Toggle Script

Technical Documentation

TriggerActivators and ToggleScript

In principle, a user roaming an interactive virtual environment uses some way of interaction (a trigger) to toggle a target object. As the object in question gets toggled, its state/properties are changed. This can then aid the overall impression, immersion, function and utility of the environment.

Such triggers can be, e.g.:

- Key presses / releases / holds (i.e., direct interface)
- User collisions with objects (interaction derived from direct interface movement)
- User raycasts targeting objects (reticle on the screen, eye tracking)
- Further interaction derivatives (e.g., grabbing and holding an object in VR)

TriggerActivators and ToggleScript (2)

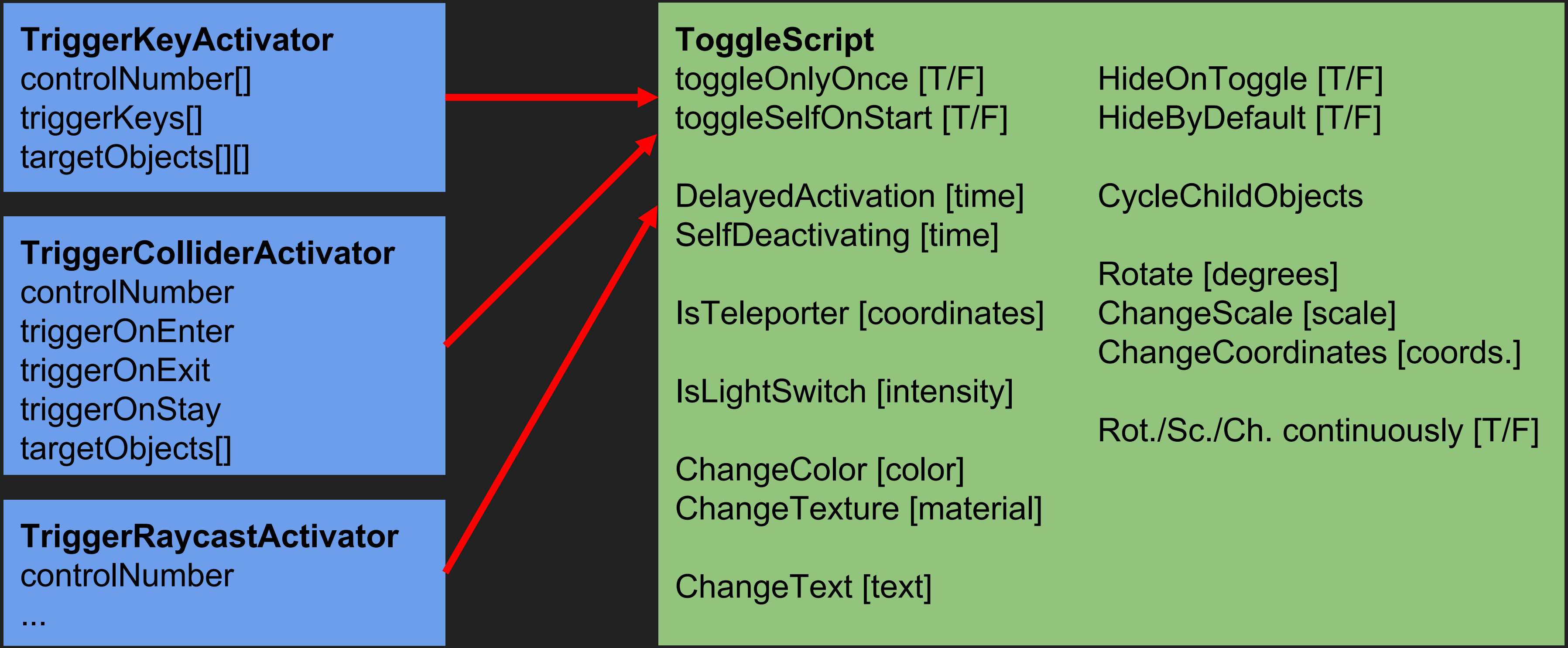
Therefore, TriggerActivators can take many forms.

This contrasts with the ToggleScript - it is a universal script, that once toggled (regardless of the source of such toggling) changes its state/properties.

Although some properties can only be toggled on special objects that have them in the first place (e.g., text values on canvases, light intensity on lights), this is still a single script.

Due to its universality, new TriggerActivators can be adapted to a ToggleScript; an object can trigger itself, or create a triggering chain of other objects; multiple triggers can toggle a single object, and they can create different results due to toggling different states of the object.

TriggerActivators and ToggleScript (3)



TriggerKeyActivator
controlNumber[]
triggerKeys[]
targetObjects[][]

TriggerColliderActivator
controlNumber
triggerOnEnter
triggerOnExit
triggerOnStay
targetObjects[]

TriggerRaycastActivator
controlNumber
...

ToggleScript
toggleOnlyOnce [T/F]
toggleSelfOnStart [T/F]

DelayedActivation [time]
SelfDeactivating [time]

IsTeleporter [coordinates]

IsLightSwitch [intensity]

ChangeColor [color]
ChangeTexture [material]

ChangeText [text]

HideOnToggle [T/F]
HideByDefault [T/F]

CycleChildObjects

Rotate [degrees]
ChangeScale [scale]
ChangeCoordinates [coords.]

Rot./Sc./Ch. continuously [T/F]

TriggerActivators: TriggerKeyActivator

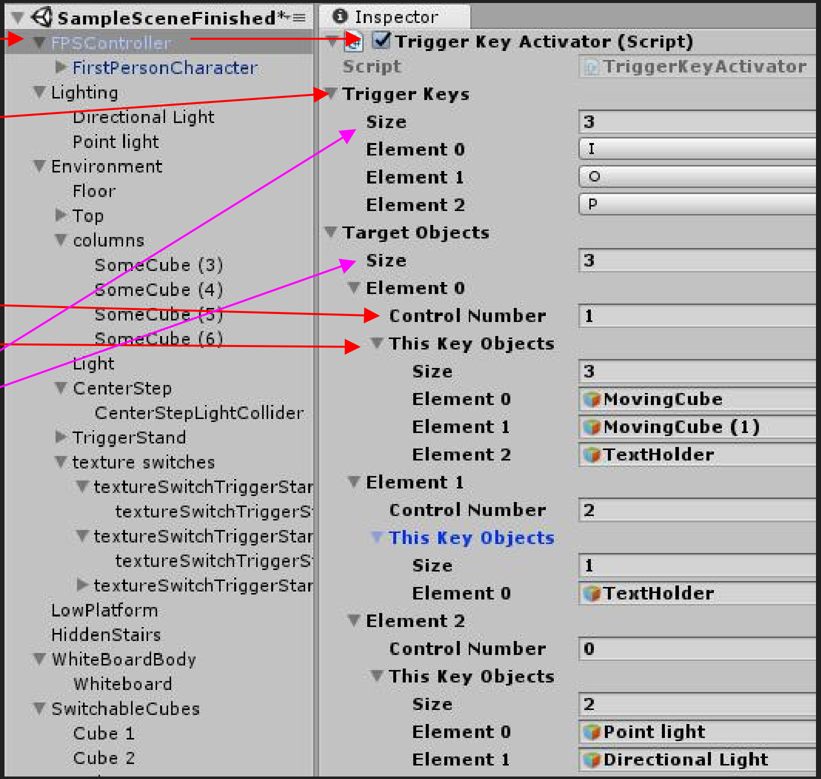
To be attached to an FPSController

Trigger Keys: keyboard keys that activate the trigger

Per each key:

- Control Number (to specify the state of ToggleScript)
- An array of objects to be toggled

Size of TriggerKeys should == size of TargetObject array



```
▼ SampleSceneFinished*
  ▼ FPSController
    ▶ FirstPersonCharacter
  ▼ Lighting
      Directional Light
      Point light
  ▼ Environment
      Floor
    ▶ Top
    ▼ columns
        SomeCube (3)
        SomeCube (4)
        SomeCube (5)
        SomeCube (6)
      Light
    ▼ CenterStep
        CenterStepLightCollider
    ▶ TriggerStand
    ▼ texture switches
      ▼ textureSwitchTriggerStar
          textureSwitchTriggerS
      ▼ textureSwitchTriggerStar
          textureSwitchTriggerS
      ▶ textureSwitchTriggerStar
    LowPlatform
    HiddenStairs
  ▼ WhiteBoardBody
      Whiteboard
  ▼ SwitchableCubes
      Cube 1
      Cube 2
```

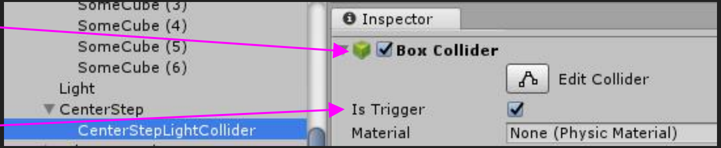
```
Inspector
▼ ☑ Trigger Key Activator (Script)
  Script                 TriggerKeyActivator
▼ Trigger Keys
    Size                 3
    Element 0            I
    Element 1            O
    Element 2            P
▼ Target Objects
    Size                 3
  ▼ Element 0
      Control Number     1
    ▼ This Key Objects
        Size             3
        Element 0        MovingCube
        Element 1        MovingCube (1)
        Element 2        TextHolder
  ▼ Element 1
      Control Number     2
    ▼ This Key Objects
        Size             1
        Element 0        TextHolder
  ▼ Element 2
      Control Number     0
    ▼ This Key Objects
        Size             2
        Element 0        Point light
        Element 1        Directional Light
```

TriggerActivators: TriggerColliderActivator

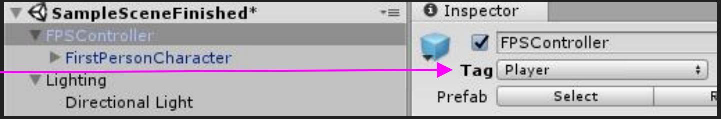
To be attached to an object with a collider to it

The collider has to have “IsTrigger” toggled

The FPSController has to be tagged as “Player”


SomeCube (3)
SomeCube (4)
SomeCube (5)
SomeCube (6)
Light
CenterStep
CenterStepLightCollider

Inspector
Box Collider ✓
Edit Collider
Is Trigger ✓
Material: None (Physic Material)


SampleSceneFinished*
FPSController
FirstPersonCharacter
Lighting
Directional Light

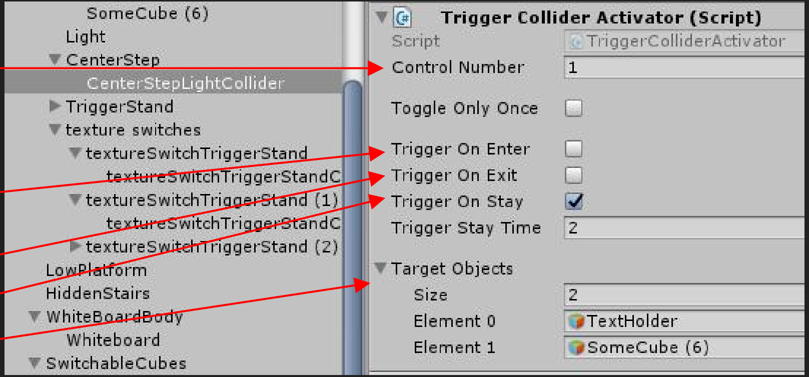
Inspector
FPSController ✓
Tag: Player
Prefab: Select

Control Number per this collider

Specify when the trigger gets executed:

- When the collider gets entered
- When exited (can be combined with the aforementioned)
- When inside the collider for a specified amount of seconds

A list of objects that get triggered


SomeCube (6)
Light
CenterStep
CenterStepLightCollider
TriggerStand
texture switches
textureSwitchTriggerStand
textureSwitchTriggerStandC
textureSwitchTriggerStand (1)
textureSwitchTriggerStandC
textureSwitchTriggerStand (2)
LowPlatform
HiddenStairs
WhiteBoardBody
Whiteboard
SwitchableCubes

Trigger Collider Activator (Script)

Script	TriggerColliderActivator
Control Number	1
Toggle Only Once	☐
Trigger On Enter	☐
Trigger On Exit	☐
Trigger On Stay	✓
Trigger Stay Time	2
Target Objects	
Size	2
Element 0	TextHolder
Element 1	SomeCube (6)

TriggerActivators: TriggerRaycastActivator

Static Raycast: Reticle in the middle of the screen

Dynamic Raycast: Eye-tracking gaze

To be implemented later...

TriggerActivators: Derivatives

TriggerKeyInColliderActivator

Keypress is receptive only if the user remains within the bounds of a collider that enables this. As a result of which, localized keyboard/controller interface is available.

TriggerRaycastInColliderActivator

Similarly to the aformentioned, triggering of objects happens only if (1) the user is in the vicinity of a collider and/or (2) if the object is in collider’s range, too.

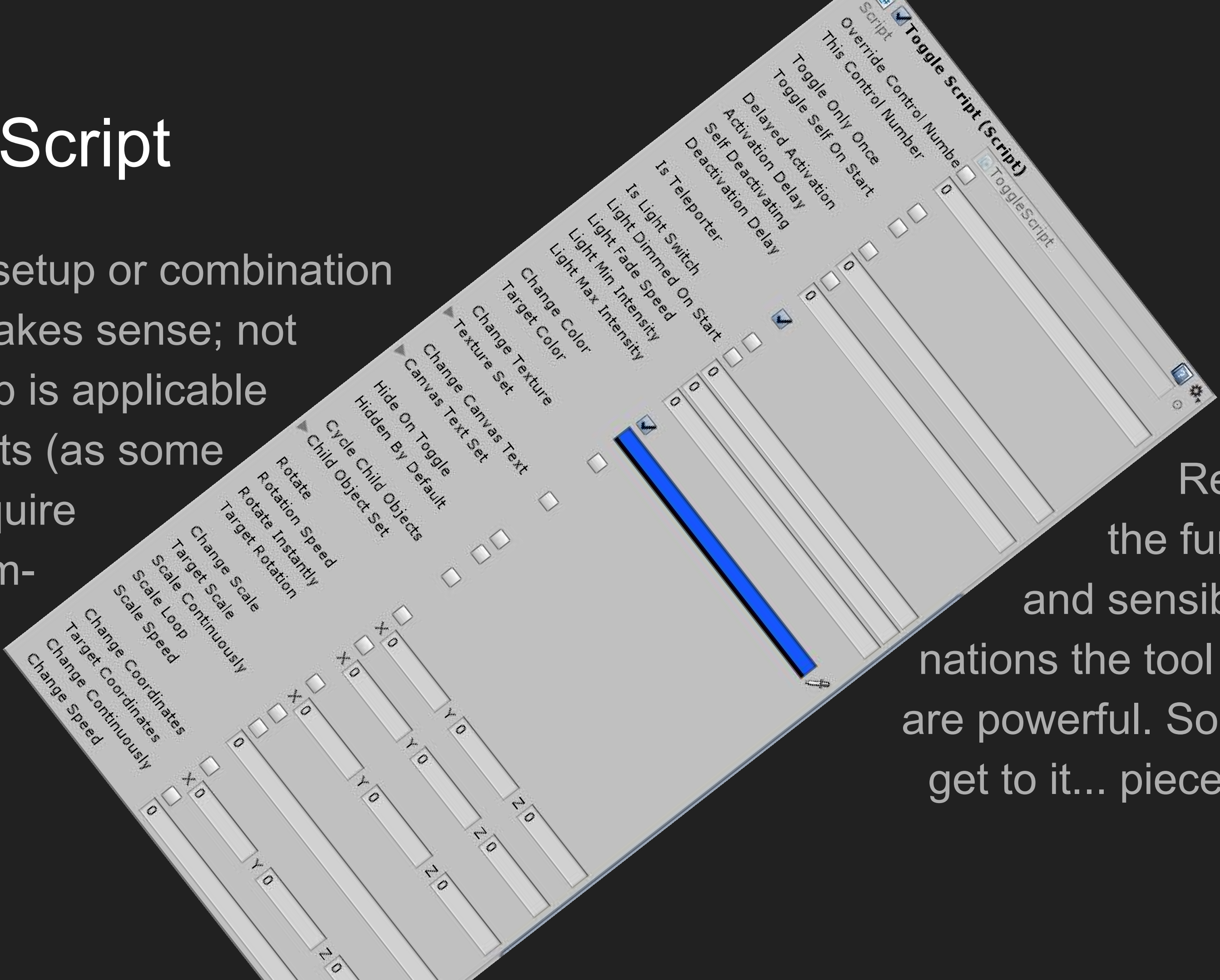
TriggerServer(?)

A solution that is designed as an intermediary of multiple triggers, for more complex scenarios. A prescription: what to do with object(s) when a specific trigger is called. The trigger server can also store object states (n, n+1...), iterate on them.

All: To be implemented later...

ToggleScript

Not every setup or combination of items makes sense; not every setup is applicable to all objects (as some objects require special components.



Toggle Script (Script)

Script ToggleScript

Override Control Number ✓

This Control Number 0

Toggle Only Once

Toggle Self On Start

Delayed Activation

Activation Delay 0

Self Deactivating

Deactivation Delay 0

Is Teleporter

Is Light Switch ✓

Light Dimmed On Start

Light Fade Speed 0

Light Min Intensity 0

Light Max Intensity 0

Change Color ✓

Target Color

Change Texture

Texture Set

Change Canvas Text

Canvas Text Set

Hide On Toggle

Hidden By Default

Cycle Child Objects

Child Object Set

Rotate

Rotation Speed X 0 Y 0 Z 0

Rotate Instantly

Target Rotation X 0 Y 0 Z 0

Change Scale

Target Scale X 0 Y 0 Z 0

Scale Continuously

Scale Loop

Scale Speed 0

Change Coordinates

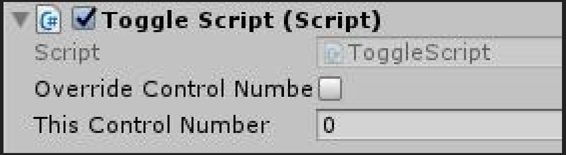
Target Coordinates X 0 Y 0 Z 0

Change Coordinates Continuously

Change Speed 0

Regardless, the functionality and sensible combinations the tool allows for are powerful. So we better get to it... piece by piece.

ToggleScript: controlNumber override


Toggle Script (Script)

Script	ToggleScript
Override Control Numbe	☐
This Control Number	0

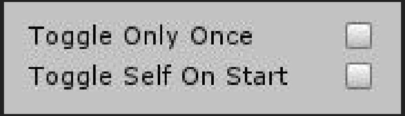
A simple object with a ToggleScript to it is just on/off.

The on/off states are taken care of from within the script.

For more complicated objects with multiple possible states, controlNumbers are provided, and generally so from a Trigger. This is good enough a solution 99% the time. If, however, an override needs to be specified, this is the setting for it.

When “OverrideControlNumber” is active, the ToggleScript ignores controlNumbers sent to it by Triggers, choosing to self-inject a controlNumber specified here instead.

ToggleScript: toggleOnlyOnce, toggleSelfOnStart

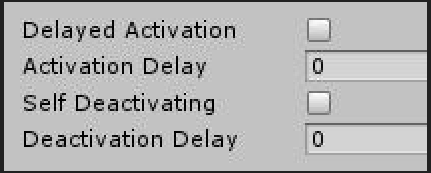

Toggle Only Once ☐
Toggle Self On Start ☐

If an object is required to get triggered to a state and stay there, using the “toggleOnlyOnce” option is the way. Such object then remains in this state for the rest of the virtual experience.

To toggle an object to a default activation state right on start, use the “toggleSelfOnStart” option. It will be toggled with no user input required.

These two options can easily be combined to create an object defined in an active state (e.g., rotating along its axis) with no option of any further input turning it off.

ToggleScript: delayedActivation, selfDeactivating



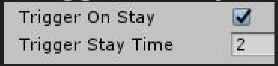
Delayed Activation	☐
Activation Delay	0
Self Deactivating	☐
Deactivation Delay	0

In standard cases, ToggleScript activates itself when triggered.

However, if some delay is to be introduced into this activation, the two options of “DelayedActivation” and “SelfDeactivating” come in handy.

“DelayedActivation” stores the controlNumber passed to it externally and waits. Once the ActivationDelay, in seconds, is over, the object activates itself.

FYI, an object can also be activated with delay using “TriggerOnStay” option of a TriggerColliderScript. The difference is that “DelayedActivation” activates the object in all cases, while for “TriggerOnStay” to execute, one has to stay within a collider for a specified amount of time (in seconds).

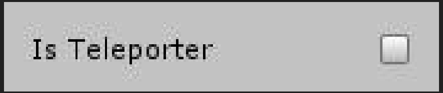


Trigger On Stay	✓
Trigger Stay Time	2

A “SelfDeactivating” objects waits for a specified amount of time after activation and then it deactivates itself (that is, it toggles self with controlNumber == 0).

ToggleScript: isTeleporter

When such object is toggled, user is teleported to its location.


Is Teleporter ☐

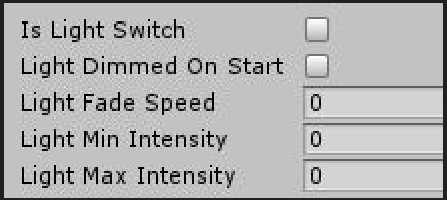
It is recommended for an object like this to have no renderer, or conventional collider, as it is merely to serve as a 3D coordinate to which a user is to be teleported (to prevent any clipping, or graphical glitches).

Some recommendations:

- Let the user know beforehand there's a teleporter (by text, signs, effects, etc.), so that they don't become puzzled by the sudden change of locations
- Let the teleporter (e.g., a collider) have visual cues to it (e.g., a pad)
- Use it as a time-saver that bridges long distances across a vast environment
- Test for collisions (so that the user doesn't fall through the floor, etc.)

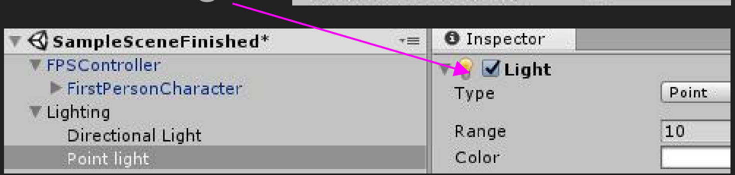
ToggleScript: isLightSwitch

Such an object is to serve as an artificial on/off light source.



Is Light Switch	☐
Light Dimmed On Start	☐
Light Fade Speed	0
Light Min Intensity	0
Light Max Intensity	0

For this functionality to work, the object needs to have a Light (of any kind) attached to it.



SampleSceneFinished*
FPSController
FirstPersonCharacter
Lighting
Directional Light
Point light

Inspector
✓ Light

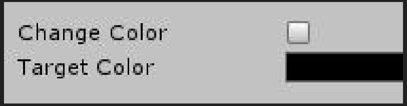
Type	Point
Range	10
Color	

When the virtual scene starts, the intensity of the object's light is set to the value of "LightMaxDensity" (float, with recommede values in [0-5]). Once the object is triggered, light intensity begins to fade back into the "LightMinDensity" value (which can be even zero, for no light at all). "LightFadeSpeed" determines how long such transition takes.

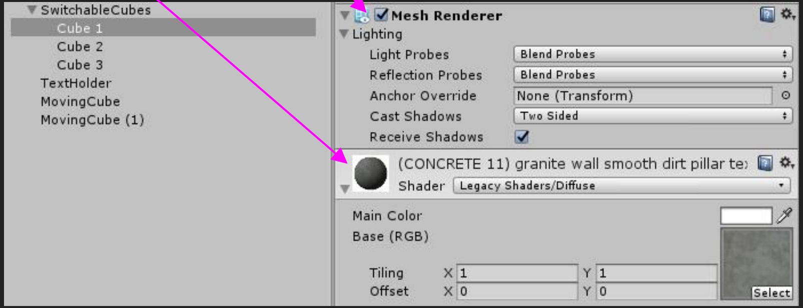
If "LightDimmedOnStart" option is enabled, the process is reversed (going from Min to Max).

ToggleScript: changeColor

This function serves as a simple object highlighter.


Change Color ☐
Target Color ■

For this to work, the object has to have a MeshRenderer, and a Shader (any shader with basic diffuse texture will do).


SwitchableCubes
- Cube 1
- Cube 2
- Cube 3

TextHolder
MovingCube
MovingCube (1)

Mesh Renderer ✓
Lighting

Light Probes	Blend Probes
Reflection Probes	Blend Probes
Anchor Override	None (Transform)
Cast Shadows	Two Sided
Receive Shadows	✓

(CONCRETE 11) granite wall smooth dirt pillar te:
Shader Legacy Shaders/Diffuse

Main Color
Base (RGB)

Tiling	X 1	Y 1
Offset	X 0	Y 0

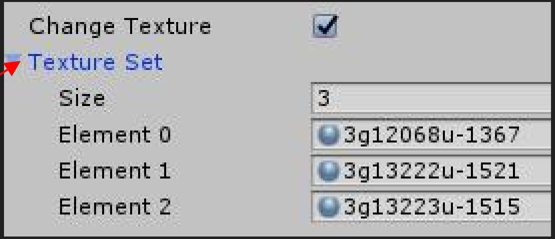
Select

Once it is triggered for the first time, the ToggleScript will change Shader color to the one specified by "TargetColor".

When triggered again, Shader color is reverted back to its original value.

ToggleScript: changeTexture

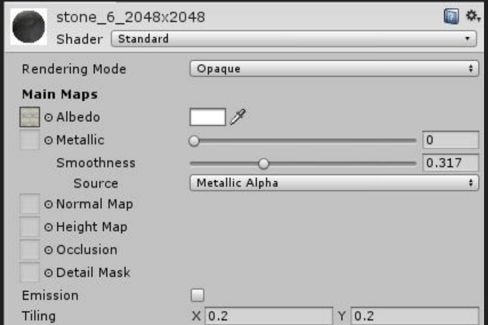
This is a potentially more powerful version of the aforementioned changeColor function. This time, the whole Shader of an object is Changed.



Change Texture	✓
Texture Set	
Size	3
Element 0	3g12068u-1367
Element 1	3g13222u-1521
Element 2	3g13223u-1515

Same requirements (a MeshRenderer attached to an object) apply.

When triggered with controlNumber >= 1, ToggleScript assigns a Shader to the object based on the TextureSet list. When controlNumber = 0, object's original shader is reapplied.



stone_6_2048x2048
Shader Standard

Rendering Mode	Opaque
Main Maps	
Albedo	
Metallic	0
Smoothness	0.317
Source	Metallic Alpha
Normal Map	
Height Map	
Occlusion	
Detail Mask	
Emission	
Tiling	X 0.2 Y 0.2

All sorts of Shaders with various rich/prescripted properties can be applied to the object.

ToggleScript: changeCanvasText

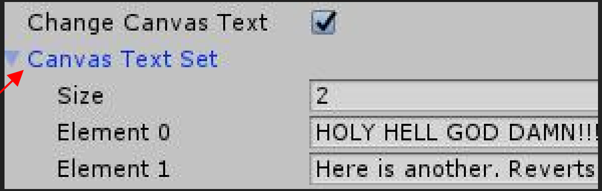
This is to change the value of a text container.

For this to work, the object has to have Canvas and Text attached to it (with no Canvas, no Text will render).

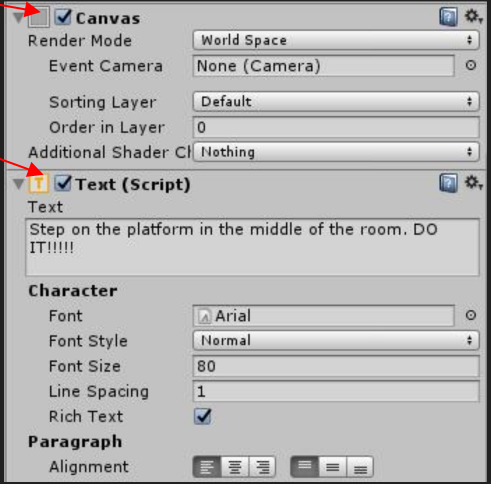
When called with controlNumber >= 1, the text value will change to a value per the CanvasTextSet list. When controlNumber == 0, original text will return.

Text per GUI (screenSpace) or scattered across an environment (worldSpace) can be of a significant (navigational or stylistic) help. One is advised to experiment with this - e.g. by playing around with text size (e.g. huge man-sized labels spanning across corridors). Original text can be empty, too - only showing a label in place on trigger.

However, screenSpace text in VR (fixated to one's camera) is usually not a good idea.

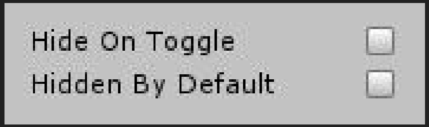


Change Canvas Text	✓
Canvas Text Set	
Size	2
Element 0	HOLY HELL GOD DAMN!!!
Element 1	Here is another. Reverts



Canvas	✓
Render Mode	World Space
Event Camera	None (Camera)
Sorting Layer	Default
Order in Layer	0
Additional Shader Cl	Nothing
Text (Script)	✓
Text	Step on the platform in the middle of the room. DO IT!!!!!
Character	
Font	Arial
Font Style	Normal
Font Size	80
Line Spacing	1
Rich Text	✓
Paragraph	
Alignment	

ToggleScript: hideOnToggle, hiddenByDefault


Hide On Toggle ☐
Hidden By Default ☐

Sometimes an object needs to be simply turned off.

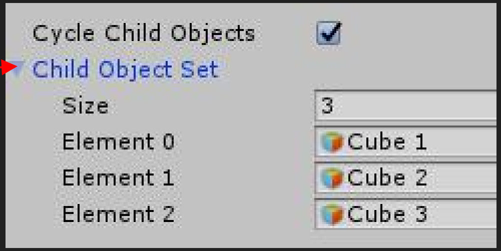
By enabling the “HideOnToggle” option, the object, once triggered, ceases to exist, in all of it (its Mesh, Collider, Light, etc., whatever it might be). When triggered again, it reappears. The “HiddenByDefault” option reverses this dynamic.

This can come in handy when just one object of many is to be presented, or when it is to be combined with an animation.

For a more complex scenario: If the object has its own thisControlNumber and controlNumber is called in a trigger, and thisControlNumber != controlNumber, the objects gets hidden. If thisControlNumber == controlNumber, it is displayed.

ToggleScript: cycleChildObjects

This is an all-in-one solution similar to the previous one.



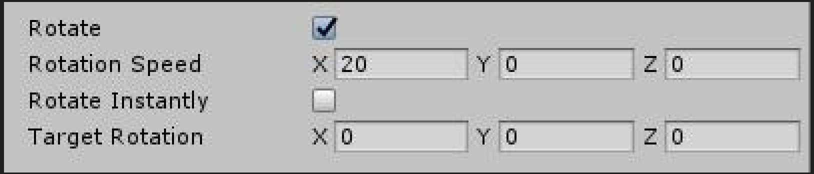
Cycle Child Objects	✓
Child Object Set	
Size	3
Element 0	Cube 1
Element 1	Cube 2
Element 2	Cube 3

A List of objects is defined (these don't need to be actual child objects of the current one). Once the scene is initiated, they are all set to be hidden. Then, once a trigger toggles this object with a controlNumber in mind, a child object in the List that corresponds to the controlNumber is displayed. The rest of them remain hidden.

This is useful for certain one-in-many objects collections or lists (e.g., exhibitions.)

Keep in mind: while a HideOnToggle called on an object hides the object along with its children objects, this function does not do so. When cycling child objects, the parent object remains intact.

ToggleScript: rotate, rotateInstantly



Rotate	✓		
Rotation Speed	X 20	Y 0	Z 0
Rotate Instantly	☐		
Target Rotation	X 0	Y 0	Z 0

Why would anyone rotate an object?

Perhaps to show it off…

There needs to a MeshRenderer to have anything to rotate. Then, once the object is triggered, it starts rotating along its axes, as per its RotationSpeed. Once triggered again, the object stops rotating, returning to its original position.

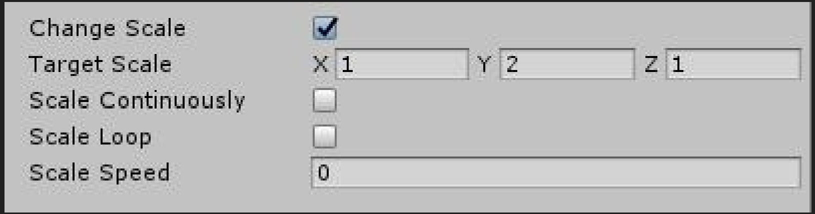
If the “RotateInstantly” option is on, there will be no continuous rotation animation. Instead, the object will rotate instantly per specified X/Y/Z degrees, as per TargetRotation.

When is the “RotateInstantly” option more useful? When rotating objects like the sunlight (directionalLight).

ToggleScript: changeScale, scaleContinuously, scaleLoop

In principle, this is similar to rotations.

Only this time, the object is set to scale along one or more of its axes.

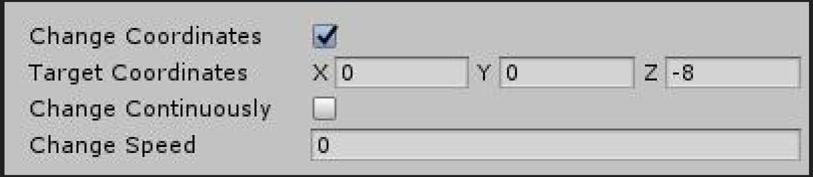


Change Scale	✓
Target Scale	X 1 Y 2 Z 1
Scale Continuously	☐
Scale Loop	☐
Scale Speed	0

With “ChangeScale” option on and triggered, the object scales instantly. Once triggered again, it returns to its original scale.

When “ScaleContinuously” is also enabled, the object scales in an animation, according to its ScaleSpeed. Finally, when the “ScaleLoop” option is enabled, the triggered object oscilates from its original scale to its TargetScale and back, until it is triggered again.

ToggleScript: changeCoordinates, changeContinuously



Change Coordinates	✓
Target Coordinates	X 0 Y 0 Z -8
Change Continuously	☐
Change Speed	0

Finally, an object can be moved.

When the “ChangeCorrdinates” option is enabled and object triggered, the object moves instantly to its new coordinates. Once triggered again, it is returned to its origin. The specified TargetCoordinates are relative to the original position (in this example, the object is moved 8 meters back).

“ChangeContinuously” and ChangeSpeed allow for dynamic transition. Also:

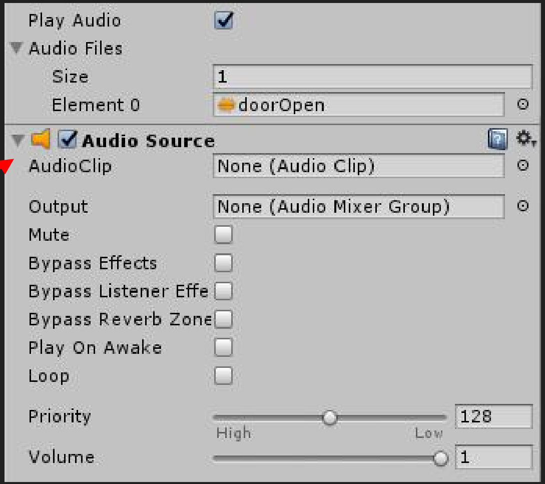
- Rotation / scaling / moving can be combined, oftentimes to decent results
- Rotation / scaling / moving cannot be applied to ProBuilder objects, as they don’t support these operations (however, ProBuilder object to standard object conversion (or vice versa) is a thing)

ToggleScript: playAudio

To play an audio file (a sound, or a music segment)

To do this, enable the “playAudio” option and add some audioFiles.

Then, make sure the object has an audioSource added to it (so that it would be able to play the sound).
Furthermore, feel free to play around with audio settings below.



Play Audio ✓
Audio Files
Size 1
Element 0 doorOpen

Audio Source ✓
AudioClip None (Audio Clip)
Output None (Audio Mixer Group)
Mute ☐
Bypass Effects ☐
Bypass Listener Effe ☐
Bypass Reverb Zone ☐
Play On Awake ☐
Loop ☐
Priority 128 (High – Low)
Volume 1

When controlNumber >= 1, specifed audioFile is played.

When controlNumber == 0, audio stops.

Examples

So much documentation for ToggleScript and nothing to show for it… yet.

The following slides describe a few demonstrations of what can be achieved with ToggleScript, beginning with the simplest examples and expanding on to a few more complex ones.

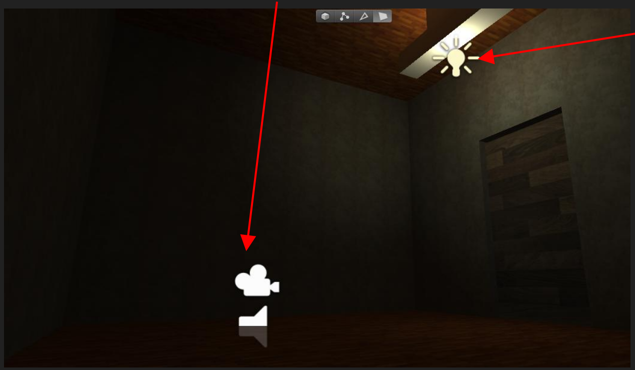
Certainly, this is not some end-all-be-all exhaustive demonstration of all its potential capabilities. One is encouraged to experiment on their own, to come up with original solutions of their own making.

Furthermore, there’s a “live” demo (a Windows build and a UnityPackage) available on the following address:

https://drive.google.com/drive/folders/1STw5OSoqeWVqlL7znQBbQUnyESAwmZRt?usp=sharing

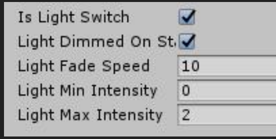
Example 1: switching a light on by pressing a key

It is an empty room, with an FPSController, and a Light.

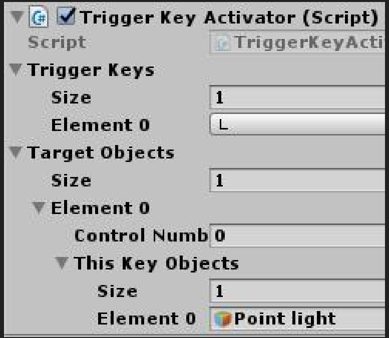


Add a ToggleScript to the Light.
Check “IsLightSwitch” and “IsDimmedOnStart”.
Set the following:

- LightFadeSpeed: 10
- LightMinIntensity: 0
- LightMaxIntensity:2



Is Light Switch	✓
Light Dimmed On St	✓
Light Fade Speed	10
Light Min Intensity	0
Light Max Intensity	2



Trigger Key Activator (Script)	
Script	TriggerKeyActi
Trigger Keys	
Size	1
Element 0	L
Target Objects	
Size	1
Element 0	
Control Numb	0
This Key Objects	
Size	1
Element 0	Point light

Go ahead, add a TriggerKeyActivator to the FPSController.
Set TriggerKeys to 1, and pick “L”.
Then set TargetObjects to “1”,
and ThisKeyObjects to “1” as well.
Drag and drop the Light onto the new Element.
Key “L” is now set to trigger the Light.
Is is up to the Light’s ToggleScript to determine the action.

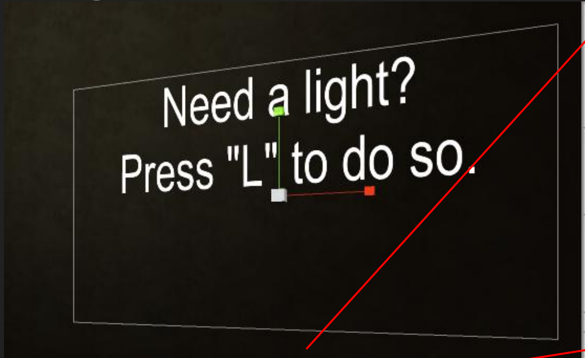
Do these two thing, and you are done!

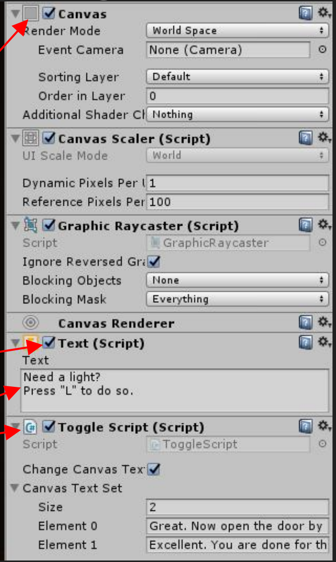
Once you run your application, you will enter a dark room.

Then, as you press “L”, the light will go on.
Pressing it again will switch the light off.

Example 2: writings on the wall

However, the user doesn't know how to control the light. Furthermore, it is just a single closed room. Let's change that...

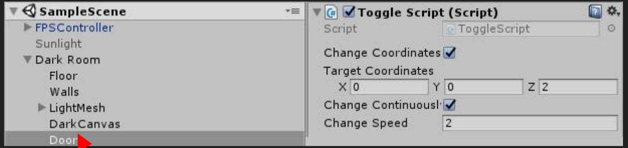




```
Canvas
  Render Mode              World Space
  Event Camera             None (Camera)
  Sorting Layer            Default
  Order in Layer           0
  Additional Shader Cl     Nothing
Canvas Scaler (Script)
  UI Scale Mode            World
  Dynamic Pixels Per U     1
  Reference Pixels Per     100
Graphic Raycaster (Script)
  Script                   GraphicRaycaster
  Ignore Reversed Gra      ✓
  Blocking Objects         None
  Blocking Mask            Everything
Canvas Renderer
Text (Script)
  Text
    Need a light?
    Press "L" to do so.
Toggle Script (Script)
  Script                   ToggleScript
  Change Canvas Tex        ✓
  Canvas Text Set
    Size                   2
    Element 0              Great. Now open the door by
    Element 1              Excellent. You are done for th
```

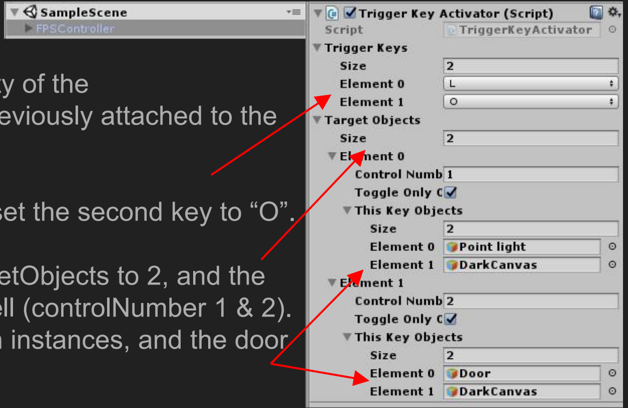
Add a Canvas to the scene (GameObject -> UI -> Canvas). Then add a Text to it. Default text value is to be this: "Need a light? Press L to do so"

Add a ToggleScript to the Canvas object.
Check the "ChangeCanvasText" option, with two addition values.
Fill them in:
[0] "Great. Now open the door by pressing O".
[1] "Excellent. You are done for this room."



```
SampleScene
  FPSController
  Sunlight
  Dark Room
    Floor
    Walls
    LightMesh
    DarkCanvas
    Door

Toggle Script (Script)
  Script                   ToggleScript
  Change Coordinates       ✓
  Target Coordinates
    X 0    Y 0    Z 2
  Change Continuousl       ✓
  Change Speed             2
```

Create a Door object (a simple rescaled cube will do). Add a ToggleScript to it, and check the "ChangeCoordinates" option. Then, to make the animation smooth, check "ChangeContinuously", and for the ChangeSpeed, input 2.



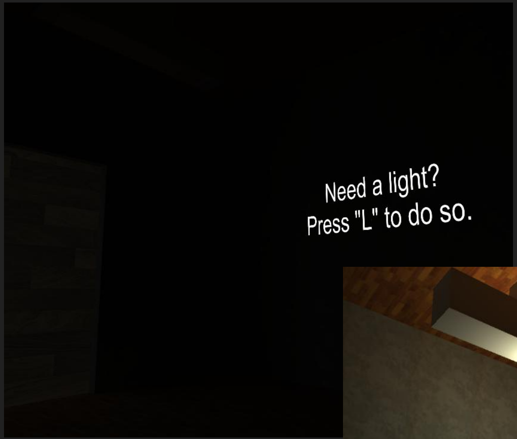
```
SampleScene
  FPSController

Trigger Key Activator (Script)
  Script                   TriggerKeyActivator
  Trigger Keys
    Size                   2
    Element 0              L
    Element 1              O
  Target Objects
    Size                   2
    Element 0
      Control Numb         1
      Toggle Only C        ✓
      This Key Objects
        Size               2
        Element 0          Point light
        Element 1          DarkCanvas
    Element 1
      Control Numb         2
      Toggle Only C        ✓
      This Key Objects
        Size               2
        Element 0          Door
        Element 1          DarkCanvas
```

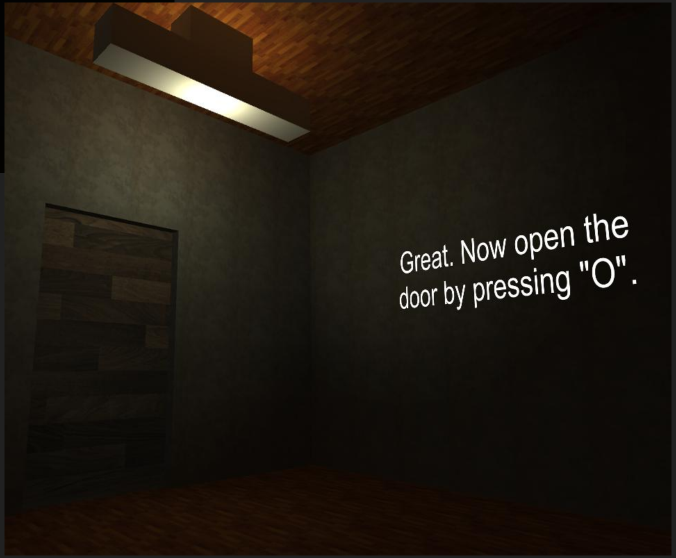
Expand the functionality of the TriggerKeyActivator previously attached to the FPSController.

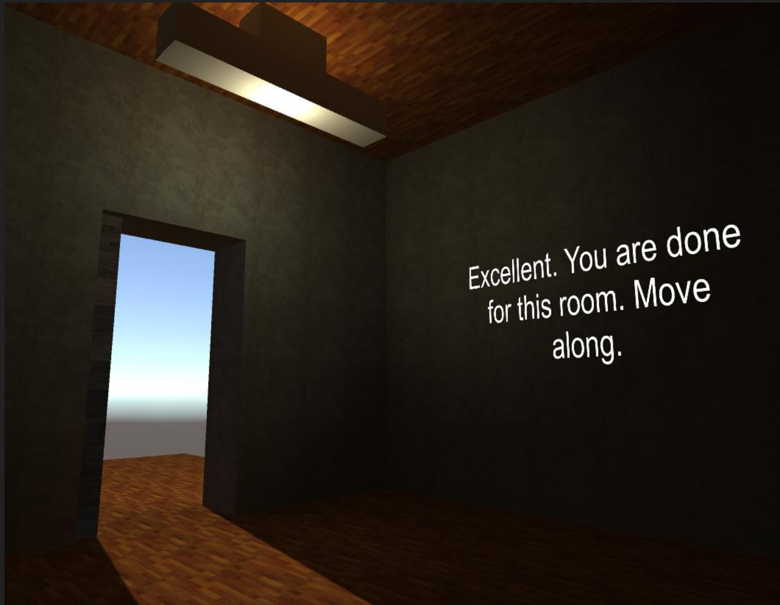
Set TriggerKeys to 2, set the second key to "O".

Similarly, expand TargetObjects to 2, and the size of each to 2 as well (controlNumber 1 & 2). Set the Canvas to both instances, and the door to the second one.

Example 2: writings on the wall





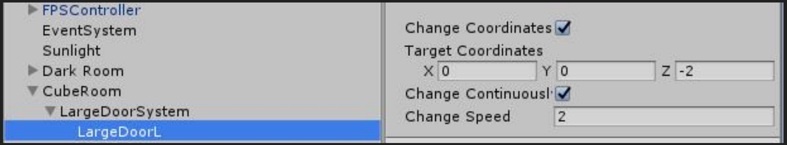


Example 3: entering a room with automatic doors

Let’s move on to another room. With a pair of automatic doors to it!

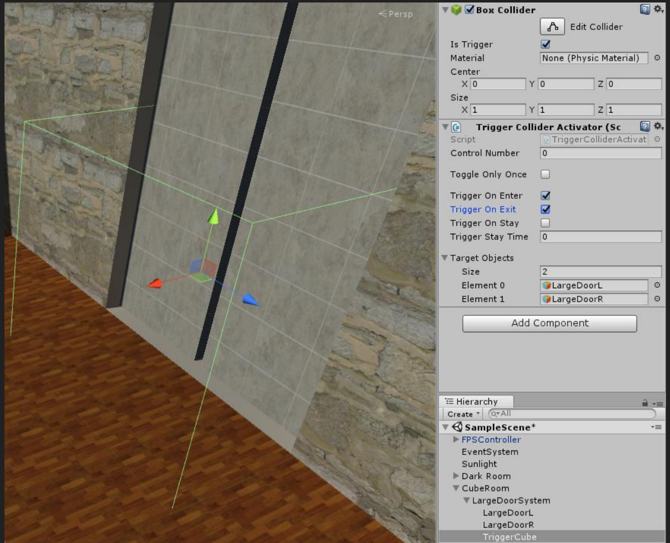
Similarly to the previous example, create a door. Add a ToggleScript to it, and check the “ChangeCoordinates” option. Then, to make the animation smooth, check “ChangeContinuously”, and for the “ChangeSpeed”, input 2.

Duplicate this door and add it side-to-side with the existing one.
Change the direction on the Z coordinate (that is, 2 to -2).
The left door will now slide open to the left, the right one to the right.



- FPSController
- EventSystem
- Sunlight
- Dark Room
- CubeRoom
  - LargeDoorSystem
    - LargeDoorL

Change Coordinates	✓
Target Coordinates	X 0 Y 0 Z -2
Change Continuousl	✓
Change Speed	2



Box Collider — Edit Collider

Is Trigger	✓
Material	None (Physic Material)
Center	X 0 Y 0 Z 0
Size	X 1 Y 1 Z 1

Trigger Collider Activator (Sc

Script	TriggerColliderActivat
Control Number	0
Toggle Only Once	☐
Trigger On Enter	✓
Trigger On Exit	✓
Trigger On Stay	☐
Trigger Stay Time	0
Target Objects	
Size	2
Element 0	LargeDoorL
Element 1	LargeDoorR

- SampleScene*
  - FPSController
  - EventSystem
  - Sunlight
  - Dark Room
  - CubeRoom
    - LargeDoorSystem
      - LargeDoorL
      - LargeDoorR
      - TriggerCube

We also need a TriggerColliderActivator. Let’s create one now. Create a cube and scale it so that it protrudes about five meters in front & to back of the door. Remove the MeshRenderer (not needed) and check the Trigger option on the collider.

Now add a TriggerColliderActivator to it.
Add both the left and the right door to its TargetObjects.
Check both the “TriggerOnEnter” and “TriggerOnExit” options.

Now, once the user moves within/outside of the vicinity of the automatic door, it will open/close, all in a real automatic door fashion.

Example 4: an annoying animated cube to be turned off forever

There’s a pesky animated cube in the middle of the room, blocking my sunlight. Might as well get rid of it (just because); but the path to the switch that disables it is blocked. How to proceed?

Let’s have a look at the cube. It has a ToggleScript with “ToggleSelfOnStart” option enabled. That, and a pink light to it!
Other pesky options: animated rotation and scaling of frivolous intensity.

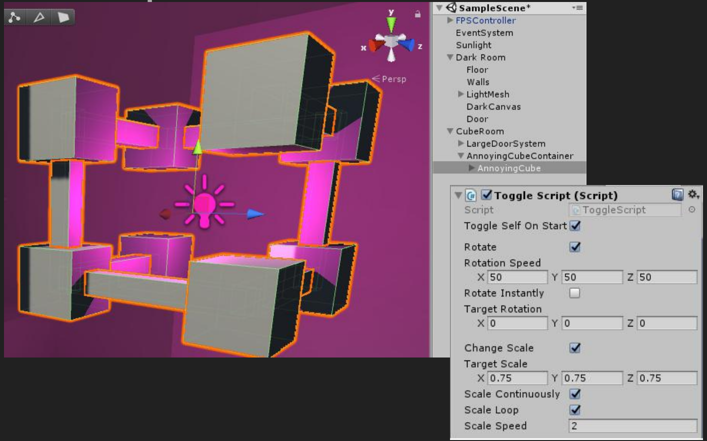
Let’s enable “HiddenOnToggle” on its parent container, so that the cube disappears on toggle.

There’s a button that triggers the cube on the other side of the room. Neat. The only problem is that it is so far up it is not accessible.

But what about these two pads with colliders? Let’s create a one-way teleport out of them to get up there!

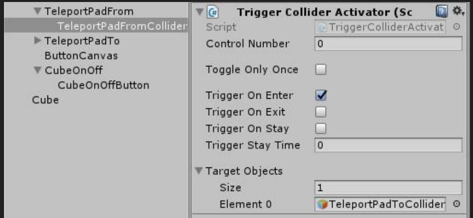
Add TriggerColliderActivator to the first collider and ToggleScript to the other. The trigger should be linking the other collider, and the ToggleScripts should have “IsTeleporter” enabled on it.

This is it! Teleport to the other side of the room, and switch the cube off!



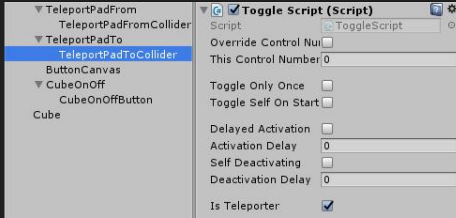
```
SampleScene*
  FPSController
  EventSystem
  Sunlight
  Dark Room
    Floor
    Walls
    LightMesh
    DarkCanvas
    Door
  CubeRoom
    LargeDoorSystem
    AnnoyingCubeContainer
      AnnoyingCube
```

```
Toggle Script (Script)
Script                 ToggleScript
Toggle Self On Start   ✓

Rotate                 ✓
Rotation Speed
   X 50        Y 50        Z 50
Rotate Instantly       ☐
Target Rotation
   X 0         Y 0         Z 0

Change Scale           ✓
Target Scale
   X 0.75      Y 0.75      Z 0.75
Scale Continuously     ✓
Scale Loop             ✓
Scale Speed            2
```



```
TeleportPadFrom
    TeleportPadFromCollider
TeleportPadTo
ButtonCanvas
CubeOnOff
    CubeOnOffButton
Cube
```

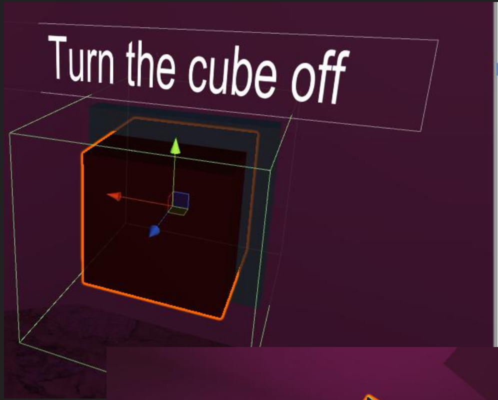
```
Trigger Collider Activator (Sc
Script                 TriggerColliderActivat
Control Number         0

Toggle Only Once       ☐

Trigger On Enter       ✓
Trigger On Exit        ☐
Trigger On Stay        ☐
Trigger Stay Time      0

Target Objects
   Size                1
   Element 0           TeleportPadToCollider
```

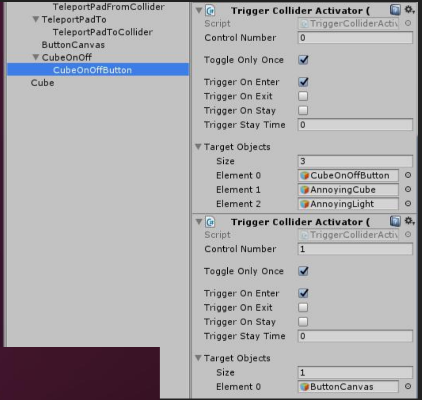


```
TeleportPadFrom
    TeleportPadFromCollider
TeleportPadTo
    TeleportPadToCollider
ButtonCanvas
CubeOnOff
    CubeOnOffButton
Cube
```

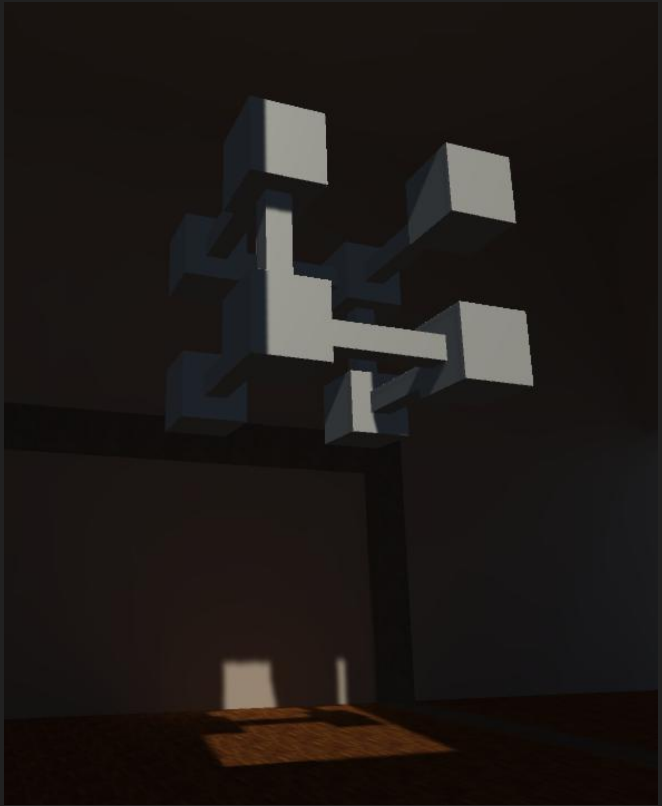
```
Toggle Script (Script)
Script                 ToggleScript
Override Control Nu    ☐
This Control Number    0

Toggle Only Once       ☐
Toggle Self On Start   ☐

Delayed Activation     ☐
Activation Delay       0
Self Deactivating      ☐
Deactivation Delay     0

Is Teleporter          ✓
```

Example 4: an annoying animated cube to be turned off forever




```
      TeleportPadFromCollider
  ▼ TeleportPadTo
      TeleportPadToCollider
    ButtonCanvas
  ▼ CubeOnOff
      CubeOnOffButton
  Cube
```

```
▼ Trigger Collider Activator (
  Script              TriggerColliderActiv
  Control Number      0

  Toggle Only Once    ✓

  Trigger On Enter    ✓
  Trigger On Exit     ☐
  Trigger On Stay     ☐
  Trigger Stay Time   0

▼ Target Objects
    Size              3
    Element 0         CubeOnOffButton
    Element 1         AnnoyingCube
    Element 2         AnnoyingLight
▼ Trigger Collider Activator (
  Script              TriggerColliderActiv
  Control Number      1

  Toggle Only Once    ✓

  Trigger On Enter    ✓
  Trigger On Exit     ☐
  Trigger On Stay     ☐
  Trigger Stay Time   0

▼ Target Objects
    Size              1
    Element 0         ButtonCanvas
```



Example 5: a setup of cycling, animated exhibits

Based on the aforementioned documentation & examples, I’m sure you can figure this one out on your own!

Wrapping up

Due to the sheer amount of possible combinations, there may be bugs (or just some solutions that don’t exactly play well together or as one would fancy).

Then, some functionalities may be lacking.

What to do? Let me know; if viable, I’ll fix it or implement it. If not viable (that is, a request that would break the current programming structure on which ToggleScript rests), I will not fix/implement it -- you will have to live with it.