## **Tutorial 9 – Additional Methods**

Estimated Time: 60 minutes

### **Prerequisites**

If, else, conditionals

### **Learning Objectives**

Teach students the remaining API methods not covered elsewhere in the tutorials

### **API Methods Covered**

* setObjectTeleportable()
* setObjectVisible()
* setObjectGravity()
* setPlayerPositionRelative()
* setPlayerPositionAbsolute()
* addToHUD()

### **Activity**

1. Explain new API methods
   1. setObjectTeleportable() – The player is able to teleport to all objects by default. If an object is created that should not be teleported to, set this to false.
   2. setObjectVisible() – Makes the object visible or invisible
   3. setObjectGravity() – If an object has gravity, it will fall and be subject to other physics
   4. setPlayerPositionAbsolute/Relative – change the position of the player in the game
   5. addToHUD() – Add text to the corner of the screen, such as point value.
2. Create 1 plane (with the player starting on it) and another adjacent plane.
   1. Add a house (Structures Category) that hovers high above the adjacent plane
3. Add a script to the house (house.cs)
   1. When the controller trigger is down
      1. Gravity is added
      2. Text is added to the player HUD: “Falling House!!!”
   2. When the house collides with the adjacent plane, that plane is made teleportable and visible
4. Add a script to the adjacent plane that the player is not on (plane.cs)
   1. Make the plane non-teleportable
   2. Make the plane invisible
   3. When the house collides with the plane
      1. Enable teleportable and visible
      2. Set timer for 2 seconds
      3. When the timer is up, move the player to a position close to where the house is resting
      4. Add the text to the HUD: “Welcome to the house”

### **Scripts**

**house.cs**

void buildGame () {

setObjectName("floatingHouse");

}

void updateGame () {

if(isControllerTriggerDown())

{

setObjectGravity(true);

setHUD("Falling House!!!");

}

}

**plane.cs**

void buildGame () {

setObjectName("newPlane");

setObjectVisible(false);

setObjectTeleportable(false);

}

void updateGame () {

if(hasCollisionWithOtherObject("floatingHouse"))

{

setObjectTeleportable(true);

setObjectVisible(true);

startTimer(2);

}

if(isTimerFinished())

{

setPlayerPositionAbsoulte(-10, 0, 30);

setHUD("Welcome to the House!");

}

}

### **Optional Tutorial 9B**

Estimated Time: 30 minutes

1. Create table (Fantasy, Table Category)
2. Create Script (“change.cs”)
   1. Every 5 seconds change visible and teleportable state of object

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

**change.cs**

bool on = true;

void buildGame()

{

startTimer(5);

}

void updateGame()

{

if (isTimerFinished())

{

if (on == true)

{

setObjectVisible(false);

setObjectTeleportable(false);

on = false;

startTimer(5);

}

else

{

setObjectVisible(true);

setObjectTeleportable(true);

on = true;

startTimer(5);

}

}

}

}

### **Optional Tutorial 9C**

Estimated Time: 20 minutes

1. Create object (from SciFi, Decorative Elements)
2. Set object high above the play area
3. Create Script (gravity.cs)
   1. Add gravity to the object
   2. When a trigger is held, turn off gravity

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

**gravity.cs**

void buildGame()

{

setObjectGravity(false);

}

void updateGame()

{

if (isControllerTriggerDown())

{

setObjectGravity(true);

}

else

{

setObjectGravity(false);

}

}