## **Tutorial 5 – Conditional OR and Object Behavior Methods**

Estimated Time: 90 minutes

### **Prerequisites**

IF statements, Else statements, Unity familiarity

### **Learning Objectives**

Students will learn about the conditional OR construct. Students will learn about methods that affect object behavior (e.g. changing object size, moving objects)

### **API Methods Covered**

* setObjectVelocity()
* setObjectSizeRelative()
* setObjectSizeAbsolute()
* setObjectPositionRelative()
* setObjectPositionAbsolute()

### **Activity**

1. Explain the different methods
   1. Setting the object velocity is equivalent to a one time push of an object, the object does not end up in a fixed position
   2. Setting an object’s position moves the object a fixed amount.
      1. setObjectPositionRelative() changes the object by the amount specified in the parameters. The original coordinates are used to calculate the new position.
      2. setObjectPositionRelative() moves the object to the new coordinates, without regard for the original coordinates.
   3. Setting an objects size by absolute or relative amounts work the same as changing the object position.
      1. setObjectSizeRelative() changes the object size by the amount specified. The original size is used to calculate the new size.
      2. setObjectSizeAbsolute() changes the object size to the new size specified, without regard for the original coordinates.
2. Explain the conditional OR construct
3. Set plane and add 2 different trees (from Vegetation Category)
4. Set script on cube (moveTreeA.cs)
   1. If the player holds the right trigger or holds the right grip, move the object up
5. Set script on sphere (moveTreeB.cs)
   1. If the player holds the left trigger or holds the left grip, add vertical velocity

### **Scripts**

**moveTreeA.cs**

void updateGame () {

if (isControllerRightTriggerDown() || isControllerRightGripDown())

{

setObjectPositionRelative(0, 1, 0);

}

}

**moveTreeB.cs**

void updateGame () {

if (isControllerLeftTriggerDown() || isControllerLeftGripDown())

{

setObjectVelocity(0, 5, 0);

}

}

### **Optional Tutorial 5B**

Estimated Time: 30 minutes

1. Create Machine object (SciFi, then machines category)
2. Set script on object (machine.cs)
   1. If the left or right controller is tilted forward, move the object along the z-axis
   2. If the player is one side of the game area or the other move the object to the center of the area

**machine.cs**

void updateGame () {

if (isControllerRightTiltedDown() | isControllerLeftTiltedDown())

{

setObjectPositionRelative(0, 0, 2);

}

if(getPlayerPositionX() > 5 || getPlayerPositionX() < -5){

setObjectPositionAbsolute(0, 0, 0);

}

}