1. **Title**: Jump
   **Actor**: Player
   **Precondition**: The application is running and a game has started, the character is currently grounded (standing on a platform or the ground).
   **Maine Success Scenario:**
   1. The player presses the key or button bound to the jump action.
   2. The system detects the jump input.
   3. The system applies upward force to the character, moving them upwards.
   4. The character's jump animation is played.
   5. The character begins moving upwards until they reach the peak of their jump.
   6. Gravity begins pulling the character downward.
   7. The character continues moving down until they collide with a solid surface (ground or platform) or another object.
   8. The character's jumping animation ends.

   **Alternatives:**
   - 1a: The character has limited jump power due to the duration of the press.
     - 1a1: The character jumps with a reduced height compared to the normal jump.
   - 2a: The player presses the jump button while the character is in mid-air (not touching the ground).
     - 2a1: No jump animation or force is applied
   - 3a: The character jumps and collides with an overhead obstacle
     - 3a1: The character's movement is stopped upon collision

   **Test Situations:**
   1) Jump with normal force from solid ground
   2) Jump with reduced force due to low duration of press
   3) Attempt to jump while in mid-air
   4) Jump and collide with a ceiling

   **Test Coverage:**
   Base: Number of main and alternate scenarios = 8
   Test situations cover all cases
   100% coverage of use case

2. **Title**: Move Left
   **Actor**: Player
   **Precondition**: The application is running and a game has started.
   **Maine Success Scenario:**
   1. The player presses the key or button bound to the "Move Left" action.
   2. The system detects the "Move Left" input.
   3. The character's facing direction changes to the left (if not already facing left).
   4. The character begins moving left at a constant speed.
   5. The character's running animation plays while moving left.
   6. The character continues moving left until the player releases the "Move Left" button/key, or encounters a collision

   **Alternatives:**

- 1a: The character is immobilized due to some gameplay effects
    - 1a1: No movement or animation changes happen
- 2a: The character encounters a solid object or wall while moving left.
    - 2a1: The character's movement stops on collision.

**Test Situations:**

1) Move left for a short duration
2) Move left for a long duration
3) Attempt to move left while immobilized
4) Move left and collide with a solid obstacle
5) Verify that character's facing direction changes to the left
6) Verify that character animation changes to running animation while moving left

**Test Coverage:**

Base: Number of main and alternate scenarios = 6
Test situations cover all cases
100% coverage of use case

3. **Title**: Move Right
**Actor**: Player
**Precondition**: The application is running and a game has started.
**Maine Success Scenario:**
    1. The player presses the key or button bound to the "Move Right" action.
    2. The system detects the "Move Right" input.
    3. The character's facing direction changes to the right (if not already facing right).
    4. The character begins moving right at a constant speed.
    5. The character's running animation plays while moving right.
    6. The character continues moving right until the player releases the "Move Right" button/key, or encounters a collision.

**Alternatives:**

- 1a: The character is immobilized due to some gameplay effects
    - 1a1: No movement or animation changes happen
- 2a: The character encounters a solid object or wall while moving right.
    - 2a1: The character's movement stops on collision.

**Test Situations:**

7) Move right for a short duration
8) Move right for a long duration
9) Attempt to move right while immobilized
10) Move right and collide with a solid obstacle
11) Verify that character's facing direction changes to the right
12) Verify that character animation changes to running animation while moving right

**Test Coverage:**

Base: Number of main and alternate scenarios = 6
Test situations cover all cases
100% coverage of use case

4. **Title**: Pause
**Actor**: Player
**Precondition**: The application is running and a game has started.
**Maine Success Scenario:**

1. The player clicks the pause button.
2. The system detects the pause button click.
3. The game state is frozen (all in-game action pauses).
4. The pause menu window displays on the screen.
5. The pause menu information fills in, including options to resume or choose a help screen

**Alternatives:**

- 1a: The pause button functionality is broken, or the menu elements are not properly loaded.
    - 1a1: The game might freeze, but the pause menu is not displayed.
    - 1a2: The system might display an error message indicating an issue
- 3a: Pausing the game causes a significant performance drop
    - 3a1: The game state might freeze, but with noticeable lag or visual glitches.

**Test Situations:**

1) Pause the game during normal gameplay
2) Attempt to pause the game multiple times in quick succession
3) Pause the game and explore the pause menu options
4) Test when pause menu fails to load
5) Pause the game on a low-performance system and monitor for lag

**Test Coverage:**
Base: Number of main and alternate scenarios = 5
Test situations cover all cases
100% coverage of use case