

# John Wiesner: Excavator Simulator Test Plan

## Description of Overall Test Plan:

It is really hard to automate the testing of a VR application. You can automate triggers and game logic but anything the user must interact with must be tested manually. This is due to not being able to properly simulate the user's fov, head position, and user hand positions.

The overall test plan for this project is to test each main feature and list the steps to test said main feature. Once all the main features have been validated smaller edge cases will be tested using a list of steps for each. At the lowest level redundant logic will be added in case a user gets the simulator in an impossible state. This needs to be done since it is impossible to test every single scenario of a user in a virtual reality sandbox game.

## Test Case Descriptions

### UI1.1 Menu User Interface Test 1

UI1.2 Ensure menu UI buttons and sliders are interactable and to be used.

UI1.3 Determine if any menu UI buttons are not interactable. If UI buttons cannot be interacted with, they do not do what they should, or do not do what a user thinks should happen, documentation steps may be needed and/or a bug report may need to be created.

UI1.4 Inputs: User squeezes trigger of oculus controller while pointing at a menu UI item.

UI1.5 Results: Menu UI should navigate the user to the correct menu or apply changes to the game.

UI1.6 Normal

UI1.7 Whitebox

UI1.8 Functional

UI1.9 Integration

### UI2.1 Menu User Interface Test 2

UI2.2 Ensure save files are saved and loadable.

UI2.3 Makes sure user data is saved to the correct slot (out of 3) and that when switching profiles data is loaded correctly.

UI2.4 Inputs: User squeezes trigger of oculus controller while pointing at the main menu and selects play and slot number.

UI2.5 Results: Users should load into the main game menu with their correct amount of progress saved.

UI2.6 Normal

UI2.7 Whitebox  
UI2.8 Functional  
UI2.9 Integration

### **M1.1 Mission Test 1**

M1.2 Make sure controls tutorial is completable  
M1.3 VR controllers can control the excavator properly by moving and grabbing onto interactable objects highlighted in a blue bubble.  
M1.4 Inputs: User moves VR hands to proper highlighted areas and holds down clutch button.  
M1.5 Results: Visual feedback on the virtual controls should be present for each interaction and the excavator should perform some movement based on the selected control pattern setting.  
M1.6 Normal  
M1.7 Whitebox  
M1.8 Functional  
M1.9 Integration

### **M2.1 Mission Test 2**

M2.2 Make sure excavator digs and dumps dirt  
M2.3 Use a controller to move the excavator in such a way so the bucket scoops up dirt. The bucket should stop moving when full. If the bucket is lifted up and then dumped, dirt should fall out of the bucket and pile up on the ground.  
M2.4 Inputs: User grabs excavator joysticks and move excavator to dig and drop dirt  
M2.5 Results: Excavator creates a hole in the ground at the location of the bucket and then drops dirt below the bucket's location.  
M2.6 Normal  
M2.7 Whitebox  
M2.8 Functional  
M2.9 Integration

### **M3.1 Mission Test 3**

M3.2 Make sure excavator can pick up objects and place them down  
M3.3 Use a controller to move the excavator in such a way so the bucket and thumb grasp an object. Move the boom up and the object should still be in the bucket and thumb. Open the thumb or bucket and the object should fall out onto the ground.  
M3.4 Inputs: User grabs excavator joysticks and moves the excavator to pick up and drop an object.  
M3.5 Results: Excavator grabs, picks up, moves, and drops object.  
M3.6 Normal

M3.7 Whitebox  
M3.8 Functional  
M3.9 Integration

**M4.1 Mission Test 4**

M4.2 Make trees can be cleared on the Remove Trees mission  
M4.3 Use a controller to move the excavator in such a way to dig out trees, pick them up, move them to the highlighted areas.  
M4.4 Inputs: User grabs excavator joysticks and moves the excavator to dig two holes around a tree, picks up fallen tree, places tree in highlighted area.  
M4.5 Results: Excavator grabs, picks up, moves, and drops trees. Mission is cleared.  
M4.6 Normal  
M4.7 Whitebox  
M4.8 Functional  
M4.9 Integration

**M5.1 Mission Test 5**

M5.2 Make rocks can be cleared on the Remove Large Boulders mission  
M5.3 Use a controller to move the excavator in such a way to dig out rocks, pick them up, move them to the highlighted areas.  
M5.4 Inputs: User grabs excavator joysticks and moves the excavator to dig around a boulder, picks up rock, places rock in highlighted area.  
M5.5 Results: Excavator grabs, picks up, moves, and drops boulders. Mission is cleared.  
M5.6 Normal  
M5.7 Whitebox  
M5.8 Functional  
M5.9 Integration

**M6.1 Mission Test 6**

M6.2 Make sure pond mission can be passed  
M6.3 Use a controller to move the excavator in such a way to dig out dirt from the highlighted area in the Dig Pond mission.  
M6.4 Inputs: User grabs excavator joysticks and moves the excavator to dig a pond out while placing dirt outside of the highlighted area.  
M6.5 Results: Excavator digs out dirt from the highlighted area and the mission is cleared.  
M6.6 Normal  
M6.7 Whitebox  
M6.8 Functional

## M6.9 Integration

### M7.1 **Mission Test 7**

M7.2 Make sure Covert Pipe missions can be passed

M7.3 Use a controller to move the excavator in such a way to place down a culvert pipe. Then cover it with dirt.

M7.4 Inputs: User grabs excavator joysticks and moves the excavator to dig a trench for the culvert pipe, places pipe down, covers pipe again.

M7.5 Results: Excavator performs the required tasks and the user passes the mission.

M7.6 Normal

M7.7 Whitebox

M7.8 Functional

M7.9 Integration

### M8.1 **Mission Test 8**

M8.2 Make sure the Place In RipRap mission can be passed.

M8.3 Use a controller to move the excavator in such a way to grab scoops of rocks and lay them out for the drainage of the culvert pipe.

M8.4 Inputs: User grabs excavator joysticks and moves the excavator to scoop up rocks and place them in the highlighted area.

M8.5 Results: Excavator performs the required tasks and the user passes the mission.

M8.6 Normal

M8.7 Whitebox

M8.8 Functional

M8.9 Integration

### P1.1 **Performance Test 1**

P1.2 Make sure each scene runs above 90 fps on minimum supported hardware.

P1.3 Open each mission/scene, look around the scene in VR and make sure the fps doesn't drop below 90 fps.

P1.4 Inputs: User enters cheat code. User selects each scene from the menu and looks around.

P1.5 Results: Each scene should never drop below 90 fps once loaded in.

P1.6 Normal

P1.7 Blackbox

P1.8 Performance

P1.9 Unit Testing

### P2.1 **Performance Test 2**

P2.2 Quality settings work and reduce required processing power on CPU and/or GPU.

P2.3 Select different quality settings in the settings menu.

P2.4 Inputs: User selects low, normal, high, ultra

P2.5 Results: Level of detail in shadows, geometry, and post processing should be noticeable on the CPU and GPU.

P2.6 Normal

P2.7 Blackbox

P2.8 Performance

P2.9 Unit Testing

### **I1.1 Installation Test 1**

I1.2 Users can install VR simulator from steam.

I1.3 Open steam, uninstall application if necessary, install new version of app, make sure app runs.

I1.4 Inputs: Open steam, uninstall application if necessary, install new version of app, and open up app.

I1.5 Results: New version of app runs as expected

I1.6 Normal

I1.7 Blackbox

I1.8 Functional

I1.9 Unit Testing

### **I2.1 Installation Test 2**

I2.2 Users can update VR simulator from steam.

I2.3 Open steam, install old application if necessary, update new version of app, make sure app runs.

I2.4 Inputs: Open steam, install old application if necessary, update new version of app, and open up app.

I2.5 Results: Updated version of app runs as expected and user data has persisted.

I2.6 Normal

I2.7 Blackbox

I2.8 Functional

I2.9 Unit Testing

### **S1.1 Stability Test 1**

S1.2 If user save data does not exist, new data will be created.

S1.3 Delete save data in steam game folder and run app.

S1.4 Inputs: Delete save data in steam game folder and run app

S1.5 Results: New save data should be generated on game startup. If save data was not deleted, the user's save data should persist.

S1.6 Normal

S1.7 Whitebox  
 S1.8 Functional  
 S1.9 Unit Testing

## S2.1 **Stability Test 2**

S2.2 Corrupted or changed save data manipulation stability.  
 S2.3 Modify save data in steam game folder outside of steam and run app.  
 S2.4 Inputs: Modify save data in steam game folder to large values and/or letters where numbers should be and run app.  
 S2.5 Results: Any values outside of thresholds should be reverted to defaults and app loads normally.  
 S2.6 Normal  
 S2.7 Whitebox  
 S2.8 Functional  
 S2.9 Unit Testing

## Test Case Matrix

	<b>Normal/ Abnormal</b>	<b>Blackbox/ Whitebox</b>	<b>Functional/ Performance</b>	<b>Unit Testing/ Integration</b>
<b>UI1</b>	Normal	Whitebox	Functional	Integration
<b>UI2</b>	Normal	Whitebox	Functional	Integration
<b>M1</b>	Normal	Whitebox	Functional	Integration
<b>M2</b>	Normal	Whitebox	Functional	Integration
<b>M3</b>	Normal	Whitebox	Functional	Integration
<b>M4</b>	Normal	Whitebox	Functional	Integration
<b>M5</b>	Normal	Whitebox	Functional	Integration
<b>M6</b>	Normal	Whitebox	Functional	Integration
<b>M7</b>	Normal	Whitebox	Functional	Integration
<b>M8</b>	Normal	Whitebox	Functional	Integration
<b>P1</b>	Normal	Blackbox	Performance	Unit Testing
<b>P2</b>	Normal	Blackbox	Performance	Unit Testing
<b>I1</b>	Normal	Blackbox	Functional	Unit Testing
<b>I2</b>	Normal	Blackbox	Functional	Unit Testing
<b>S1</b>	Normal	Whitebox	Functional	Unit Testing
<b>S2</b>	Normal	Whitebox	Functional	Unit Testing