Requirements Bubble Spinner Group 15

Functional requirements

Must-Haves:

1. The game shall show a login page when the game starts.
2. The game shall create the bubble structure at the center of the board the player starts a new game.
3. The game shall create a shooter object at the top of the page when the game starts.
4. The player should be able to play up to at least 3 consecutives levels.
5. The game shall reward the player achievements based on his statistics.
6. The player should be able to shoot a bubble with a range of 180 degrees from the shooter position.
7. The game shall calculate the direction and reflection of the bubble following proper physics laws.
8. The game shall not allow the bubbles to bounce outside of the screen.
9. The game shall not allow the current bubble to overlap with other bubbles.
10. Bubbles that are shot against the side of the screen are reflected back inwards.
11. Every bubble will be assigned a color randomly.
12. The game shall destroy the bubbles that are touched by the shot bubble if they are of the same color and there’s 3 bubbles or more attached of the same color.
13. The game shall destroy floating bubbles that have no attachment to the central structure.
14. The player shall be able to select different difficulties of the game.
15. The game shall randomize the way it draws the bubbles on the screen.
16. The game shall create more difficult bubble structures based on the difficulty selected.
17. The game shall store the statistics of the games played and its users.
18. The game shall show the best scores of all the players in
19. The game shall show a splash screen where the player is allowed to choose what to play.
20. When you miss more than a set amount of balls, new bubbles get added to the structure at random.
21. The game shall show a leaderboard containing the top players and scores.
22. The game should be able to detect collisions between bubbles.
23. The player should be able to end the game.

Should-Haves:

1. The player shall be able to receive and show badges if certain achievements are met.
2. The game shall offer the possibility to switch between dark and light mode.
3. The game shall allow the player to play it for an infinite amount of time.
4. The game shall introduce special bubbles with their own set of properties.
5. The game shall offer a walk-through at the start of the game for new players.

6. The game shall allow the player to register with username, email and password.
7. The game shall draw a line where the player is aiming.
8. The center of the structure should be indestructible.
9. The level should end when there are only bubbles of one color remaining in the structure.

Could-Haves:

1. The game shall allow the player to play as a guest without the need for a password.
2. The game shall have a timer for each game such that an idle user will lose the game or if the difficulty is set to the maximum the user will need to complete it as fast as possible.
3. The player shall be able to add other players to his friend list.
4. The game shall allow the user to recover his own password if lost.
5. The game shall have a multiplayer version where two users can play against each other where the first one to complete the game or a series of levels win.
6. The game shall allow to choose between different modes the game can be played, competitive or casual.
7. The player shall be able to play in multiplayer against a computer bot that simulates the behavior of a real player.
8. The game shall show a loading screen for the transitions.

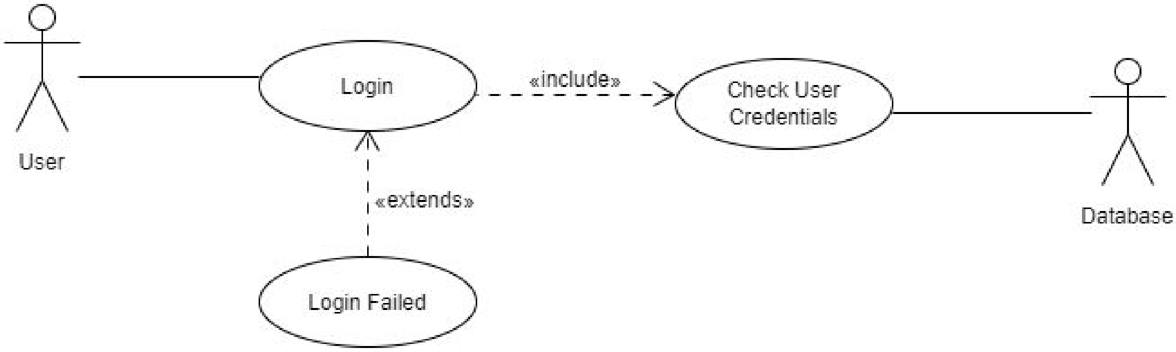
Won’t-Haves:

1. The game shall have a theme song that plays when it’s started.
2. The game shall show advertisement between different sessions for monetization purposes.
3. The game shall show pop-ups asking for reviews.
4. The game should allow to play it offline.

Non-Functional Requirements

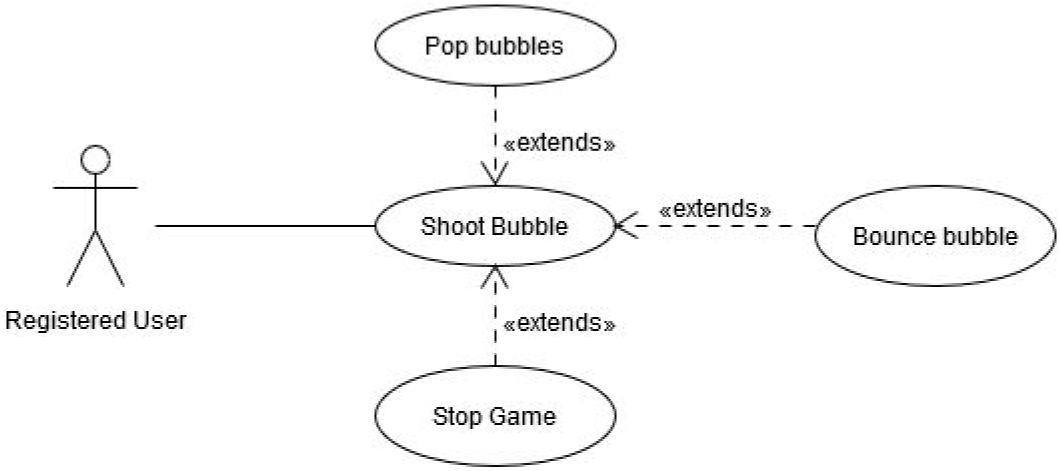
1. The game shall be implemented in Java.
2. The game should be runnable on Windows, Mac OS X and Linux.
3. The game shall have at least 75% branch coverage.
4. Each SQL query shall be executed using Prepared Statement.
5. The game shall save the user information in a SQL database.

2- Modelling Use Cases



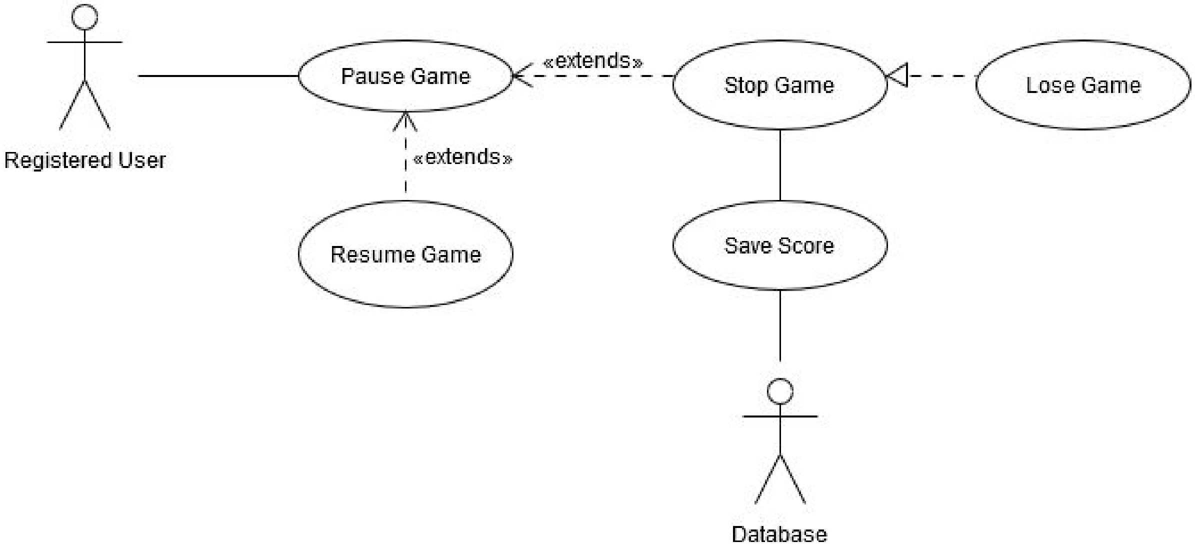
U1

Use Case: Login User
Author: Victor van Wieringen
Date: 25/11/2019
Purpose: Have the user login using email/username and password
Overview: The user enters their credentials in the login form. The application hashes the password and validates the combination using the database. If the credentials match an entry in the database, the user is logged on to their account, and the application closes the login screen. If one or more credentials are missing or if there is no match in the database, the application displays an error message, and the user can try again.
Cross-reference: Requirements R1, R3



U2

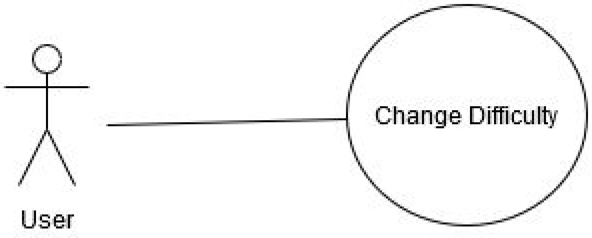
Use Case: Shoot Bubble
Author: Victor van Wieringen
Date: 25/11/2019
Purpose: When playing the game, the user is able to shoot bubbles at the central structure starting from the top. When a shot bubble hits the structure, it snaps to a discrete position closest to where it hit the structure and the structure rotates depending on the torque the impact generates. If the bubble gets placed in a position on the structure where it is connected with two or more bubbles of the same colour, the bubbles disappear, and the user is awarded points. When a bubble hits the side of the screen, it is reflected and continues traveling on.
Cross-reference: Requirements R2, R3, R6, R7, R8, R10, R12, R22



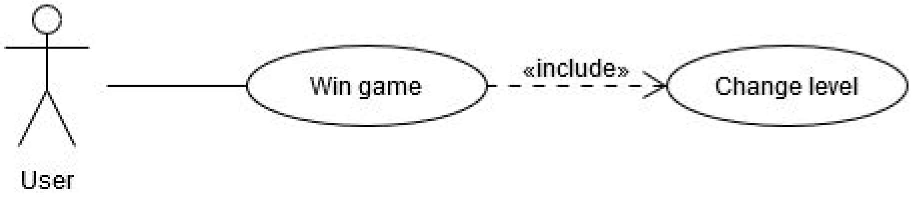
U3

Use Case: Pause Game
Author: Victor van Wieringen
Date: 25/11/2019
Purpose: When playing the game, the user is able to pause the game by pressing the escape key.
When the game is paused, the in-game timer is paused, all input to the game screen is disabled and a small menu is drawn over the game screen. From within this menu the user is able to resume and exit the game. The way exiting the game works is described in use case U4
Cross-reference: Requirements R23

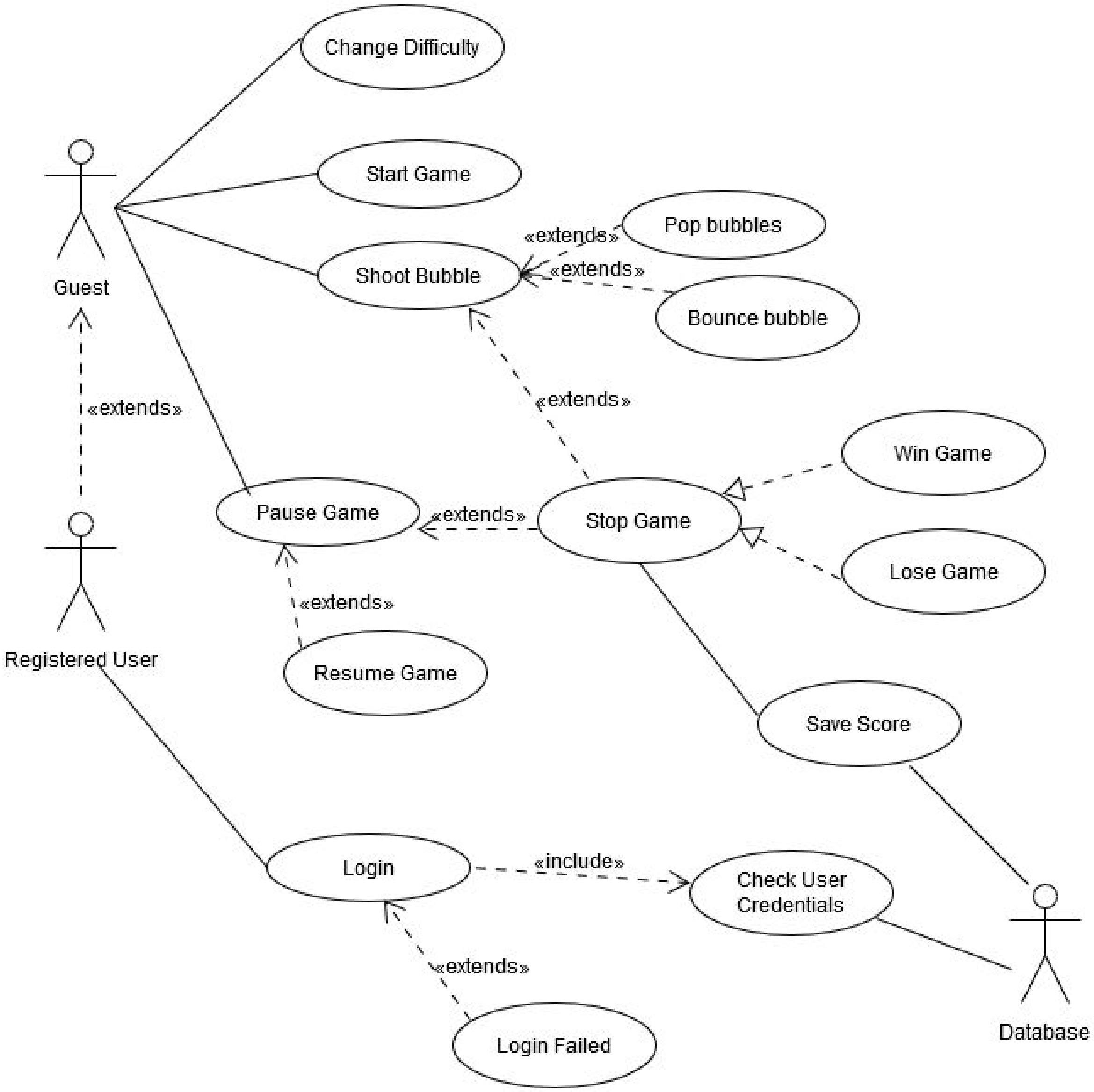
U4
Use Case: Exit Game
Author: Victor van Wieringen
Date: 25/11/2019
Purpose: When the user exits the game, the game stores the current score of the game in progress and submits it to the database. The game screen is closed and the user is taken back to the splashscreen.
Cross-reference: Requirements R23



U5
Use Case: Change difficulty
Author: Victor van Wieringen
Date: 25/11/2019
Purpose: The user is able to set the difficulty from within the splashscreen, choosing from a couple of predefined options. The next game that is started gets started with the difficulty set to this value.



U6
Use Case: Change levels
Author: Victor van Wieringen
Date: 29/11/2019
Purpose: The user can win the game by clearing all of the bubbles. After this the game will create a new level with an increased amount of difficulty.
Cross-Reference: R4


Change Difficulty
Start Game
Shoot Bubble
Pop bubbles
Bounce bubble
«extends»
«extends»
Guest
«extends»
«extends»
Win Game
Pause Game
«extends»
Stop Game
Lose Game
Registered User
«extends»
Resume Game
Save Score
Login
«include»
Check User Credentials
«extends»
Login Failed
Database