

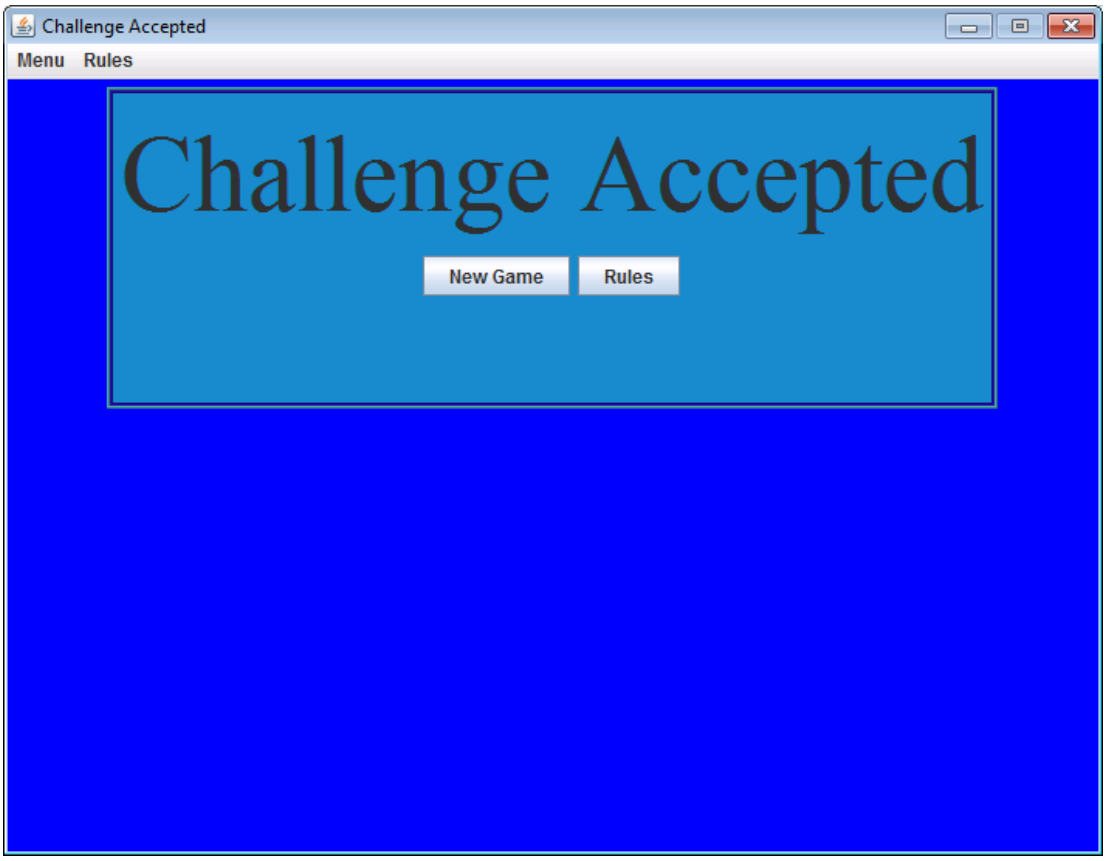
In case of extra time to improve application and add extra features, possible extensions could be:

- Add various levels of difficulty, which for example could consist of different time limits for different teams. In case of teams being of very different ages, this might be very desirable.
- Have clues.
- Possibility to save game.
- Have all the words in the Backwards category, read aloud by the application.

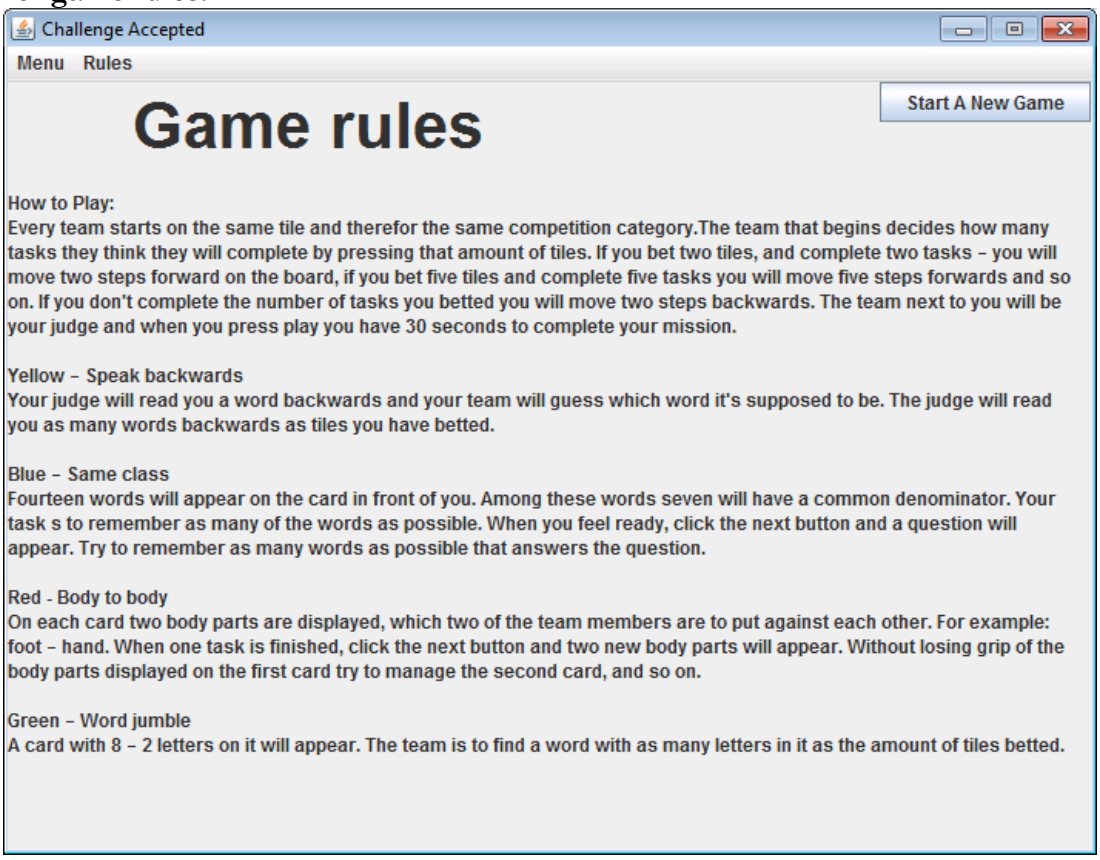
APPENDIX

GUI

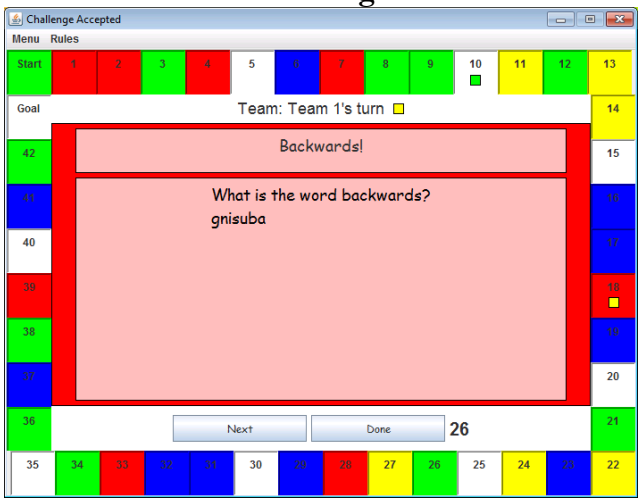
Start screen

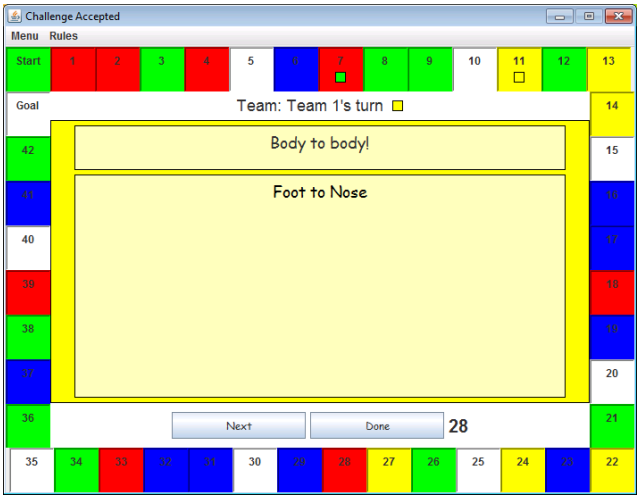


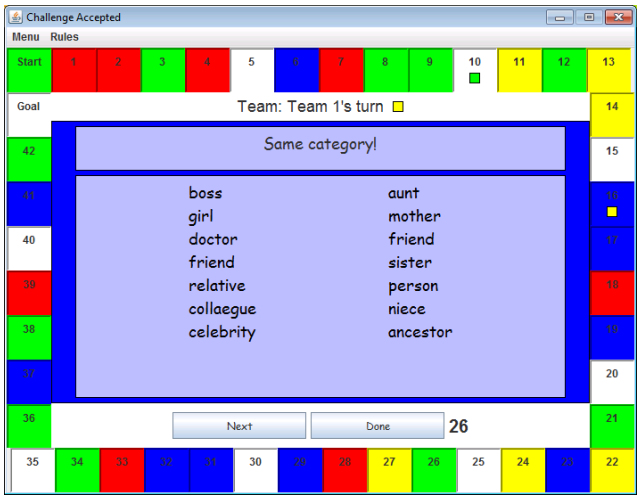
Screen for game rules.



Missions for different categories.

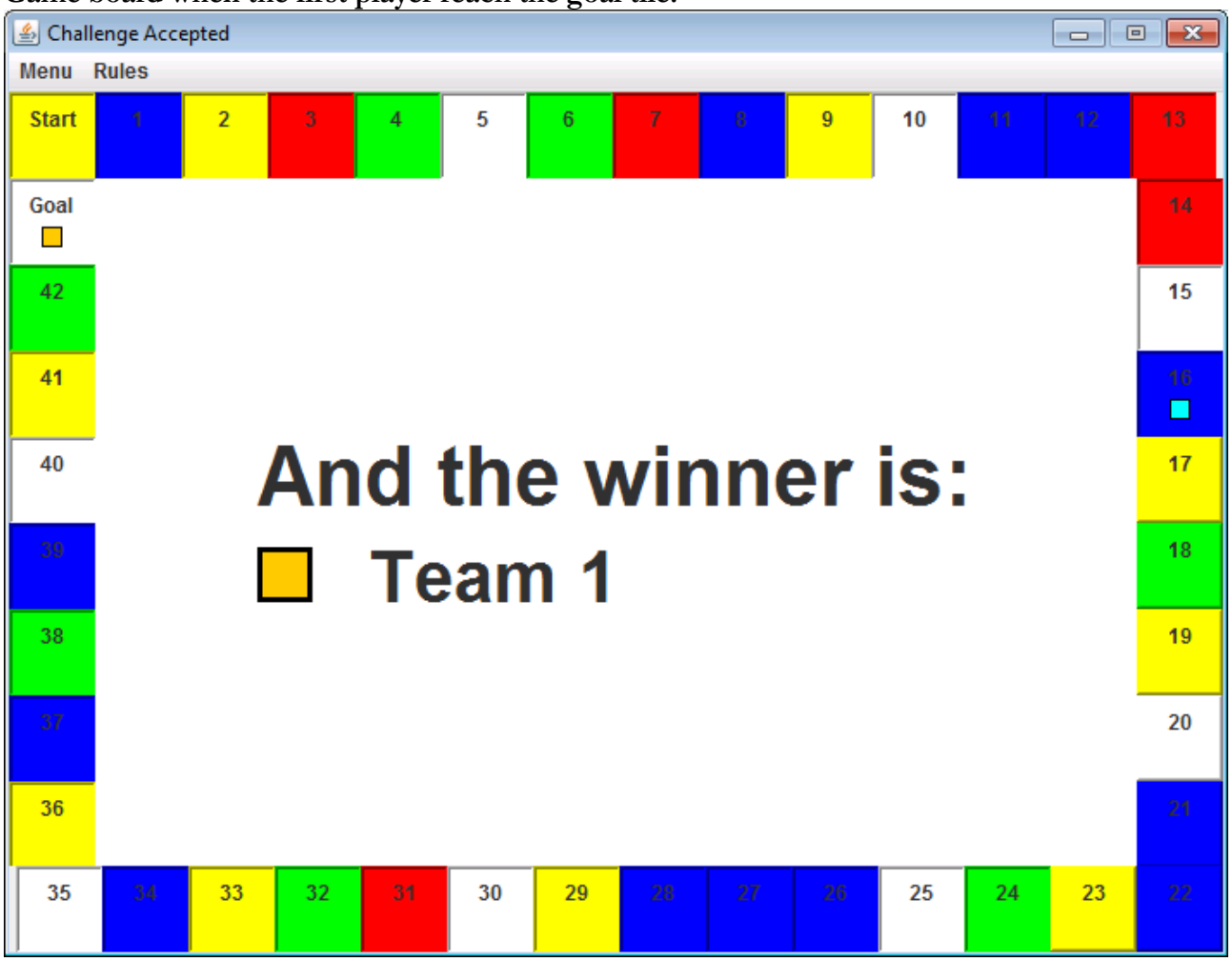




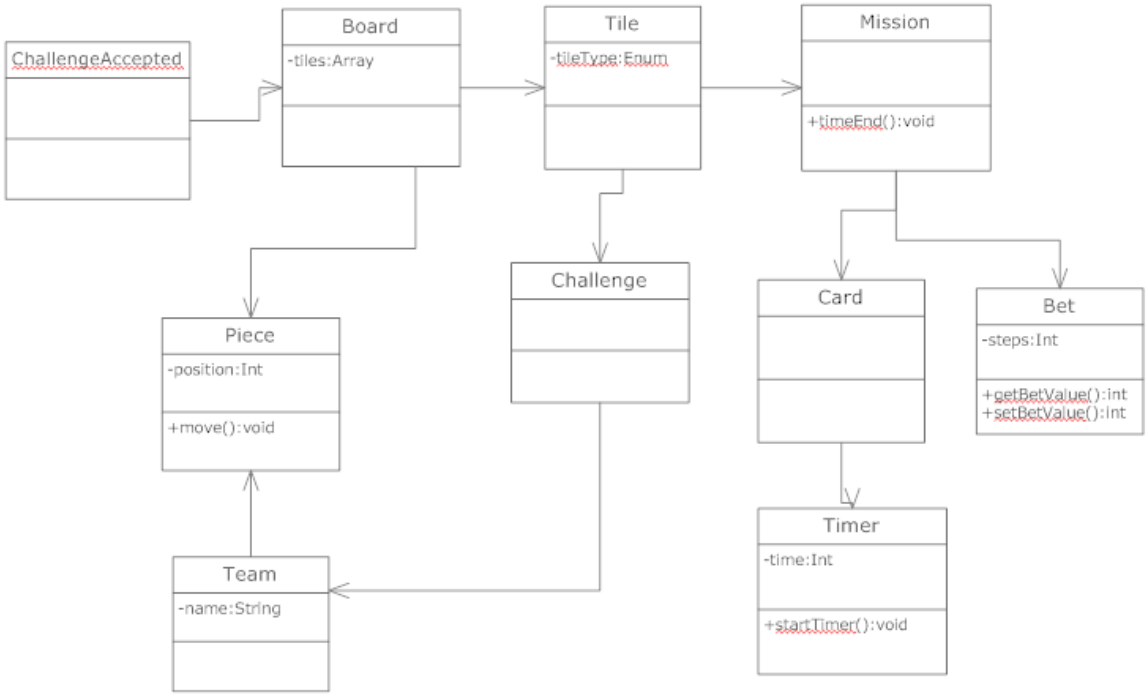


Challenge.

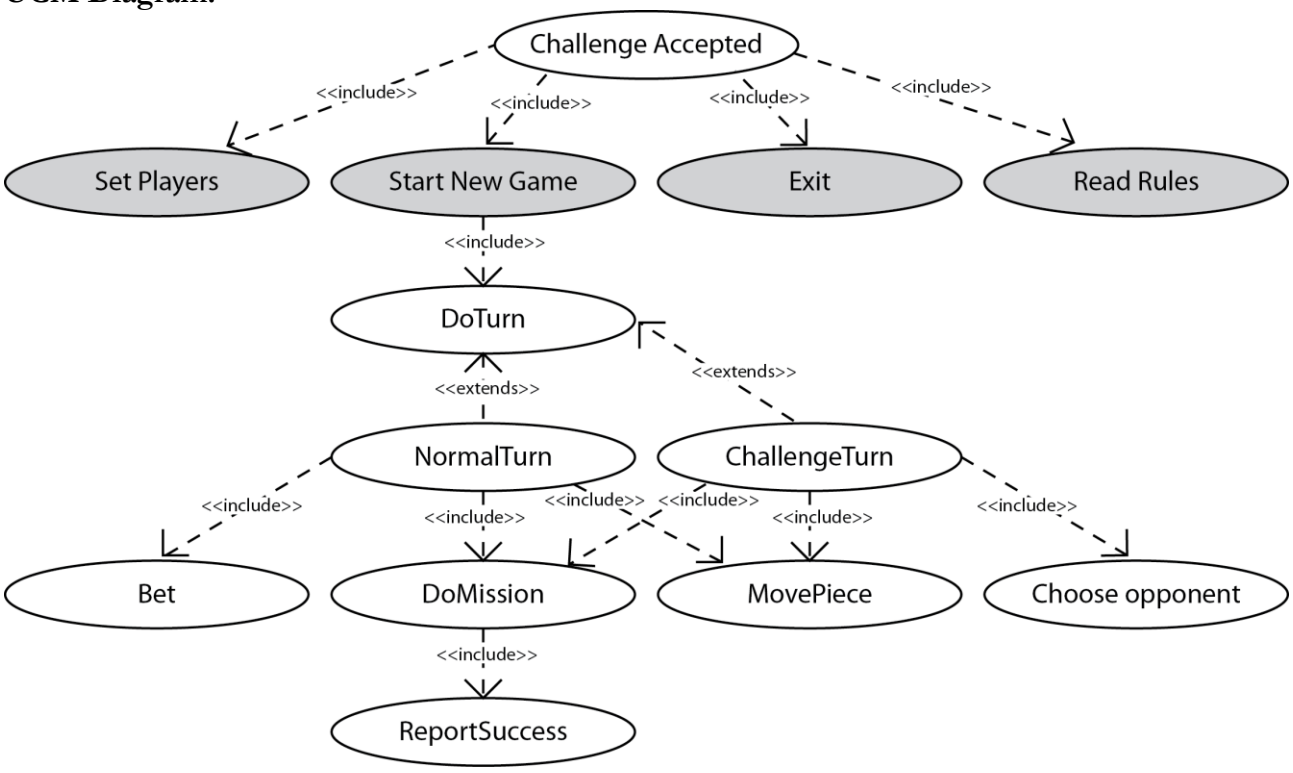
Game board when the first player reach the goal tile.



Domain model


ChallengeAccepted
Board
-tiles:Array
Tile
-tileType:Enum
Mission
+timeEnd():void
Piece
-position:Int
+move():void
Challenge
Card
Bet
-steps:Int
+getBetValue():int
+setBetValue():int
Timer
-time:Int
+startTimer():void
Team
-name:String

UCM Diagram.


Challenge Accepted
<<include>>
<<include>>
<<include>>
<<include>>
Set Players
Start New Game
Exit
Read Rules
<<include>>
DoTurn
<<extends>>
<<extends>>
NormalTurn
ChallengeTurn
<<include>>
<<include>>
<<include>>
<<include>>
<<include>>
Bet
DoMission
MovePiece
Choose opponent
<<include>>
ReportSuccess