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

– A digital version of 'Upp till Bevis'.

Contents

Version: 1.1
Date: 2012-07-30
Authors: Cecilia Edwall 920223-5769
Isabelle Frölich 900831-2846
Johan Gustavsson 871024-7134
Madeleine Appert 891110-4845

This version overrides all previous versions.

1. Introduction

This is an application that originally is a board game that we turned into a digital version. To play the game the teams makes a bet, thereafter they have to be able to do the amount of missions they betted to accomplish. If they manage to do the missions they can move forward on the board and hopefully reach the goal faster than the other teams. When the first playing team reaches goal, it's game over!

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

This is a multi-player application which is to be used on a computer platform. It will not require a connection to a network and will be used in social home environments.

The application will require at least 2 teams of 2 players, but will be able to manage 8 teams. Any amount of players can be a part of a team, it will not affect the application at all. The game is turn based, and will require the teams to approve each other's missions in order for the game to work. For completing a mission, the program will have a time limit of 30 seconds, but besides this, there will be no time constraints at all.

The game will end once a team has made it round the board, or if the game is cancelled manually.

1.3 Scope of application

System will allow 2-8 teams. There will be no computer-player.

Application will not keep track of high score etc. but it will show whose turn it is.

1.4 Objectives and success criteria of the project

The applications should function for a game which can be considered within the frame of what the game should manage.

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.

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

- Add different levels of difficulty.
- Make the board tiles placed randomly upon the board and not hard coded.
- Have different time limits for different teams, which might be in good in case of teams being of different ages, for example.
- Have clues.
- Possibility to save game.

1.5 Definitions, acronyms and abbreviations

- GUI, graphical user interface.
- Java, platform independent programming language.
- Turn, all teams will have their turns. A turn begins when a team are able to make a bet, and is over when the given assignment has either been approved or disapproved. The turn then passes to another team.

2. Requirements

2.1 Functional requirements

The players/teams should be able to:

1. Read the game rules.
2. Start a new game.
   a) Select how many teams will take part in the turn.
3. Do a turn. During a turn the team will be able to:
   a) Choose how many steps they bet they will manage.
   b) Change that bet before the actual mission starts.
   c) Achieve the assignment cards.
   d) Choose to see the next card within the time limit.
4. At the end of a team's turn the other playing team(s) will be able to approve/disapprove the mission accomplished by the team in turn.
3. Exit the application. Will end turn, round and game.

2.2 Non-functional requirements

2.2.1 Usability

Normal players are to understand how the game works without being forced to seek information elsewhere than in the application itself. The language will be English, hence the game requires all participants to speak English

2.2.2 Reliability

N/A

2.2.3 Performance

N/A

2.2.4 Supportability

N/A

2.2.5 Implementation

To achieve platform independence the application will use the Java environment.

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 or 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 UML diagram and textual descriptions.

2.3.2 Use case priority

1. New Game (high)
2. Start Mission (high
3. Bet (high)
4. End Game(high)
5. Show next card (mid)
6. Show rules (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 game (in Swedish).

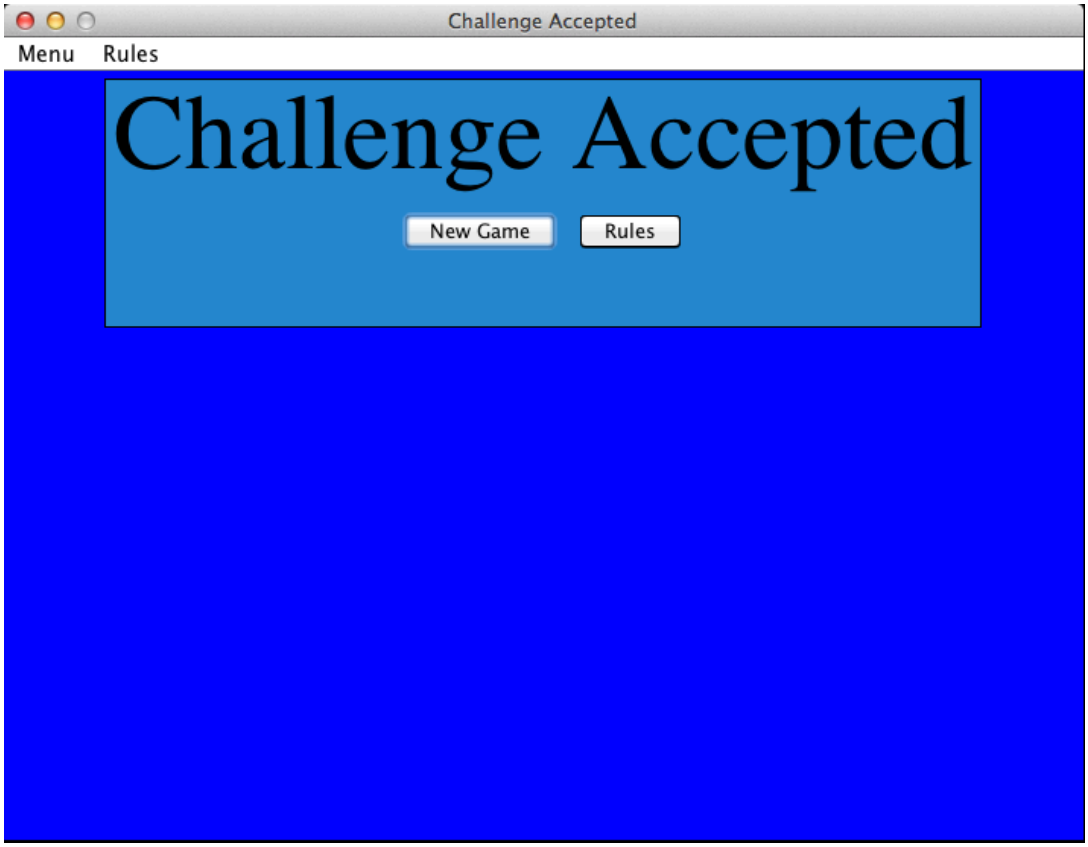
http://www.braspel.com/?id=317

Thorough explanation of game rules (in Swedish).

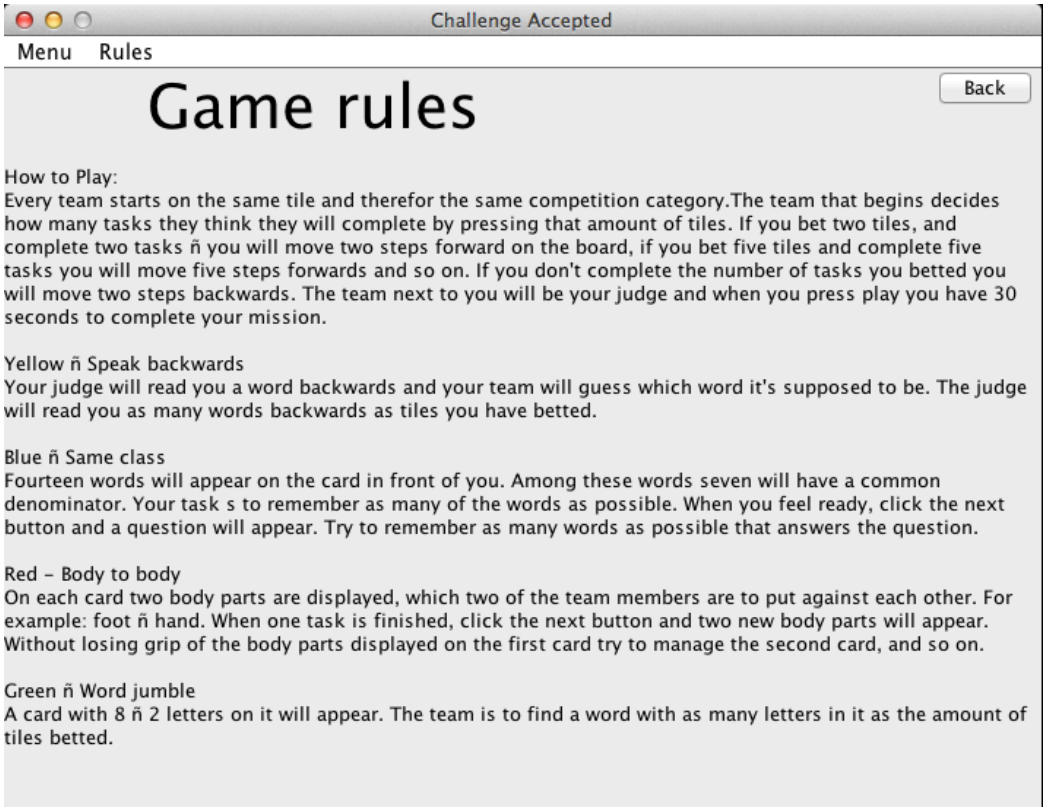
http://www.braspel.com/filearchive/1/1917/rules_UTB%20new.pdf

APPENDIX

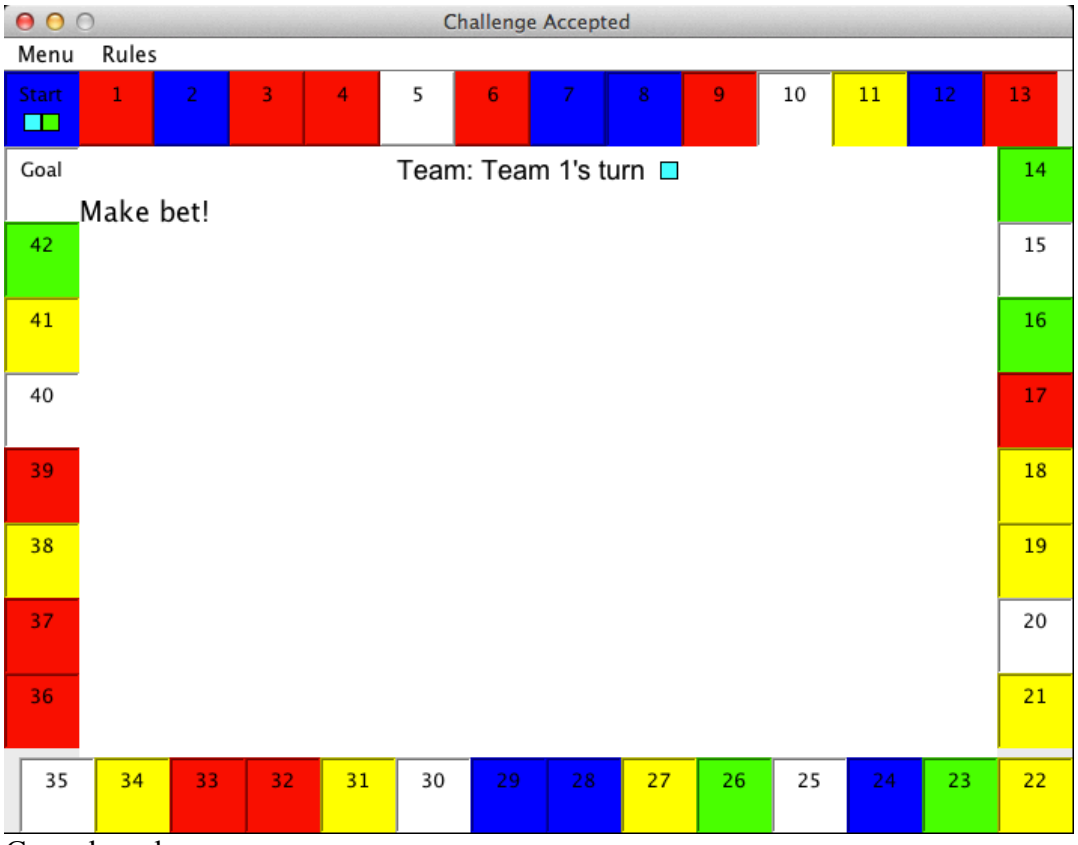
GUI



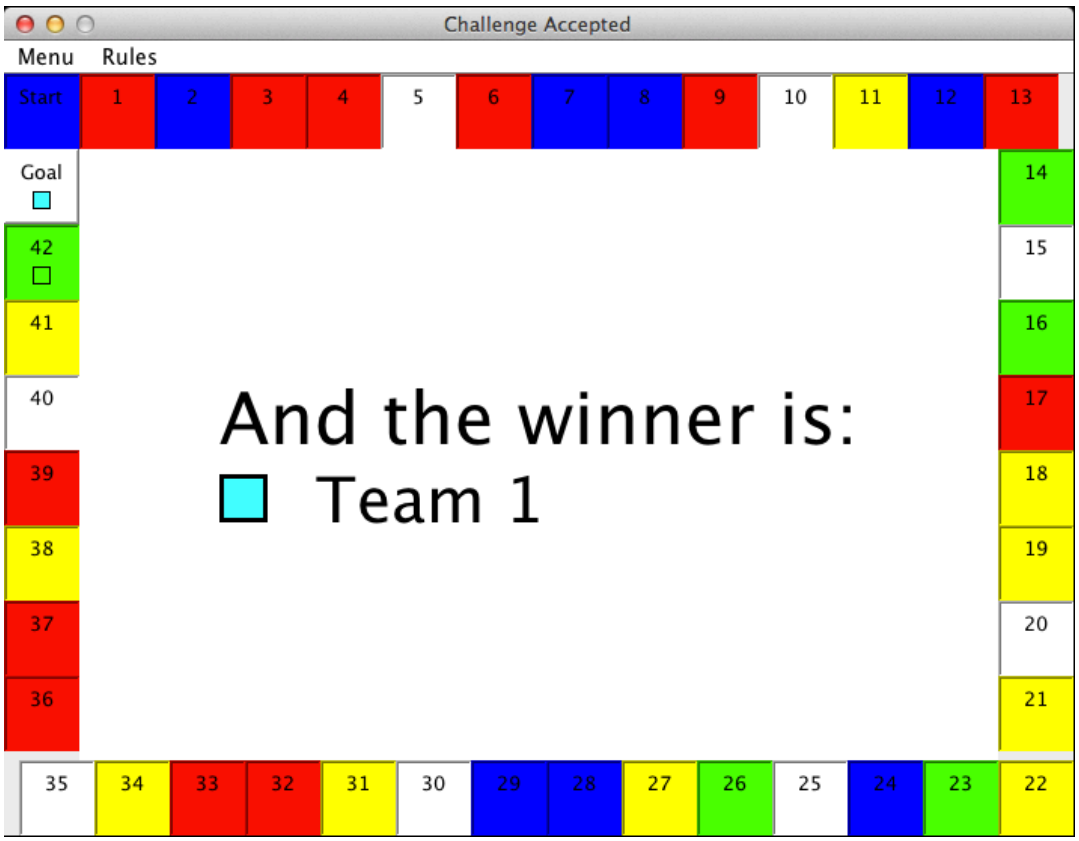
Start screen.



Screen for game rules.

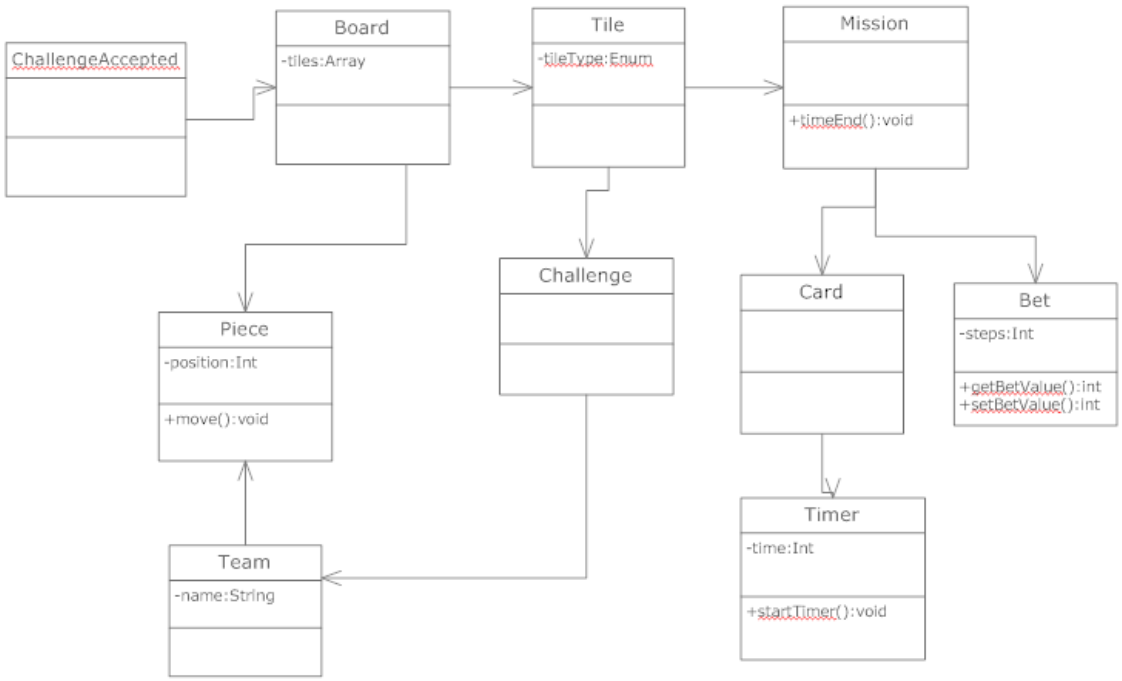


Game board.



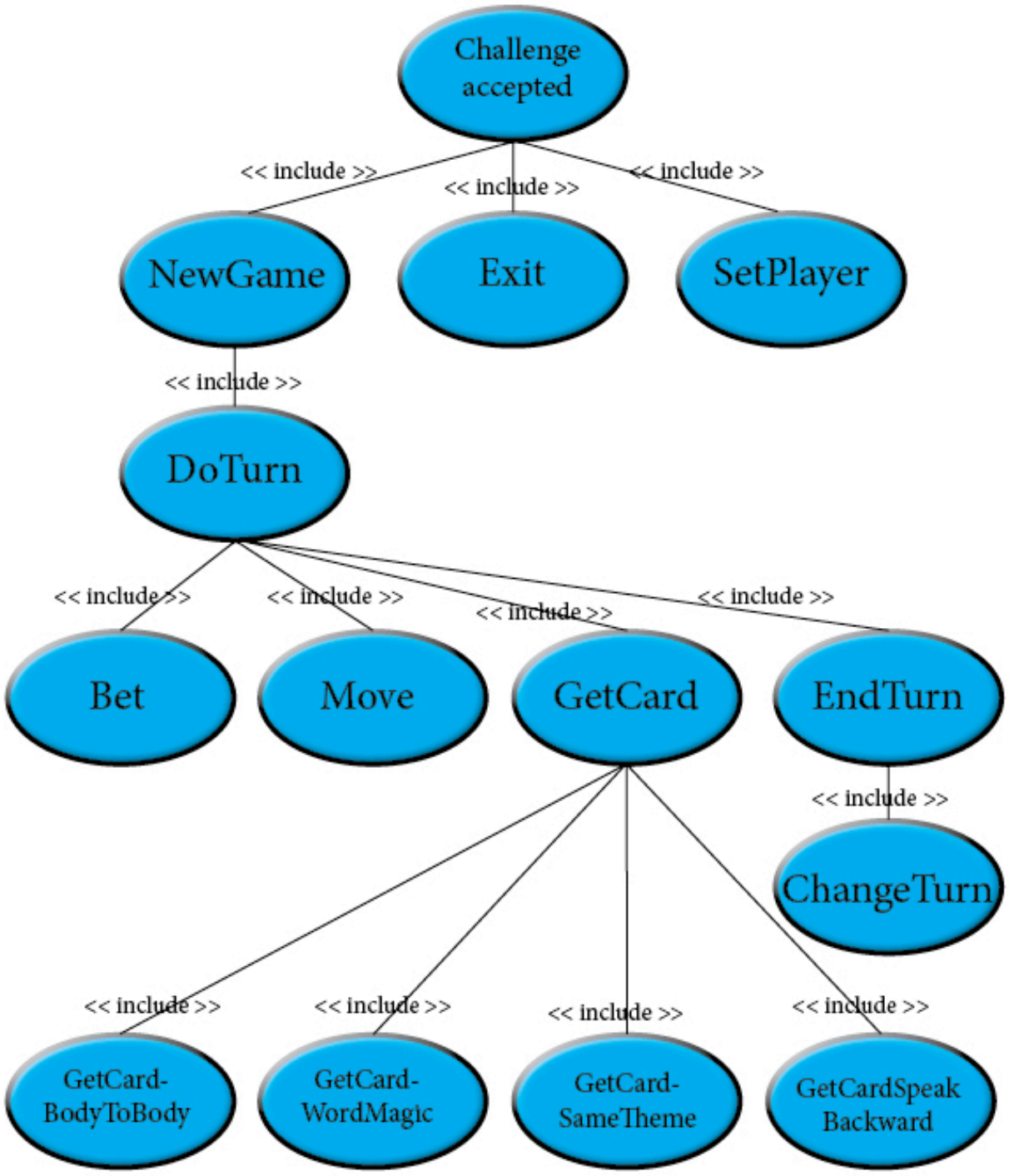
Game board when the first player reach the goal tile.

Domain model



Use Cases

Use Case Model



Use case: *Bet*

Short description: How a user makes a bet in the game.

Priority (high)

Extends or Includes (none)

Participating actors: ActivePiece which has a Team

Normal flow of events

Actor	System
ActivePiece clicks one of the betable tiles (one of the raised tiles) to make a bet.	
	The tile the ActivePiece clicks becomes lowered, and the bet is shown as a number in the higher left corner.
	The “Start Mission” button appears.

Alternate flow

3.1 Team clicks one of the menu items, for example the rules.

4.1 The rules will be displayed. See UC: Rules.

Exceptional flow

3.1 ActivePiece clicks one of the tiles that isn't betable.

4.1 Nothing happens and the ActivePiece will have to try again.

Use case: End*Game*

Short description: What happens when the game ends.

Priority (high)

Extends or Includes: Includes StartMission

Participating actors: ActivePiece

Normal flow of events

Actor	System
See UC: StartMission	
	If the “YES” button is clicked and the ActivePiece's bet amount is enough to take them to the goal tile the ActivePiece will be announced as the winner in the middle of the board.

Alternate flow

3.1 Team clicks the “Rules” button

4.1 The rules will be displayed. See UC: ShowRules.

Exceptional flow

3.1 If team types in a letter or a number that isn't between 2-8.

4.1 A dialog will appear and the Team will be able to type in the number of players again.

Use case: *New Game*

Short description: What happens when the user clicks New Game button.

Priority (high)

Extends or Includes (none)

Participating actors: Team

Normal flow of events

Actor	System
Team clicks “New Game” button	
	A dialog will appear where the team can type in how many teams that are participating.
Team types in a number between 2-8.	
Team clicks the “OK” button	
	The board appears and the number of teams that the team typed in are displayed as pieces on the board.

Alternate flow

3.1 Team clicks the “Rules” button

4.1 The rules will be displayed. See UC: Rules.

Exceptional flow

3.1 If team types in a letter or a number that isn't between 2-8.

4.1 A dialog will appear and the Team will be able to type in the number of players again.

Use case: *ShowNextCard*

Short description: How the player can see another card.

Priority (mid)

Extends or Includes: Includes StartMission

Participating actors: ActivePiece which has a Team

Normal flow of events

Actor	System
The ActivePiece clicks the “Next Card” button. See UC: StartMission	
	Depending on which category the ActivePiece stands on a card with different texts on it will appear. If the ActivePiece stands on the Backwards category the answer to the word will appear. If the ActivePiece stands on the SameCategory tile a question to be answered will appear. If the ActivePiece stands on the BodyToBody category the next body parts will appear. If the ActivePiece stands on the WordJumble category the answer to the word will appear.

Alternate flow

1 Team clicks one of the menu items, for example the rules.

1.2 The rules will be displayed. See UC: ShowRules

Use case: ShowRules

Short description: How a user sees the rules.

Priority (low)

Extends or Includes (none)

Participating actors: Any player that wants to see the rules

Normal flow of events

Actor	System
The player clicks on the “Rules” button in the start-view or on the Menu Item “Rules”	
	The Rules pop up on the screen, with a “Back” button (And a “Continue game” button if you are in the middle of a game).
The player clicks the “Back” button (or the “Continue game” button).	
	The start-view is shown again (Or if the “Continue Game” button was clicked the player will be taken back to the game).

Alternate flow

None

Exceptional flow

None

Use case: *StartMission*

Short description: How a mission is executed.

Priority (high)

Extends or Includes: Extends Bet

Participating actors: ActivePiece which has a Team

Normal flow of events

Actor	System
The ActivePiece clicks the “Start Mission” button. See UC: Bet	
	Depending on which category the ActivePiece stands on a card in that category will appear along with a “Done” button and a “Next Card” button.
	The countdown will begin and the ActivePiece will have 30 seconds to finish their mission.
The ActivePiece will click “Done” or the time will run out. If the ActivePiece clicks the “NextCard” button see UC: ShowNextCard.	
	The question “Was the mission completed successfully?” will appear along with the buttons “NO” and “YES”
The ActivePiece will have to click either the “NO” or the “YES” button for the mission to end	