## **Tutorial 3 – Else and Advanced Controller Interactions**

Estimated Time: 90 minutes

### **Prerequisite**

If Statements, Unity familiarity

### **Learning Objectives**

Students will learn the else statement. They will also learn methods that allow for more advanced controller interactions as well as methods to manipulate the player object.

### **API Methods Covered**

* isControllerTouching()
* isControllerRightTouching()
* isControllerLeftTouching()
* setTetherObjectToRightController()
* setTetherObjectToLeftController()
* isObjectTetheredToController()
* setObjectThrowable()
* isObjectGrabbed()

### **Activity**

1. Explain the different advanced controller groups
   1. Controller Touching – These methods will return true when the players controller interacts with the object that the script is on.
   2. Tether Objects – When an object is tethered to a controller, it allows for third-person like controls of the object. The object will move in the direction that the controller is tilted while it is tethered to the controller.
   3. Throwable Objects – When an object is set to throwable, it allows the player to pick up and throw the object. When the player controller collides with a throwable object and the player holds the controller trigger, the object will then move with the controller.
2. Demonstrate method groups by creating objects
   1. One plane for the player to move upon, and the object to reside
   2. One sphere on the plane to make throwable
   3. One cube on the plane to tether
3. Add Scripts to the sphere and cube
   1. Add a script (sphereThrow.cs), when the player touched the object, make it throwable
   2. Demonstrate an else statement by adding to the script that if an object is grabbed, change its color and it has text (“Throw me”);
   3. Add a script(cubeTether.cs), to tether the object when it is touched, else change color and add text (“tether me”)

### **Scripts**

**sphereThrow.cs**

void updateGame () {

if (isControllerTouching())

{

setTetherObjectToRightController(true);

setObjectColor("yellow");

setObjectText("");

}

else

{

setObjectColor("gray");

setObjectText("Tether me");

}

}

**cubeTether.cs**

void updateGame () {

if (isControllerTouching())

{

setObjectColor("red");

setObjectThrowable(true);

setObjectText("");

}

else

{

setObjectColor("blue");

setObjectText("Throw me");

}

}

### **Optional Tutorial 3B**

Estimated Time: 30 minutes

1. Create cube, make it throwable
2. Add script (colorChange.cs)
   1. If object is grabbed, make object yellow
   2. Else if right controller is tilted right make blue
   3. Else if right controller is tilted left, make red
   4. Else, make gray

#### ***Source Code***

**colorChange.cs**

void buildGame () {

setObjectThrowable(true);

}

void updateGame () {

if (isObjectGrabbed()){

setObjectColor("yellow");

}

else if (isControllerRightTiltedRight()){

setObjectColor("blue");

}

else if (isControllerRightTiltedLeft()){

setObjectColor("red");

}

else{

setObjectColor("gray");

}

### }

### **Optional Tutorial 3C**

Estimated Time: 30 minutes

1. Make Barrel (in BasicAssets)
2. Add script (barrelText.cs)
   1. If controller trigger is down, add text “trigger down”
   2. Else if controller grip is down, add text “grip down”
   3. Else, set text as (“no buttons pressed”)
   4. Set object text height of 1 in buildGame

#### ***Source Code***

**barrelText.cs**

void buildGame () {

setObjectTextHeight(1);

}

void updateGame () {

if(isControllerTriggerDown()) {

setObjectText("trigger down");

}

else if (isControllerGripDown())

{

setObjectText("gripDown");

}

else

{

setObjectText("No buttons pressed");

}

}