

# *Requirements*

## *JavaFX implementation Bejeweled*

*by*

*Group 33*

***Teaching Assistant***

*B. Reijm*

***Group members***

*Wytze Elhorst*

*Thomas Kolenbrander*

*Steven Meijer*

*Bart van Oort*

*Steven Meijer*

# Contents

|                                    |          |
|------------------------------------|----------|
| <b>Functional requirements</b>     | <b>3</b> |
| Must Haves . . . . .               | 3        |
| Should Haves . . . . .             | 3        |
| Could Haves . . . . .              | 3        |
| Would/Won't Haves. . . . .         | 3        |
| <b>Non-functional requirements</b> | <b>4</b> |

# Functional Requirements

The requirements for the JavaFX implementation of Bejeweled are divided into functional and nonfunctional requirements. The functional requirements are divided into four categories using the MoSCoW model.

## Must Haves

- When a new game is started the graphical user interface displays the board.
- When a user selects a gem it should be clear which gem has been selected.
- When a user makes a move the graphical user interface should display the updated board.
- When a user makes a move the graphical user interface should display the updated score.

## Should Haves

- The graphical user interface should include a button to restart the game.
- When there are no moves left the graphical user interface should display 'Game over!'.
- When a move can not be made the player has to be informed of this.
- There should be a button which will save the game.
- There should be a button which will load a previous game.

## Could Haves

- When a new game is started there is an animation of the gems appearing on the screen.
- After a successful move, there will be an animation of the gems falling down.

## Would/Won't Haves

- The game can be displayed in full-screen without any borders.

# Nonfunctional Requirements

The following requirements do not describe what the system should do, but they describe constraints that apply to the system or the development process of the system.

- The graphical user interface should make use of JavaFX.
- The graphical user interface should be visually appealing.