

Use Cases

Play the New Game

Primary Actors: Player

Goal in Context Successfully control the dog through the game, collecting candies, avoiding threats, and finishing the game.

Precondition:

1. The game software is functional and loaded.
2. Player is at the main menu

Scenario:

1. Player click the New Game button
2. Player chooses the desired game level.
3. Game loads the chosen level.
4. Game loads the first level
5. Dog appears at the center of the screen.
6. Player uses arrow keys or controller to move the dog.
7. As the dog moves, the game system checks for any overlaps or collisions with candies, ghosts, spiders, or the exit door.
8. If the dog overlaps with candies: Candies disappear and score increases.
9. If the dog collides with a ghost: Dog loses a score and enters an "untouchable" state.
10. The game continues until: All candies have been collected and the dog reaches the exit door.

Exception

1. If the dog encounters a wall, it stops moving. see use case–Knock Wall
2. If the dog's score reaching negative: A "Game Over" message displays. see use case–Game Over
3. At any point, the player can access the in-game menu to "Save and Exit". See use case– save and exit
4. If the dog collides with a spider. A "Game Over" message displays. see use case–Game Over

Use Case: Knock Wall

Primary Actor: Dog

Goal in Context: Prevent the dog from passing through or moving into walls.

Preconditions: The dog is navigating the game grid.

Trigger: The dog moves towards a wall.

- Scenario:
1. The dog moves in a direction.
 2. The game detects a potential collision between the dog and the wall.
 3. The dog stops its movement upon contact with the wall.

Use Case: Game Over

Primary Actor: Game System

Goal in Context: Notify the player when the game conditions for losing are met.

Preconditions: The game is in progress.

Trigger: The dog's score reaches a negative value, or dog collides with spider.

- Scenario:
1. The game constantly checks the player's score.
 2. Once the score is detected to be negative, the game processes are halted.
 3. A "Game Over" message is displayed to the player.

Use Case: Save and Exit

Primary Actor: Player

Goal in Context: Save the current game progress and exit the game safely.

Preconditions: The game is in progress.

Trigger: Player decides to stop playing and wants to save the current game state.

Scenario:

1. Player accesses the pause or in-game menu.
2. Player selects the "Save and Exit" option.
3. The game saves the current progress, including the level, score, character's lives, and other relevant data.
4. A confirmation message is displayed indicating the game has been successfully saved.
5. The game closes or returns to the main menu, depending on design.

exception:

1. If there isn't enough storage space or there's an error while saving, a warning message is displayed. The game might not close immediately, allowing the player to retry saving.