

TestRail Summary Report: Snapshot 4

Jose Holguin, Andrew Wun, Alyssa Tu, Hyun Seok Song

May 9, 2025

1 Overview

This document summarizes the test cases for Inventory Access, Audio Interaction, and Task Progress, all of which were successfully tested and met the expected results.

2 Snapshot Summary Image

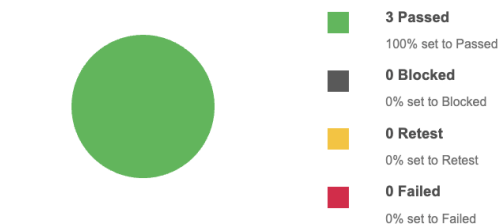


Final Touches and Future Implementations Test Runs (Summary) 5/9/2025

Project: Final Project - SoCal Edison Virtual Reality Training Simulation
By Alyssa Tu, 5/9/2025 2:45 AM

Tested three features in the VR environment: Audio and Sound Feedback, Inventory Access in VR, and Menu Task Screen: Task Progress in VR. All three test cases were completed and passed successfully. Everything worked as expected, sounds played properly, the inventory could be accessed and used, and the task progress was shown correctly on the menu screen.

Run: Final Touches and Future Implementations Test Run 5/8/2025



100%
passed
0 / 3 untested (0%).

2.1 Test Case #1

Name: Inventory Access in VR

Type: Functional

Priority: High

Estimated Time: 5 minutes

Actual Time: 5 minutes

Preconditions:

- Unity project is running in Play Mode.
- Menu screen is open in the VR environment.
- VR headset and controllers are connected and working.

Test Steps and Expected Results:

Step	Action	Expected Result	Actual Result
1	From the menu screen, locate the Inventory Access button.	The Inventory Access button should be visible and highlighted on the VR menu screen.	The Inventory Access button was visible and highlighted on the VR menu screen.
2	Point the VR controller at the Inventory Access button and hover over it.	The button should highlight (e.g., color change or visual effect).	The button highlighted (e.g., color change or visual effect).
3	Press the trigger on the VR controller to select the Inventory Access button.	The inventory UI should appear in the VR environment, displaying a list of available items.	The inventory UI appeared in the VR environment, displaying a list of available items.
4	Look through the inventory and attempt to select an item by pointing at it with the VR controller and pressing the trigger.	The selected item should become highlighted, and a description or additional options for the item should appear.	The system did not allow the user to select an item from the inventory when pressing the trigger.

2.2 Test Case #2

Name: Audio Interaction in VR

Type: Functional

Priority: High

Estimated Time: 5 minutes

Actual Time: 5 minutes

Preconditions:

- Unity project is running in Play Mode.
- Menu screen is open in the VR environment.
- VR headset and controllers are connected and working.

Test Steps and Expected Results:

Step	Action	Expected Result	Actual Result
1	Navigate to a section within the VR environment that includes sound or audio prompts (e.g., Smart Navigator installation or another task with sound).	The section should load successfully, and user should see objects or tasks designed to trigger sound effects, such as button clicks or task completions.	The section loaded successfully, and user saw objects/tasks designed to trigger sound effects, such as button clicks or task completions.
2	Interact with objects or tools that are designed to trigger sound effects (e.g., clicking on buttons, using tools).	Appropriate sound plays for each interaction: e.g., button clicks should sound like a button press, tools should make tool-specific sounds.	Appropriate sound played for each interaction: e.g., button clicks sounded like a button press, tools made tool-specific sounds.

2.3 Test Case #3

Name: Menu Task Screen: Task Progress in VR #3

Type: Functional

Priority: High

Estimated Time: 10 minutes

Actual Time: 10 minutes

Preconditions:

- The Unity project is running in Play Mode.
- The VR headset is on and functioning.
- The menu screen is visible and active in the VR environment.
- A task (e.g., Smart Navigator installation) has been started.

Test Steps and Expected Results:

Step	Action	Expected Result	Actual Result
1	Start a task, such as Smart Navigator installation, from the menu.	The task should begin, and the task progress bar should update on the Menu Task Screen.	The task began, and the task progress bar updated on the Menu Task Screen.
2	Monitor the task's progress on the Menu Task Screen.	The screen should automatically update the progression bar with real-time progress.	The screen automatically updated the progression bar with real-time progress.
3	Complete the task.	Upon completion, the task should be marked as complete on the Menu Task Screen.	Upon completion, the task was marked as complete on the Menu Task Screen.
4	Repeat the process for multiple tasks.	Each task should have an accurate progress indicator that tracks the task's completion.	Each task had an accurate progress indicator that tracked the task's completion.