Enhanced Test Cases - Garbage Collector Game

Use Case References

This test document validates the following use cases:

- **UC-01: Host Game** - Creating multiplayer game sessions
- **UC-02: Join Game** - Connecting to hosted game sessions
- **UC-03: Move Character** - Player movement controls
- **UC-04: Collect Trash** - Trash item pickup mechanics
- **UC-05: Sort Trash** - Trash sorting and scoring system
- **UC-06: Win Game** - Victory condition handling

TC-01: Starting the Game

Name: Game Initialization and Menu Navigation Test

Test Objective: Verify that the multiplayer game mode can be started successfully and menu interactions work properly

Use Case Reference: UC-01: Host Game, UC-02: Join Game

Tested Request: Game startup and menu click events

Precondition:

- Game application is ready to launch
- Computer meets minimum system requirements
- No other instances of the game are running

Test Infrastructure:

- Computer with godot installed
- Mouse/keyboard input devices
- Display monitor

Positive Test:

Description of Test Steps:

1. Launch the game application
2. Click on "Start Game" button
3. Select Multiplayer mode
4. Verify game initializes properly

Input: Mouse click on "Start Game" button **Expected Output:** Game starts successfully, multiplayer mode becomes available **Expected Exceptions:** None

Result: ✅ WORKS - The mode starts fine

Postcondition: Game is running in multiplayer mode, ready for player connection

Negative Test:

Description of Test Steps:

1. Launch the game application
2. Click on random areas of the screen (non-interactive elements)
3. Verify no crashes or unexpected behavior occurs

Input: Random mouse clicks outside interactive elements **Expected Output:** No response to invalid clicks, game remains stable **Expected Exceptions:** None

Result: ✅ WORKS - No crashes

Postcondition: Game remains in stable state, ready for valid user input

---

TC-02: Player Movement

Name: Player Character Movement Control Test

Test Objective: Verify that player movement controls respond correctly to keyboard input

Use Case Reference: UC-03: Move Character

Tested Request: Keyboard input handling for player movement

Precondition:

- Game is running in multiplayer mode
- Player character is spawned in the game world
- Keyboard is connected and functional

Test Infrastructure:

- Running game instance
- Functional keyboard
- Game world with movement space

Positive Test:

Description of Test Steps:

1. Press W key - verify player moves up
2. Press S key - verify player moves down
3. Press A key - verify player moves left
4. Press D key - verify player moves right
5. Test arrow keys as alternative controls

Input: WASD keys and arrow keys **Expected Output:** Smooth player movement in corresponding directions **Expected Exceptions:** None

Result: ✅ WORKS - Player moves smoothly

Postcondition: Player character position updated according to input, movement system confirmed functional

Negative Test:

Description of Test Steps:

1. Press non-movement keys (spacebar, numbers, letters not assigned to movement)
2. Verify player character does not move
3. Confirm game remains responsive

Input: Invalid keys (spacebar, random letters/numbers) **Expected Output:** No player movement, no system errors **Expected Exceptions:** None

Result: ✅ WORKS - Player doesn't move with wrong keys

Postcondition: Player character remains in original position, input system properly filters invalid commands

---

TC-03: Multiplayer Connection

Name: Network Multiplayer Connection Test

Test Objective: Verify that two players can successfully connect and interact in multiplayer mode

Use Case Reference: UC-01: Host Game, UC-02: Join Game

Tested Request: Network connection establishment between host and client

Precondition:

- Two computers with game installed
- Both computers on same network
- Game launched on both systems
- Network connectivity confirmed

Test Infrastructure:

- Two computers with game installed
- Network connection (LAN/WiFi)
- Host and client game instances

Positive Test:

Description of Test Steps:

1. Computer 1: Click "Host" to create game session
2. Computer 2: Click "Join" and enter host IP address
3. Verify both players appear in game world
4. Test basic interaction between players

Input: Host creation command, IP address for joining **Expected Output:** Both players visible in shared game world **Expected Exceptions:** None

Result: ✅ WORKS - Both players connect and can see each other

Postcondition: Multiplayer session established, both players can interact in shared game environment

Negative Test:

Description of Test Steps:

1. Attempt to join with incorrect IP address
2. Verify appropriate error handling
3. Confirm game remains stable after failed connection

Input: Invalid IP address **Expected Output:** Connection failure message displayed **Expected Exceptions:** Connection timeout or invalid address error

Result: ✅ WORKS - Shows connection failed message

Postcondition: Game returns to menu state, ready for valid connection attempt

---

TC-04: Picking Up Trash

Name: Trash Item Collection Mechanism Test

Test Objective: Verify that players can successfully pick up trash items when in range

Use Case Reference: UC-04: Collect Trash

Tested Request: Item pickup interaction system

Precondition:

- Player is in active game session
- Trash items are spawned in game world
- Player character can move freely

Test Infrastructure:

- Active multiplayer game session
- Spawned trash items
- Functional keyboard input

Positive Test:

Description of Test Steps:

1. Move player character near a trash item
2. Press space bar to initiate pickup (as per UC-04)
3. Verify trash item is collected and follows player
4. Confirm item is removed from world

Input: Space bar press when near trash item **Expected Output:** Trash item attached to player character, item removed from ground **Expected Exceptions:** None

Result: ✅ WORKS - Trash gets picked up correctly

Postcondition: Player is carrying trash item, item no longer available for other players

Negative Test:

Description of Test Steps:

1. Position player far from any trash items
2. Press space bar multiple times
3. Verify no items are picked up inappropriately

Input: Space bar press when not near any items **Expected Output:** No pickup action occurs **Expected Exceptions:** None

Result: ✅ WORKS - Nothing happens when far from trash

Postcondition: Player remains without carried items, trash items remain in original positions

---

TC-05: Dropping Trash in Bins

Name: Trash Sorting and Scoring System Test

Test Objective: Verify that trash can be properly sorted into bins with correct scoring

Use Case Reference: UC-05: Sort Trash

Tested Request: Item disposal and scoring calculation system

Precondition:

- Player is carrying a trash item
- Sorting bins are available in game world
- Scoring system is active

Test Infrastructure:

- Game session with spawned bins
- Player carrying trash items
- Active scoring system

Positive Test:

Description of Test Steps:

1. Pick up paper trash item
2. Move to blue bin (paper bin)
3. Press space bar to drop item (as per UC-05)

4. Verify score increases by +1
5. Confirm item is properly disposed

Input: Space bar press near correct bin while carrying appropriate trash **Expected Output:** Score increases by 1, trash item disappears, bin accepts item **Expected Exceptions:** None

Result: ✅ WORKS - Score increases by 1 for correct sorting

Postcondition: Player score increased, trash item removed from game, bin updated

Negative Test:

Description of Test Steps:

1. Pick up plastic trash item
2. Move to wrong bin (not plastic bin)
3. Press space bar to drop item (as per UC-05)
4. Verify score decreases by -1 (Note: UC-05 states incorrect sorting removes trash without points, but test shows penalty)
5. Confirm penalty is applied correctly

Input: Space bar press near incorrect bin while carrying wrong trash type **Expected Output:** Score decreases by 1, penalty applied for incorrect sorting **Expected Exceptions:** None

Result: ✅ WORKS - Score decreases by 1 for wrong sorting

Postcondition: Player score penalized, incorrect sorting registered by system

---

TC-06: Winning the Game

Name: Victory Condition Achievement Test

Test Objective: Verify that game properly detects and handles win condition at 20 points

Use Case Reference: UC-05: Sort Trash (alternative flow), UC-06: Win Game

Tested Request: Win condition detection and game state management

Precondition:

- Player has score of 19 points
- Game is in active play state
- Scoring system is functional

Test Infrastructure:

- Active game session
- Player with 19 points
- Available trash and bins for final point

Positive Test:

Description of Test Steps:

1. Ensure player score is at 19 points
2. Pick up one trash item
3. Sort it correctly into appropriate bin
4. Verify score reaches 20 and win condition triggers

Input: Correct trash sorting action when at 19 points **Expected Output:** Score reaches 20, victory message/screen displayed **Expected Exceptions:** None

Result: ✅ WORKS - Game shows victory when reaching 20 points

Postcondition: Game in victory state, player declared winner, session may end or restart

---

TC-07: Trash Spawning

Name: Dynamic Trash Generation System Test

Test Objective: Verify that new trash items spawn at regular intervals during gameplay

Use Case Reference: UC-05: Sort Trash (step 6 - system spawns new trash item)

Tested Request: Automatic trash spawning mechanism

Precondition:

- Game session is active
- Initial trash items may or may not be present
- Spawning system is enabled

Test Infrastructure:

- Running game session
- Timer/observation capability
- Game world with spawn locations

Positive Test:

Description of Test Steps:

1. Start game and observe initial state
2. Wait and monitor for new trash appearance
3. Count frequency of new trash spawning
4. Verify spawning occurs regularly

Input: Time passage during active gameplay **Expected Output:** New trash items appear every few seconds **Expected Exceptions:** None

Result: ✅ WORKS - New trash spawns regularly

Postcondition: Game world continuously populated with new trash items for ongoing gameplay

---

TC-08: Score Display

Name: User Interface Score Tracking Test

Test Objective: Verify that player scores are accurately displayed and updated in real-time

Use Case Reference: UC-05: Sort Trash (step 4 - system updates score display)

Tested Request: UI score display and update system

Precondition:

- Game session is active
- Players are able to perform scoring actions
- UI elements are visible

Test Infrastructure:

- Active game with UI elements
- Players capable of scoring actions
- Visual display system

Positive Test:

Description of Test Steps:

1. Perform correct trash sorting actions
2. Perform incorrect trash sorting actions

3. Observe score changes in UI
4. Verify accuracy of displayed scores

Input: Various scoring actions (positive and negative) **Expected Output:** Score display updates immediately and accurately **Expected Exceptions:** None

Result: ✅ WORKS - Scores update and are clearly visible

Postcondition: Score display reflects accurate current scores for all players

---

TC-09: Player Animations

Name: Character Animation System Test

Test Objective: Verify that player character animations function correctly during gameplay

Use Case Reference: UC-03: Move Character (step 4 - system renders updated game state)

Tested Request: Character animation rendering and state management

Precondition:

- Player character is loaded in game world
- Animation systems are initialized
- Graphics rendering is functional

Test Infrastructure:

- Game session with rendered characters
- Animation system components
- Input devices for movement testing

Positive Test:

Description of Test Steps:

1. Test idle animation when character is stationary
2. Test movement animations during keyboard input
3. Verify smooth transitions between animation states
4. Check animation responsiveness to input

Input: Keyboard movement inputs and idle states **Expected Output:** Appropriate animations play for idle and movement states **Expected Exceptions:** None

Result: ❌ DOES NOT WORK - Player animations for movement are not working

Postcondition: Animation system identified as needing fixes, gameplay functionality not affected

---

Summary of Test Infrastructure Requirements

Hardware:

- Two computers for multiplayer testing
- Functional keyboards and mice
- Network connectivity (LAN/WiFi)
- Adequate display monitors

Software:

- Garbage Collector game installed on test machines
- Operating system compatible with game requirements
- Network configuration allowing local connections

Test Environment:

- Controlled testing environment
- Multiple test scenarios per feature
- Manual testing methodology
- Team of 5 testers for comprehensive coverage