Requirements

Functional requirements

1.1 Must Haves

1. At the start of the game, a board is created and filled with gems. No sets of 3 or more gems shall be horizontally or vertically aligned.
2. The game's board shall consist of an 8x8 grid.
3. Each gem shall take up a 1x1 unit of space.
4. Sets of three aligned gems (vertically or horizontally) or more will disappear from the playing field.
5. When gems are cleared horizontally, all the gems above the ones that were cleared will go down one row.
6. When X gems are cleared vertically, the ones above will go down X rows.
7. Whenever a particular row or column is cleared, every gem above that row or column will move down until the bottom row of gems land on the last row of the grid on the board. On the empty spaces that appear on the board, there will be new random gems.
8. Whenever a gem is clicked, and none are currently selected, the clicked gem will be selected.
9. Whenever a gem is clicked which is not beside a selected gem, the clicked gem will be selected and the previous one will be unselected.
10. Whenever a gem is clicked that is located horizontally or vertically beside the selected gem, the gems will be switched and the gems are unselected only if the switch will create an aligned set of 3 gems or more.
11. Whenever two gems are switched, and no match of 3 or more gems of the same colour or more is made, the switch will be undone.
12. Whenever two gems are switched, and a match of 3 or more gems of the same colour is made, the matched gems will be cleared.
13. If there are no more possible moves, the player loses the game.

1.2 Should Haves

1. The player should have the option to start a new game of Bejeweled after a game over.
2. When a new game of Bejeweled is started, the player's score should be initialized at 0.
3. The player should be able to stop a game of Bejeweled that is currently in progress.
4. A game of Bejeweled should end once the player stops it or loses the game.
5. When the player loses, a message with the words 'Game Over' appears on the screen.

6. When gems are cleared, the player's score increases.
7. The game keeps track of the player's score, using the following score values for certain combo's and blocks:

    -A 3-gem match will award 50 points times the level number
    -A 4-gem match will award 100 points times the level number
    -A 5-gem match will award 500 points times the level number
    -When you make 2 simultaneous matches, an additional 50 points times the level number is awarded
    -If a match is created, and the gems fall in such a way that another match is made, an additional 50 points per level is awarded. Then if another match happens, this will increase to 100. For the next one it will be 150 etcetera.

1.3 Could Haves

1. The game plays music in the background.
2. Highscore counter / local leaderboard.
3. The game should have multiple levels. A new game starts at level 1 and whenever clearing a particular level, the level increases by 1.
4. Tutorial screen.
5. Main menu screen.
6. Timers on later levels to increase difficulty.
7. Clearing multiple sets in the same move will reward more points.
8. Clearing a set of 4 gems creates a power gem.
9. Clearing a set of 5 cubes will create a hyper cube.
10. Clearing a power gem also clears all 8 gems surrounding it.
11. When a hyper cube gem is cleared all gems of the same color will be cleared from the screen.

1.4 Won't Haves

1. Multiple versions of Bejeweled (only Classic).
2. The player will be able to change the background of the game board.
3. Multiplayer mode, there is only single player mode.
4. No online functionality.
5. Global leaderboard. There will be a local leaderboard.

Non-functional requirements

1. A first fully functional version of the game will be delivered on 11-9-2015

2. The game will be playable on Windows (7 or higher), Mac OS X (10.10 and higher), and Linux(distributions Arch Linux and Ubuntu).
3. IDE that will be used during the development process is Eclipse(Luna and Mars).
4. Git version control system will be used for development along with Github as repository hosting service.
5. The game will be implemented in Java 8.
6. For the iterations after the delivery of the first fully working version, the Scrum methodology shall be applied.
7. 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)

Sources:
https://en.wikipedia.org/wiki/Bejeweled, http://www.popcap.com/games/bejeweled2/online