Requirements Bejeweled

by

Irene van der Blij, Mayke Kloppenburg, Samuel Sital,
Kiran Kaur, Takang Kajikaw Etta Tabe

TI2206 Software Engineering Methods

Content

1. Functional requirements

For the game Bejeweled, the requirements regarding functionality and service are grouped under the functional requirements. Here the MoSCoW model is used to further work out these requirements, considering the importance of the requirements.

1.1 Must haves

- The entire grid shall be filled with jewels; each cell has to contain one jewel.
- The board shall consist of a grid of 8 by 8 cells.
- The player shall be able to switch two jewels only if it results a valid combination. A valid combination can be:
  - Three jewels of the same color in a horizontal or vertical row.
  - Four jewels of the same color in a horizontal or vertical row.
  - Five jewels of the same color in a horizontal or vertical row.
  - A combination of 5 jewels in an L shape.
  - A combination of 5 jewels in a T shape.
- If the player makes a valid combination, the jewels in the combination vanish from the board.
- When there is an empty cell, the jewel above this cell shall fall down to fill the empty cell.
- When there is an empty cell on the topmost row, new random jewels shall appear here.
- The game shall end when there is no possible move left.
- The game shall show the player's current level number.
- The game shall show the player's score.

1.2 Should haves

- The game shall have multiple levels, and the player shall start at level number 1 when the game starts.
- The game shall reset the player's score and the level number when a game ends.
- A player shall be able to stop a Bejeweled game in progress.
- A player shall be able to start a new Bejeweled game.
- If a combination of 4 or more jewels is made, a special jewel shall appear in one of the cells where the combination was made. The following special jewels shall appear:
  - When a valid combination of four jewels in a line is made, a 'Flame gem' shall appear. This gem shall explode when matched, destroying itself as well as all eight surrounding jewels.
  - When a valid combination of five jewels in a line is made, a 'Hypercube' shall appear. This gem can be matched to any adjacent jewel, regardless of the color, in order to destroy all jewels of that color on the board.
  - When a valid T or L shape combination is made, a 'Star gem' shall appear. This gem shall send explosive bolts of energy right, left, up and down, destroying all jewels on those lines.
- The game shall keep track of the player´s score using the following scoring system:
  - For making a combination of three jewels the player earns 50 points.
  - For making a combination of four jewels the player earns 150 points.

- For making a L or T shaped combination the player earns 150 points.
- For making a combination of five jewels the player earns 500 points.
- For making a combination with a ‘Flame gem’ the player earns 20 points for detonating the special stone, and 20 points per jewel destroyed.
- For making a combination with a ‘Hypercube’ the player earns 50 points for detonating the special stone, and 50 points per jewel destroyed.
- For making a combination with a ‘Star gem’ the player earns 50 points for detonating the special stone, and 50 points per jewel destroyed.

1.3 Could haves

- The game shall have the possibility to give a hint to the player when he asks for it.
- The game shall show the player his final score and level number when the game has ended.
- The game shall show an encouraging message (such as ‘Good Job!’) when more than four combinations are made in one turn.
- The game shall award the player with bonus points when several combinations are made in one turn.
- The game shall indicate the jewel which was lasted clicked on by the player.

1.4 Won’t haves

- The game shall play a music theme when in progress.
- The player shall be able to customize the layout of the game.
- The game shall save the scores of all its players using a high-score list with the top ten of names and scores of players with the highest scores.
- The player shall be able to select another game mode.

2. Non functional requirements

In this chapter requirements besides functionality and services will be discussed. It will also include the design constraints. These requirements do not indicate what the system should do, but instead indicate the constraints that apply to the system or the development process of the system.

- The game shall be playable on Windows (7 or higher), Mac OS X (10.8 and higher).
- The game shall be implemented in Java.
- A first fully working version of the game shall be delivered at September 11, 2015.
- For the iterations after the delivery of the first fully working version, the Scrum methodology shall be applied.
- The implementation of the game shall have at least 75% of meaningful line test coverage (where meaningful means that the tests actually test the functionalities of the game and for example do not just execute the methods involved).