Use Case UC1: Access Help Screen

Scope: The Game

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to reach help information

Preconditions: Running Application

Postconditions: User welcomed with a help screen

Main Success Scenario:

1. In the login screen, the user wants to access the help screen
2. The user clicks the help button
3. The help screen is shown to the user

Extensions:

- *a. The program fails.
    1. The user runs the game again.

Special Requirements:

- The Help screen should be readable and easy to understand

Technology and data variations:

- Provide the help screen in multiple languages
- Include a text version, and a visual version (provide video guide)

Frequency of occurence: Whenever the users wishes to seek help

Use Case UC2: Activate a Manual Power-Up

Scope: Gameplay

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to activate a manual power-up in their inventory

Preconditions: User has the power-up in their inventory.

Postconditions: The power-up is active

Main Success Scenario:

1. User gets a power-up
2. User clicks the power-up icon or presses the appropriate key
3. The game removes the power-up from the inventory
4. The game activates the power-up

Extensions:

- *a. The program fails
    - 1. The user runs the game again.
- *2a. The user clicks an icon/types a letter corresponding to a power-up they don't have
    - 1. The power-u doesn't activate
    - 2. The game continues without change

Special Requirements:

- The power-up icons are in the user's inventory. A power-up icon is transparent if the user doesn't have the power-up, solid they have it.

Technology and data variations:

- Power-up can be either activated using mouse or keyboard.

Frequency of occurence: The number of times a manual power-up is activated is either equal to or less than the number times manual power-ups are acquaired.

Use Case UC3: Authenticate

Scope: Game login screen

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to authenticate himself so that he can access his saved data and play the game

Preconditions: The game is initialized and on the login screen

Postconditions: The user is authenticated and has access to the game

Main Success Scenario:

1. User provides his username and password, chooses to continue
2. Game authenticates the User and shows a welcome message

3. User presses continue
4. Game changes the board to the building mode view

Extensions:

- *2a. The username or password is incorrect
  - 1. The Game informs the user that the login credentials are incorrect
  - 2. User enters the correct information
  - 3. Game authenticates the user and shows a welcome message
  - 4. User presses continue
  - 5. Game changes the board to the building mode view

Special Requirements:

- Password should be hidden from view
- Care should be taken when handling the password for security reasons

Technology and data variations:

- Keyboard might have caps lock on or might be in another language

Frequency of occurence: Once per game session

Use Case UC4: Break a Brick

Scope: Gameplay

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to destroy all the blocks in order to win the game

Preconditions: User is playing.

Postconditions: Brick is broken and removed from Board

Main Success Scenario:

1. Ball is moving around on the screen and heads towards the bottom of the screen
2. User moves the paddle to be directly below the ball: Include Move the Paddle
3. The ball hits the paddle
4. The Game changes the trajectory of the ball
5. The ball hits a brick
6. The Game removes the brick

Extensions:

- *a. The program fails.

-
    - 1. The user runs the game again.
- *3a. The paddle misses the ball and the balls falls off screen
    - 1. The player loses a life
    - 2. Game shows another ball (if the use still has a life) and the game continues
- *4a. The ball misses all bricks
    - 1. The ball reflects back to the player and we go back to step 1 in the main scenario
- *5a. The ball is a half-metal brick type and the ball hits it from the metal side
    - 1. The ball reflects back normally and the brick is not destroyed

Special Requirements:

- none

Technology and data variations:

- none

Frequency of occurence: Nearly continuous

Use Case UC5: Build a Map

Scope: Building mode

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to create a new map that can be loaded in future in order to play the game

Preconditions: User has a valid account with a username and password

Postconditions: The user-created map appears in the map list

Main Success Scenario:

1. The user enters building mode by clicking the “building mode” button
2. The user interacts with the Game to specify the number of each brick type
3. The Game creates bricks on random places
4. The user moves the randomly created bricks into new positions if s/he wants
5. The user saves the map if the minimum requirements are met
6. The map is saved

Extensions:

- *a. The program fails
    - 1. The User runs the game again and starts over
- 2a. The numbers don’t meet the requirements
    - 1. Change numbers to satisfy requirements

- 3a. The display cannot contains the number of bricks specified by the user
  - 1. The Game warns the user to decrease the number of bricks.
- 4a. The user tries to place a brick such it overlaps with another brick
  - 1. The Game warns the user to place the brick into another place where it doesn’t overlap with a brick.
- 5a. The minimum requirements are not met
  - 1. The Game warns the user to satisfy minimum requirements
- 6a. The disk is full so the map cannot be saved.
  - 1. Delete another map or increase the disk capacity

Special Requirements:

- The buttons and bricks on the screen should be visible
- Write permission for storage access

Technology and data variations:

- Provide different color schemes for color blind people
- Different storage devices such as cloud, local etc.

Frequency of occurence: At User's demand

Use Case UC6: Make an Account

Scope: Game login screen

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to have an account that they can use in order to play the game

Preconditions: User does not have an account and is on the login screen

Postconditions: User has a valid account with a username and password

Main Success Scenario:

1. User chooses the option of making a new account
2. The Game displays the account registration screen
3. User provides a username and password to the fields
4. Game creates an account for the user and informs the user that the account has been created successfully.

Extensions:

- *a. The program fails
  - 1. The User runs the game again and starts over

- 3a. The username is not valid
    - 1. The Game informs the user that the username entered is not valid
    - 2. User enters a valid username
- 3b. The password is not valid
    - 1. The Game informs the user that the username entered is not valid
    - 2. User enters a valid username
    - 3. Game creates an account for the user and informs the user that the account has been created successfully
- 4a. Another user with the chosen username already exists.
    - 1. The Game informs the user that a user with the chosen username already exists and displays login screen

Special Requirements:

- The password entered by the user should not be visible on the screen

Technology and data variations:

- Keyboard of the user might give input in varying languages

Frequency of occurence: Once per user

Use Case UC7: Fire Destructive Laser Gun

Scope: Gameplay

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: wants to destroy a brick using the destructive laser gun power-up

Preconditions: The Destructive Laser Gun is activated

Postconditions: There laser is fired

Main Success Scenario:

1. User presses the appropriate key or clicks.
2. The laser gun fires
3. The game decrements number of laser gun shots remaining by 1.
4. If the gun hits a brick, it gets destroyed.

Extensions:

- *3a. The shot fired was the last shot of the laser.
    - 1. The gun at the ends of the paddle dissappears.
    - 2. The Destructive Laser Gun becomes inactive, user can't fire again.

Special Requirements:

- The number of shots remaining are displayed above the laser gun.

Technology and data variations:

- Either mouse or keyboard can be used to fire the laser gun

Frequency of occurence: Destructive Laser Gun can be fired at most 5 times after being activated once.

Use Case UC8: Get a Power-Up

Scope: Gameplay

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to acquire the power-up to be able to use it later in the game

Preconditions: User is playing the game

Postconditions: User has a power-up in his inventory

Main Success Scenario:

1. User breaks a wrapper brick that contains a power-up
2. Game releases power-up from broken wrapper brick and makes it fall.
3. User moves the paddle to be directly below the power up: include Move The Paddle
4. Power-up touches the paddle.
5. Game removes the power-up and gives it to the User

Extensions:

- *a. The program fails
  - 1. The user runs the game again.
- *3a. The paddle misses the power-up
  - 1. The player doesn't get the power-up
  - 2. The game continues
- *5a. The power-up user gets is a manual power-up
  - 1. The power-up is added to the user's inventory
  - 2. The power-up is automatically activated.
- *5b. The automatic power-up is Destructive Laser Gun
  - 1. The power-up is automatically activated.
  - 2. A laser gun appears at the both ends of the paddle
  - 3. The Destructive Laser Gun power-up is active.
- *5c. The automatic power-up is Fireball
  - 1. The power-up is automatically activated.

    - 2. The ball changes to a fireball
    - 3. The fireball damages also the bricks next to one it hits
    - 4. The fireball can destroy metal sides of bricks in two hits.
    - 5. The fireball return to normal when the user loses it.
- *5d. The automatic power-up is Gang-of-balls
    - 1. The power-up is automatically activated.
    - 2. After the ball hits the paddle, it multiplies by 10.

Special Requirements:

- Any acquired manual power-ups are displayed at the inventory using icons.
- If the 10 balls created by the Gang-of-balls power-up move with the same speed, but with an angle equals to the ball index multiplied by 360 and divided by 10.

Technology and data variations:

- A keyboard is used to move the paddle.

Frequency of occurence: In a game session, a possibility every time a wrapper brick is broken.

Use Case UC9: Hit Harmful Alien

Scope: Gameplay

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to hit harmful aliens so that they disappear from screen.

Preconditions: At least one harmful alien appears in game screen.

Postconditions: At least one harmful alien has disappeared

Main Success Scenario:

1. User performs **Move the Paddle** in order to direct the ball towards the harmful alien
2. Game shows the user the movement of the ball on the board.
3. User directs the ball towards the harmful alien.
4. Harmful Alien is hit by the ball in the way that causes it to disappear.
5. Game removes Harmful Alien from board.

Extensions:

- *a. The program fails
    - 1. The user runs the game again.
- *4a. The ball does not hit any harmful alien.
    - 1. User performs **Move the Paddle** again to hit alien.

- *4b. Harmful Alien was not hit in the proper way that makes it disappear.
    - 1. User performs **Move the Paddle** again to hit the alien

Special Requirements:

- none

Technology and data variations:

- none

Frequency of occurence: Throughout the game.

Use Case UC10: Load Saved Game

Scope: The Game

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to continue a previously saved game

Preconditions: User has an account and at least one game saved by the same User

Postconditions: Game is resumed from the saved state.

Main Success Scenario:

1. User opens the "Load Game" menu
2. User picks one of the previously saved sessions
3. The game loads that session
4. The User can press "Resume" and plays the game.

Extensions:

- *a. The program fails.
    1. The user runs the game again.
- 3b. Session File is corrupt:
    1. Game refuses to continue loading
    2. Game returns to "Load Game" menu

Special Requirements:

- Access to disk space (Read permission)

Technology and data variations:

- Different storage devices (cloud, local, ...)

Frequency of occurence: Whenever the users wishes to

Use Case UC11: Move the Paddle

Scope: Gameplay

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to move the paddle in order to hit the ball or catch the falling power-up

Preconditions: User is playing

Postconditions: The position of the paddle has changed

Main Success Scenario:

1. The user predicts where s/he should move the paddle
2. The user moves the paddle by either pressing and releasing responsible buttons or keeping them down to move further
3. User can rotate the paddle by pressing required buttons up to {45, 135} degrees
4. The paddle moves according to input of the user
5. The paddle stops at the final location

Extensions:

- *a. The program fails.
  - 1. The user runs the game again
- *3a. The paddle stops
  - 1. The paddle hits the border of the game window so user can't move the paddle further

Special Requirements:

- A working keyboard is needed

Technology and data variations:

- none

Frequency of occurence: Nearly continuous

Use Case UC12: Pause the Game

Scope: Gameplay

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to pause the game in order to resume later

Preconditions: A game is in progress

Postconditions: The game is paused

Main Success Scenario:

1. User clicks on the pause button or presses the pause shortcut on the keyboard
2. Game halts the game and displays the pause screen
3. Game changes the pause button to a resume button

Extensions:

- *a. The program fails
    - 1. The User runs the game again and starts over

Special Requirements:

- The pause button on screen should be visible and easily identifiable as a pause button
- The button should be easily accessible on the keyboard

Technology and data variations:

- Different keyboard layouts to keep in mind

Frequency of occurence: Multiple times per game

Use Case 13: Quit the Game

Scope: The Game

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to quit the game

Preconditions: A game is in progress

Postconditions: The game is closed

Main Success Scenario:

1. User clicks on the quit button or shortcut on the keyboard
2. Game halts the game and displays a message to be confirmed by user

3. User confirms quitting
4. Game closes the game

Extensions:

- *a. The program fails
    - 1. The game is already closed, reached post condition

Special Requirements:

- The quit button on screen should be visible and easily identifiable
- The button should be easily accessible on the keyboard

Technology and data variations:

- Different keyboard layouts to keep in mind

Frequency of occurence: Once per game

Use Case 14: Release Magnetized Ball

Scope: Gameplay

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: wants to release the ball captured by the magnetized paddle

Preconditions: The Magnetized Ball power-up is active

Postconditions: The ball is released

Main Success Scenario:

1. The ball touches the paddle, Game stops it and makes it stuck to the paddle
2. The user presses the appropriate key or clicks
3. The Game releases the ball from the paddle
4. The Game deactivaes the Magnet power-up

Extensions:

- *1a. The paddle misses the ball
    - 1. The power-up is lost.

Special Requirements:

- After being relesead from the paddle, the ball preserves it's previous speed and direction.

Technology and data variations:

- Either keyboard or mouse can be used to release the ball.

Frequency of occurence: Happens once after every activation of a magnet power-up

---

Use Case 15: Resume the Game

Scope: The Game

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to continue the game from pause

Preconditions: The game is paused

Postconditions: The game is in progress

Main Success Scenario:

1. User clicks on the resume button or presses the resume shortcut on the keyboard
2. Game continues the game and displays the removes the pause screen
3. Game changes the resume button to a pause button

Extensions:

- *a. The program fails
    - 1. The User runs the game again and starts over

Special Requirements:

- There should be a short delay after choosing resume in order for the user to get ready.
- The resume button on screen should be visible and easily identifiable as a resume button
- The button should be easily accessible on the keyboard

Technology and data variations:

- Different keyboard layouts to keep in mind

Frequency of occurence: Multiple times per game

Use Case UC16: Save the Game

Scope: The game

Level: user goal

Primary Actor: User

Stakeholders and interests:

- User: Wants to save the state of game to resume at the same state later

Preconditions: The game is paused.

Postconditions: The game is saved according to its state at the moment of pausing.

Main Success Scenario:

1. Game shows the user the option menu.
2. User chooses the save option.
3. Game prompts user to enter a save name identifying the saved state of the game.
4. User enters a save name to identify the save.
5. Game saves the state of the game and links it with the save name the user entered.
6. Game shows the User that the operation is successful.

Extensions:

- *a. The program fails
    - 1. The user runs the game again.
- *3a. The storage disk is full.
    - 1. Game shows the user that there is no enough storage after choosing save option.
    - 2. User frees storage and retries to save steps.
- *5a. Save name entered by user already exists as a save name for a previously saved game.
    - 1. Game asks the user if he wants to overwrite this save name or choose another save name.
    - 2. User either chooses to overwrite, in which case previous save information is lost, or enter another save name.

Special Requirements:

- none

Technology and data variations:

- none

Frequency of occurence: At User's demand.
