



Bubble Spinner Assignment 1

SEM-GROUP 13
2019-20

Ali Kahawati
Rosalie van de Vuurst
Toma Zamfirescu
Marije Tromp
Jordy van der Tang

Exercise 1: Requirement Engineering

FUNCTIONAL REQUIREMENTS:

■ Must have:

- The game shall have an authentication system for the player.
- The player shall have his/her own unique username and password to login to the game.
- The player shall be able to login by entering his/her credentials on a log-in screen.
- The game shall have a cannon that shoots bubbles.
- The game shall have a hexagonal grid on which the bubbles are located.
- The game shall only have bubbles of the same size.
- The game shall randomly generate the color of a bubble, from a set of predetermined colors.
- The game shall have a bubble at the center that cannot be removed by playing, it will have a special color (i.e. gray).
- The game shall have a random selection of bubbles at the middle of the grid around the center bubble when it starts.
- The player shall be able to change the direction of the cannon by moving the mouse.
- The player shall be able to shoot a bubble from the cannon by pressing the left mouse button.
- The game shall reflect bubbles when they hit walls.
- The game shall make the shot bubble stick to the first other bubble it hits.
- The game shall place sticking balls in the nearest available grid position from the collision point.
- The game shall remove bubbles from the screen after a collision, when they have the same color as the shot bubble and their connected sum is ≥ 3 .
- The game shall remove all bubbles that have become disconnected from the center.
- The game shall have a start-up screen.
- The player shall win the round, when only the center bubble is left.
- The game shall be over when a bubble on the grid touches a wall.
- The player shall have the option to start a new match at the end of the previous.

■ Should have:

- The game shall apply physics when collisions happen i.e. the grid will rotate.
- The player shall have a profile page, where more information could be found.
- The start-up screen shall have at least three options i.e. start a game, see scores, and tutorial.
- The game shall show the color of the next two bubbles.
- The game shall have multiple difficulties level.
- The game shall have a score system.
- The game shall have a scoreboard where the score of each play will be saved next to his/her nickname.
- The player shall be able to have a nickname to show on the scoreboard.
- The game shall show the top 5 scores ever made at the end of each round.
- The game shall save the player's score when the player has finished the round

■ Could have:

- The game shall set a sound effect when a bubble is shot.
- The game shall have a tutorial.
- The game shall have a way to choose your own color scheme.
- When no balls have disappeared for a certain amount of time some new balls appear on the grid.
- The game shall have a pointer in the direction of the cannon.
- The game shall have an effect when the balls disappear. (i.e. popping)
- The game shall start a new round when the previous round is won, and the score keeps counting.
- When a certain color ball is no longer in the game, it won't reappear until the next game.
- The game shall start a bonus level where points earned are doubled when there are no more bubbles on the grid.
- The game shall have visual effects when a collision happens.
- The player shall have different balls to unlock. (i.e. different colors, patterns)
- The player shall be able to press an undo button, which reverses the last shot.
- The player shall be able to add friends.
- The game shall have an endless mode. (TBD)
- The game shall have a pause button.
- The game shall have a grid that spins at a constant speed.

■ Won't have:

- The game shall have extra elements in the playing field (i.e. blocks) that the ball can bounce off.
- The player shall be able to play against friends.
- The game shall have tools to help when a level is difficult i.e. exploding bubbles.
- The game shall have bubbles with different functionalities.

NonFUNCTIONAL REQUIREMENTS:

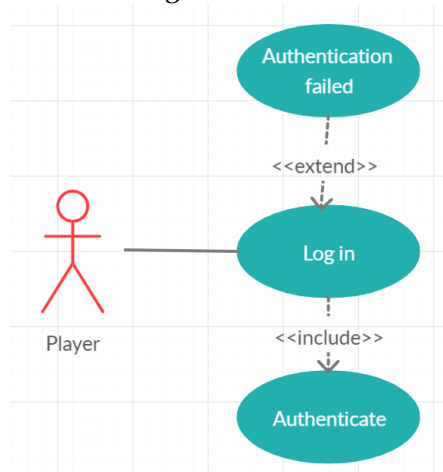
- Use SQL and JDBC driver.
- Use prepared statements in Java to avoid code-injection vulnerabilities.
- The game shall be build using Java programing language.
- The code of the game shall have a test coverage of at least 75%, but not for UI elements.
- The code shall have enough documentation for all public classes and methods.
- The documentation shall be clear and explanatory.

Exercise 2: Modelling Use Cases

Requirement: The player shall be able to login by entering his/her credentials on a log-in screen.

Natural language description: When the player opens the application, they will see a log-in screen. If the player fills in the correct username and password, they will be logged in and see the start-up screen. If the player fills in a wrong username and/or password, they will be shown a validation failure message.

Use case diagram:



Use Case: Log In

Purpose: Log in to the bubble spinner game.

Overview: The Player activates the application. The system requests username and password. The system validates username and password. If the validation is successful, the system displays start-up screen; otherwise (unsuccessful validation), the system displays validation failure message.

Cross References: Requirement R₃

Actors: Player

Pre Conditions:

- The application must be active.
- The application must be connected to the database.

Post Conditions:

- Start-up screen is shown.

The normal flow of events:

Player Actions	System Actions
1.Player activates the application	2.System asks for username and password
3.Player gives username and password	4.System validates username and password
	5.System displays start-up screen

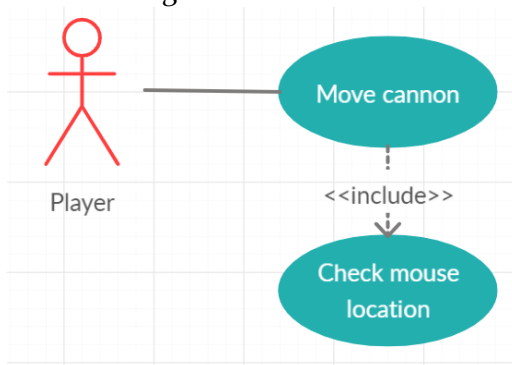
The alternative flow of events:

Step 4: The validation failed, a validation failure message is shown.

Requirement: The player shall be able to change the direction of the cannon by moving the mouse.

Natural language description: The cannon always points in the direction of the mouse, so when the player moves the mouse, the cannon moves with it.

Use case diagram:



Use Case: Move cannon

Purpose: Move the cannon by moving the mouse.

Overview: The system displays the cannon pointing in the direction of the mouse. The player moves the mouse. The system moves the cannon, so it points in the direction of the new position of the mouse.

Cross References: Requirement R10

Actors: Player

Pre Conditions:

- The application must be active.
- The application must be connected to the database.
- A game must be active.
- The mouse pointer must be on the playing field.

Post Conditions:

- The cannon points in the direction of the mouse.

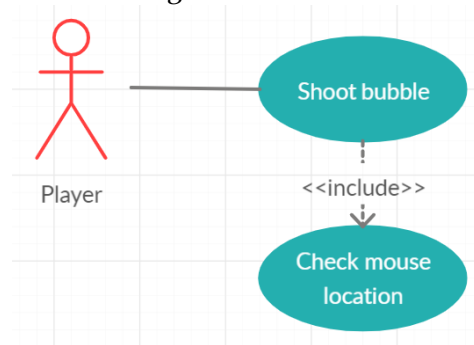
The normal flow of events:

Player Actions	System Actions
1. Player moves mouse	2. System moves cannon

Requirement: The player shall be able to shoot a bubble from the cannon by pressing the left mouse button.

Natural language description: When the player clicks the left mouse button the bubble is shot from the cannon.

Use case diagram:



Use Case: Shoot bubble

Purpose: Shoot a bubble towards the grid.

Overview: The player clicks on the left mouse button. The system saves the location of the mouse pointer. The system moves the bubble in the cannon in the direction of the saved location.

Cross References: Requirement R11

Actors: Player

Pre Conditions:

- The application must be active.
- The application must be connected to the database.
- A game must be active.
- The mouse pointer must be on the playing field.
- A bubble must be in the cannon.

Post Conditions:

- The bubble is moving in the direction of the mouse pointer.

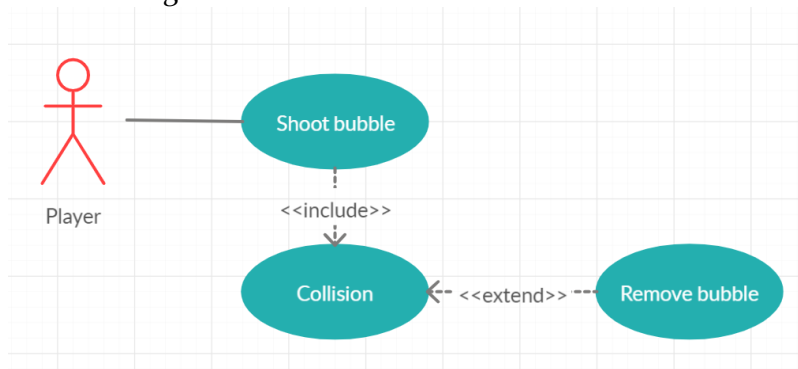
The normal flow of events:

Player Actions	System Actions
1. Player clicks left mouse button	2.System saves location mouse
	3.System moves bubble

Requirement: The game shall remove bubbles from the screen after a collision, when they have the same color as the shot bubble and their connected sum is ≥ 3 .

Natural language description: When a shot bubble collides with 2 or more bubbles of the same color, they will disappear.

Use case diagram:



Use Case: Check collision

Purpose: After a collision remove bubbles with the same color connected to the shot bubble if connected sum is ≥ 3 .

Overview: The player shoots a bubble. The system moves the bubble. The system checks the collision. If collision is true, the system removes bubbles; otherwise, the game continues.

Cross References: Requirement R15

Actors: Player

Pre Conditions:

- The application must be active.
- The application must be connected to the database.
- A game must be active.

Post Conditions:

- The game continues with less bubbles on the screen.

The normal flow of events:

Player Actions	System Actions
1. Player shoots bubble	2. System moves bubble
	3. System checks collision
	4. System removes bubbles

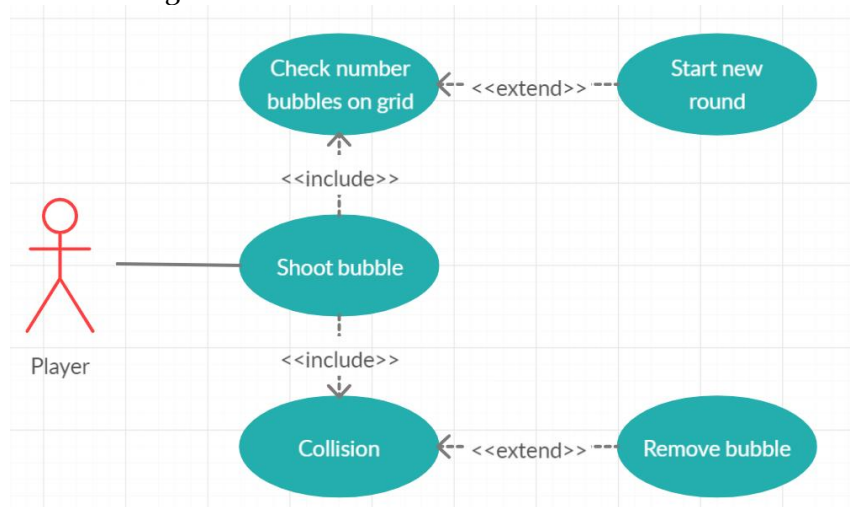
The alternative flow of events:

Step 3: The collision is false, the game continues.

Requirement: The player shall win the round, when only the center bubble is left.

Natural language description: When all the bubbles except the center bubble are gone, the player wins the round.

Use case diagram:



Use Case: Win round

Purpose: Recognize when the round is won.

Overview: The player shoots a bubble. The system moves bubble. The system checks collision. The system checks the number of bubbles left on the grid. If only the center bubble is left, the round is won and the system starts the next round; otherwise, the round continues.

Cross References: Requirement R18

Actors: Player

Pre Conditions:

- The application must be active.
- The application must be connected to the database.
- A game must be active.

Post Conditions:

- A new round is started.

The normal flow of events:

Player Actions	System Actions
1. Player shoots bubble	2.System moves bubble
	3.System checks collision
	4.System checks number of bubbles
	5.System starts new round

The alternative flow of events:

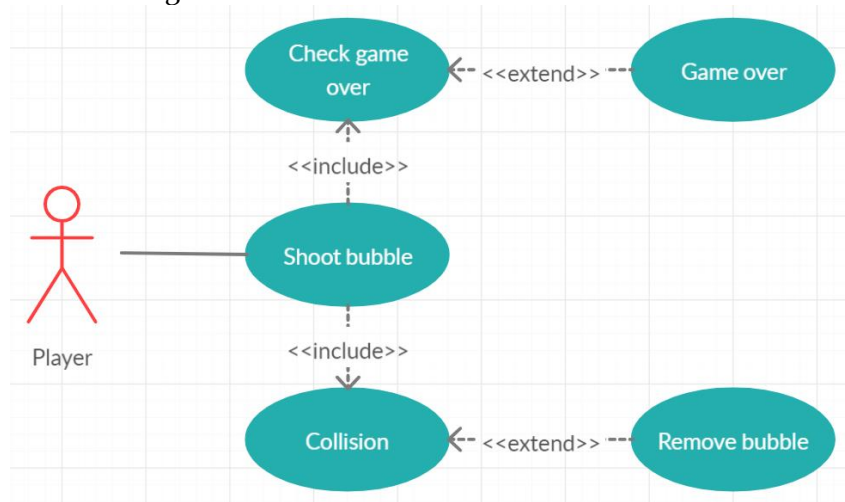
Step 3: The collision is false, the round continues.

Step 4: There are bubbles left on the grid, the round continues.

Requirement: The game shall be over when a bubble on the grid touches a wall.

Natural language description: When a bubble on the grid touches the wall, the game is over.

Use case diagram:



Use Case: Game over

Purpose: End the game

Overview: The player shoots a bubble. The system moves the bubble. The system checks collision. The system checks game over. If game over is true, the system displays game-over screen; otherwise, the game continues.

Cross References: Requirement R19

Actors: Player

Pre Conditions:

- The application must be active.
- The application must be connected to the database.
- A game must be active.

Post Conditions:

- Game-over screen must be shown.

The normal flow of events:

Player Actions	System Actions
1. Player shoots bubble	2.System moves bubble
	3.System checks collision
	4.System checks game over
	5.System displays game-over screen

The alternative flow of events:

Step 3: Collision removes bubbles that would cause game over, the game continues.

Step 4: Game over is false, the game continues.