User Guide

V 1.0

INTRODUCTION

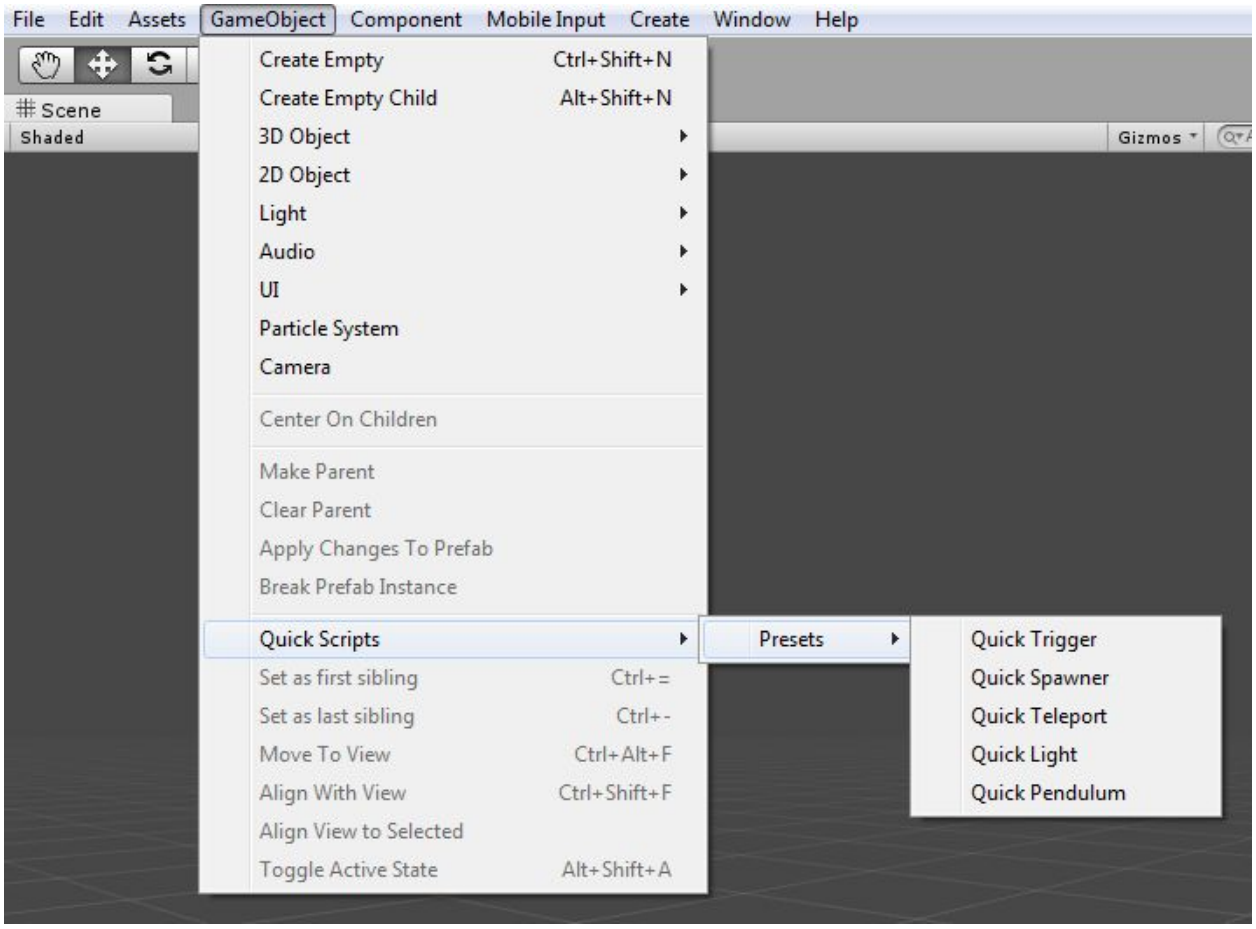
Thank you for purchasing the Quick Scripts asset package for Unity. This documentation will cover each script in the pack, what each feature does, how to use it, and will also have links to short video tutorials on ways you can combine multiple Quick Scripts to create complex features in a matter of seconds.

CONTENTS

Preset Game Objects

To make your life easier, Quick Scripts comes with some ready-made game objects that you can instantly place in your scene.

To find these, go to your toolbar at the top of the screen and navigate to:
GameObject → Quick Scripts → Presets



Click on any one of these and it will immediately appear in your scene, ready to use.

Note:
All scripts in the pack can be accessed through the toolbar: Component → Quick Scripts.

Or searched for in the Inspector:
Select a game object in your scene, then in the Inspector click Add Component → Quick Scripts

Scripts

QUICK DOOR

Overview:

The Quick Door allows users to tell a gameobject to move back and forward or pivot on an axis. If applied to a mesh, it will automatically detect the size of the mesh so that you only need to set a duration and the script will do the moving for you.

It also has the option to play audio clips for the opening and closing of the door.

Quick Tips:

1. If the Quick Door game object also has a Mesh Component, the door will automatically move a distance equal to the size of the mesh. However you cannot change the mesh during runtime. You can override it by specifying a custom distance to move.

2. Pivot doors will need to be a child of another game object. The Door will rotate around the transform position of the parent game object.

See Also

Set up a Quick Trigger to Open a Quick Door

PUBLIC VARIABLES	TYPE	DEFAULT	DESCRIPTION
Door Type	Enum Dropdown	Slide Up	Which direction will the door move when opened?
Move Type	Enum Dropdown	Linear	Linear: The door moves at a constant speed Smooth: The door slows down as it reaches its open/closed position
Custom Move Distance	Bool	Null	If the Door has a Mesh component, it will automatically measure the distance of movement based on that mesh. You can tick this box to set a custom distance.
(*If 'Custom Move Distance' is ticked)* Move Distance	Float	0	Set a custom distance (in units) for how far the door will slide open

(If type is 'Pivot') Pivot Point	Game Object	Null	This is used as the transform which the door pivots around. If using a pivot type door, it's important the game object the Quick Door is on is a child of the pivot game object.
(If type is 'Pivot') Pivot Amount X Axis	Float Slider	0	How far to rotate on the X axis (Constrained to -5 to 5 due to Quaternion mathematics)
(If type is 'Pivot') Pivot Amount Y Axis	Float Slider	0	How far to rotate on the Y axis (Constrained to -5 to 5 due to Quaternion mathematics)
(If type is 'Pivot') Pivot Amount Z Axis	Float Slider	0	How far to rotate on the Z axis (Constrained to -5 to 5 due to Quaternion mathematics)
Duration	Float	0	How long should it take to open? Note: This is not an exact measurement when using Smooth Move Type.
Open Door	Bool	False	When ticked, the door will move towards its open position. When not ticked, it will move towards it closed position. Doors should start closed (this should not be ticked)
Has Audio	Bool	False	If ticked, the door will play audio when opening and closing
(if Has Audio is ticked) Audio Source	Audio Source	Null	The Audio Source to play the audio through. Note: Quick Door will modify its loop settings. Also, check 'Play on Awake' is set to false on the Audio Source
(if Has Audio is ticked) Open Start Audio Clip	Audio Clip	Null	The audio clip to play when the door starts moving. Will loop if 'Extend Sound FX' is ticked.
(if Has Audio is ticked) Open Stop Audio Clip	Audio Clip	Null	The audio clip to play when the door reaches its open position.
(if Has Audio is ticked)	Audio Clip	Null	The audio clip to play when the door starts closing. Will loop if 'Extend Sound FX' is

Close Start Audio Clip			ticked.
(if Has Audio is ticked) Close Stop Audio Clip	Audio Clip	Null	The audio clip to play when the door reaches its closed position
(if Has Audio is ticked) Extend Sound FX	Bool	False	If true, it will loop Open Start Audio and Close Start Audio

QUICK GIZMO

Overview

A Gizmo is only seen in the editor and never at runtime. You can apply this script to any gameobject and choose from a range of gizmo types and colour. They're useful for discerning the location of empty transforms (ie mover nodes or spawn points) or for visualising collider boundaries.

Quick Tips

1. Gizmos are only seen in the editor. They are useful for locating empty game objects in the scene, defining collider boundaries and drawing lines between objects.

2. If the Gizmo is not showing, check that its color's alpha channel is above zero.

PUBLIC VARIABLES	TYPE	DEFAULT	DESCRIPTION
Gizmo Type	Enum Dropdown	Cube	0 = Cube: Draws a cube 1 = Sphere: Draws a sphere 2 = Mesh: Specify a mesh to draw as a Gizmo 3 = Wireframe Cube 4 = Wireframe Sphere 5 = Wireframe Mesh 6 = Line: Draws a line between this game object and a target game object. 7 = Collider: Draws a cube to the extents of the game object's collider boundaries
Gizmo Color	Color	Cyan	The colour of the gizmo. Set A to less than 255 for transparency.

Gizmo Radius	Float	1	How big should the gizmo be? Note: This is ignored if Gizmo Type is Collider
(*If Type is 'Mesh'*) Mesh	Mesh	Null	The Mesh to draw
(*If Type is 'Line'*) Target	Game Object	Null	The game object to draw a line to, from the center of the game object this Gizmo is on.
(*If Type is 'Collider'*) Use Specific Collider	Bool	Null	If the Gizmo is on a game object with more than one Collider, you can tick this to have the option to specify which collider to match the Gizmo to.
(*If Type is 'Collider' and 'Use Specific Collider' is ticked*) Chosen Collider	Collider	Null	The Collider you want to use. This will draw the Gizmo to the size of that Collider's boundaries.
Hide When Not Selected	Bool	False	If ticked, the gizmo will only show in the Scene View window when the game object it is attached to is selected in the Hierarchy

QUICK HOVER

Overview

The Quick Hover can be applied to any game object to make it automatically move back and forward on one or more axis. You can combine it with the Quick Rotate script to make an object bounce up and down and spin.

See Also

Hover, Rotate and Pendulum: Small but Effective Quick Scripts

PUBLIC VARIABLES	TYPE	DEFAULT	DESCRIPTION
Hover X	Float	0	Amount of movement on the X Axis
Hover Y	Float	0	Amount of movement on the Y Axis

Hover Z	Float	0	Amount of movement on the Z Axis
Speed	Float	1	How fast to move
Offset	Float (Range 0 - 3)	0	Offsets the phase of the Sin wave

QUICK LIGHT

Overview

This script can be applied to any Light in your scene. You can set different animations to instantly have dynamic lighting that mimics campfires, candles, televisions, storms, and more.

Quick Tips

1. Adjust the intensity setting on the main Light component to set the maximum intensity for the Light.

2. Tick 'Override Color' to set a custom color animation. The gradient will automatically match the duration of the light animation and will loop.

3. You can change the gradient mode in the gradient window. Experiment with Blend and Fixed to see what works best for you.

4. Lights with the Quick Light component can't be baked because they are animated. Keep them set to Realtime.

PUBLIC VARIABLES	TYPE	DEFAULT	DESCRIPTION
Light Type	Enum Dropdown Enum is called lightAnimation	Standard	Which animation to apply to the light. Options and Values: 0 = standard 1 = pulseSlow 2 = pulseFast 3 = strobe 4 = flicker1 5 = flicker2 6 = flicker3 7 = candle 8 = fire 9 = television 10 = storm

			11 = custom
Speed	Float	2	How long are the animation loops? All animations were made to a speed of 2, so only change it if you need to.
(*If Light Type is Custom*) Light Over Time	Animation Curve	Unset	Only use this if Light Type is Custom. This is where the keyframes are recorded for each animation. If Custom, all keyframes must be set manually.
Override Color	Bool	False	Tick this if you want to change the color of the Light
Color Gradient	Gradient	White	You can manually adjust the gradient over time. It will automatically match the length of the Light's animation and loop when finished

QUICK MOVER

Overview

The Quick Mover is built for the intention of creating moving platforms that can move along a path of nodes with the option to loop, reverse or automatically return to the start. You can also set parameters for if you want the game object to face where it's going with customisable features for turn speed, etc.

This does not only have to be applied to platforms. You can use the Quick Mover on any transform to create all sorts of things.

Quick Tips

1. Make sure the Mover ID matches the one on the mover nodes.

2. If you need to delete or replace nodes, click Rebuild to fix the path.
Note: It may reverse the order of your existing nodes. In which case, just tick 'reverse'.

See Also

Set Up A Moving Platform With Quick Mover

PUBLIC VARIABLES	TYPE	DEFAULT	DESCRIPTION
Mover ID	String	Null	Specify the ID for your Quick Mover. This is to automatically link it to the correct nodes, and to link the nodes together. Note: Please make sure each Quick Mover in your scene has a different Mover ID or else things will break!
Move Speed	Float	0	How fast the Quick Mover should move. (*it could be a 'slow' mover, ahaa)* Note: This can be overwritten by Move Speed on Mover Nodes
Move Type	Enum Dropdown	Linear	What movement type the mover should use. Linear means it travels at a constant, unchanging velocity. Smoothed means it speeds up as it leaves a node and smooths down as it approaches the next one
Is Moving	Bool	True	If true, the Mover will move along the path of nodes. This public bool is useful if you need to tell the Mover to stop from another script.
Reverse Direction	Bool	False	Reverse the direction of the Mover
Loop	Bool	False	If ticked, the Mover will return to the starting node after completing the path and begin moving through the nodes again
Auto Return	Bool	False	If ticked, the Mover will reach the end of the node path and then immediately make it's way back to the start, then continue to do go back and forward like this.
Face Forward	Bool	False	If ticked, the Mover will rotate so it's Z axis is always facing towards the direction it is travelling
Stay Level	Bool	False	If ticked, the Mover can still rotate left and right but never up and down

Rotation Type	Dropdown Enum	By Turn Speed	By Turn Speed = The mover will rotate to face forward at the speed specified in ‘Turn Speed’ Node To Node = The mover will average out it’s rotation over the distance of moving between nodes. Note: This setting will ignore ‘Turn Speed’
Mover Nodes	List (Game Object)	Null	You do not need to interact with this. Mover Nodes with the matching Mover ID will automatically be added to this
Turn Speed	Float	1	How fast the Mover will rotate to face forward, if ‘Face Forward’ is ticked
Node Container	Game Object	Null	This field will fill automatically with the Game Object which all related Mover Nodes are stored under. Do not delete or rename this. You can move this Game Object to somewhere else in the Hierarchy if you want to.
Mover Nodes	List (QS_MoverNode)	-	This will fill with the nodes listed in the hierarchy who are children of the Node Container.
Create Waypoint Nodes	Button	-	Click this to create new waypoint nodes
Rebuild List	Button	-	If you’ve reorganised the order of the nodes or something else has caused the path to be disconnected you can click this button to reset the nodes. Note: It will sometimes reverse the order the nodes were originally entered in, due to the way Unity handles Lists. In this case, you can tick Reverse Direction to fix the problem

QUICK PENDULUM

Overview

The Quick Pendulum can be applied to any game object to make it's transform rotate back and forward on a specified angle and at a specified speed.

Because it rotates the actual transform, it is recommended you make your meshes a child of this transform and offset their positions so they will to rotate around it.

Quick Tips

1. The pendulum will use this game object's transform position as the pivot point and it will override any rotation applied to this transform.

2. Child other game objects to this one for the pendulum effect.

See Also

Hover, Rotate and Pendulum: Small but Effective Quick Scripts

PUBLIC VARIABLES	TYPE	DEFAULT	DESCRIPTION
Angle X	Float	90	How far the pendulum should swing on the X rotation axis
Angle Y	Float	0	How far the pendulum should swing on the Y rotation axis
Angle Z	Float	0	How far the pendulum should swing on the Z rotation axis
Speed	Float	1	How quickly the pendulum will complete its swing
Offset	Float (Range 0 - 3)	0	Offset the phase of the Sin wave used to make the swinging motion

QUICK ROTATE

Overview

Quick Rotate can be applied to any game object to make it spin at a set speed. You can also change the rotation type to create an object that accelerates to a maximum speed and also decelerates.

Alternatively, you can set it to spin faster at an exponential rate but we recommend you don't leave this running forever. It is a feature useful for short effects where the object is likely to be deleted / deactivated soon after.

Quick Tips

1. Smoothed rotation type means this game object will accelerate to it's maximum speed when 'Rotating' is ticked. It will deccelerate when not ticked.

2. Only use Exponential rotation type if you plan on removing the object soon after. It is strongly recommended you do not leave it on exponential forever.

3. Acceleration for Smoothed type = Time it takes to reach maximum speed.
Acceleration for Exponential type = Speed addition per frame. In Exponential mode this number gets hardcoded to be divided by 1000 before the current speed is multiplied by it, so it is recommended to keep this number small, around 0.5 - 5.

See Also

Hover, Rotate and Pendulum: Small but Effective Quick Scripts

PUBLIC VARIABLES	TYPE	DEFAULT	DESCRIPTION
Rotation Type	Enum Dropdown	Linear	Specify the type of rotation to use. Linear = Will rotate at a constant, unchanging speed Smoothed = Will speed up to it's max speed and slow down to zero Exponential = Will continuously speed up Note: Exponential should only be used if the object is will be removed or deactivated, otherwise the float value can go to infinity
Rotating	Bool	True	Is the object currently rotating? Mainly useful for Smoothed Rotation Type, because the Rotator will slow down at the specified acceleration rate

Rotation Speed X	Float	0	How fast to rotate on the X Axis. Can be a negative number
Rotation Speed Y	Float	0	How fast to rotate on the Y Axis. Can be a negative number
Rotation Speed Z	Float	0	How fast to rotate on the Z Axis. Can be a negative number
Acceleration	Float	0	For Smoothed Rotation Type, determines time in second for Rotator to reach its max rotation speed For Exponential Rotation Type, determines rate of speed increase. (Keep it under 5 for best results)

QUICK SPAWNER

Overview

The Quick Spawner should be an empty transform and used to control a number of spawn points. You can use it to instantiate objects at specified points, and can control the spawns by sending them out in waves. It has other options like being able to spawn random objects and to change the amount of time between waves.

A spawn point will be marked as ‘occupied’ if the most recent object it spawned is still within a 0.5 unit radius and until the item is removed it will not spawn a new one. This prevents objects spawning continuously inside each other.

Quick Tips

1. This script allows you to spawn one or more objects at multiple Spawn Points. It is NOT the Spawn Point, but instead the 'central brain' that does the spawning and designates where objects will spawn.

2. You can click 'Create Spawn Point' to quickly make a new Spawn Point.

3. ALL Spawn Points must have a Spawner ID that matches their associated Quick Spawner's ID, otherwise they will not link. By default this will set itself automatically so there should be no need to edit it.

4. If you need to delete a Spawn Point, you can use the 'Refresh Spawn Points' to automatically re-organise the list of Spawn Points.

5. Seconds Between Waves means how many seconds pass until a spawn is called. This means a message is sent to the available Spawn Points to say 'It's time to spawn something!'.

You can add (or subtract) time from this with 'Add Time Between Waves'. This will let you make spawn waves happen faster or slower over time. You can put a negative number in that field to reduce time between waves. It will not go below 0.5 seconds.

6. If you click the dropdown arrow next to 'Objects To Spawn' and change 'Size' 0 to 1 or more, you can drag and drop objects into these slots. These are the objects the Quick Spawner will spawn at the Spawn Points. If you add multiple objects to the list of Objects To Spawn, the Quick Spawner will spawn random selections from that list. To set up chance, just add more or less of the same object.

NOTE: The Quick Spawner will Instantiate objects. Please keep this in mind if building to a platform that runs on a lower processor. Object pooling may be added in the future.

See Also

[How To Correctly Set Up And Use The Quick Spawner]

PUBLIC VARIABLES	TYPE	DEFAULT	DESCRIPTION
Spawner ID	String	Spawner1	Set this to a Unique ID. Used for linking the Quick Spawner with spawn points that also have the same Spawner ID. Note: Make sure no other Quick Spawner has the same Spawner ID or else it will break!
Is Active	Bool	True	If ticked, the Spawner will run as usual. And if not, all spawning will be stopped until it is ticked again.
Random Spawn Point	Bool	False	Determines whether or not objects will be spawned at a randomly selected spawn point from the list of spawn points If NOT ticked, will spawn to all spawn points all the time
Auto Respawn	Bool	True	Respawn objects automatically
Seconds Between Spawn Waves	Float	5	How many seconds to wait until starting the next wave of spawns. Only works if Auto Respawn is True
Start Spawned	Bool	True	Immediately start with a wave of objects

			spawned instead of waiting for the delay to lapse
Objects To Spawn	List (Game Object)	0	Increase the size above zero and fill the fields with any game objects you want to spawn. If more than one, it will choose randomly from the list each time Note: In its current form Quick Spawner will instantiate these objects instead of using an object pool. This may be looked into in a future version
Spawn Points	List (QS_SpawnPoint)	0	The list of spawn points associated with this Quick Spawner. Do not enter them manually, instead use the Auto-Add Button. Requires matching Spawner IDs
Create Spawn Point	Button	-	Create a new Spawn Point automatically linked to this Spawner
Refresh Spawn Points	Button	-	If you need to delete a Spawn Point, press this button to update the list

QUICK TELEPORT

Overview

The Quick Teleport is a trigger box that can send objects with certain tags to a destination. You can set up random destinations and even have particle effects and sounds that play when the teleport is used.

Quick Tips

1. Click 'Create Destination Node' to make a point to teleport to.
2. Add more than one destination node to create a teleporter that sends to random destinations.
3. Remember to fill out the list of interactable tags for what game objects can be teleported.
4. Teleported objects will face the same way as the node they teleport to. The direction the node is facing is indicated by the small line protruding from it.
5. If Cool Down Time is set to 0, the teleport will only trigger once. Default is 0.1.
6. Create an Audio Source and assign it to the Entry and Exit Audio Source fields if you want to play audio when the teleport is used.
7. Tick 'Move Exit Audio To Destination' to play the Exit Audio where at the Destination.

NOTE: This will move the game object that contains the Audio Source component.

See Also

Make A Quick Teleporter With Sound And Particle Effects

PUBLIC VARIABLES	TYPE	DEFAULT	DESCRIPTION
Is Active	Bool	True	Is the teleport active? As in, will it teleport an object when the object enters the trigger?
Cool Down Time	Float	0.1	How many seconds to wait before the teleport can be triggered again
Maintain Inertia	Bool	True	If the object passing into the Teleport is travelling at a velocity above zero, this setting will maintain that velocity when it is teleported. Untick this to set the teleported object's velocity back to zero when teleported.
Teleport Parents of Colliding Object	Bool	True	If ticked, and if a Collider which is a child of other objects is teleported, this will teleport all parent objects too.
Destination Nodes	List (Game Object)	0	A list of game objects that will be randomly selected from as the teleport destination each time an object is teleported
Random Destination	Bool	False	Set to True if you want to teleport to a randomly selected game object from the Random Destinations list
Entry Particle FX	Particle System	Null	A particle effect to play at the position of entry when an object is teleported
Play Entry FX At Point Of Contact	Bool	True	The particle effect will play at the exact location a game object collided with the Teleporter's collider box
Exit Particle FX	Particle System	Null	A particle effect to play at the destination when an object is teleported
Play Exit FX At Destination	Bool	True	The Exit Particle Effect will play at the transform position of the destination node the colliding object was teleported to
Entry Sound Fx	Audio Clip	Null	An audio clip to play when an object enters the teleporter

Exit Sound Fx	Audio Clip	Null	An audio clip to play when an object exits the teleporter
Entry Audio Source	Audio Source	Null	The Audio Source that will play the Entry Sound Fx
Exit Audio Source	Audio Source	Null	The Audio Source that will play the Exit Sound Fx
Move Exit Audio To Destination	Bool	False	If true, moves the game object the Exit Audio Source is attached to, to the position of the teleport destination when a teleport occurs. (Simply put, it means the exit sound can be played wherever the object is teleported to) NOTE: Ensure the Audio Source is not attached to a Game Object that should not be moved. This setting will change that Game Object's transform position!
Interactable Tags	List (String)	0	A list of all tags that can interact with the Quick Teleporter's collider and be teleported Note: All objects entering the teleport must have a Rigibody. This is required by Unity
Edit Tags	Bool	False	If ticked, opens the tag editing options
Interactable Tag	Dropdown	Null	Select a current tag from this drop down list
Add Tag	Button	-	Press this to add the tag in 'Interactable Tag' to the list of 'Interactable Tags'
Remove Tag	Button	-	Press this to remove the last entry of the tag specified in 'Interactable Tag' from the list of 'Interactable Tags'
Remove Last	Button	-	Press this to remove the last entry in the list of 'Interactable Tags'

QUICK TRIGGER

Overview

The Quick Trigger is a script applied to a collider which is set to ‘trigger’. You can specify events to happen when a game object (with a certain tag) enters, stays within, or exits the trigger.

Other settings include being able to specify input requirements to make the trigger activate. For example, the player might have to stand inside the trigger and press the ‘interact’ button, or the ‘e’ key, or you can even require input combinations ie Left Click + Right Click.

Quick Tips

1.You must add tags to the list of interactable tags to define what game objects will activate this trigger. To do this, scroll down and tick 'Edit Tags'. Then, select a tag from the dropdown field. Then press Add Tag. It will be added to the list. To view the list, look for the dropdown arrow next to 'Interactable Tags'. Any tags you added should appear in this list. When you're done adding or removing tags, it's recommended you untick 'Edit Tags'.

2. Any object interacting with a trigger must have a Rigidbody, this is required by Unity.

3. To set up events to happen when something enters, stays within, or exits the Trigger follow these steps:
- Click the + sign underneath TriggerEnter, TriggerStay or TriggerExit to add an event.
- An empty object field will appear. You can drag and drop any object from the Hierarchy window into this field.
- You can now use the dropdown menu for this object to access any Component on that object and edit most public variables or call public functions within those Components. These edits or calls will happen when the tigger is entered, stayed within or exited depending how you set it up.

4. To set up input requirements follow these examples:
(If you want the player to press a button etc to make the event happen. This feature only applies to Trigger Stay events)
- Key Names are written in lowercase. ie 'E' Key would be e. Some are abbreviated ie 'left ctrl'. Consult Unity's Online API for all key names.
- Axis input names are taken from Unity's Input Manager. Make sure they match case, for example 'Fire1'. To find Axis names go to Edit/Project Settings/Input.
- Mouse input: \n0 = Left Click\n1= Right Click\n2 = Middle Click

See Also

Set Up A Quick Trigger To Open and Close Quick Doors

PUBLIC VARIABLES	TYPE	DEFAULT	DESCRIPTION
Trigger Enter	Unity Event	Empty	What to run when an object enters the trigger collider
Trigger Stay	Unity Event	Empty	What to run while an object is inside the trigger collider Note: An object must pass into the trigger to be recognised as staying inside it. This means you can't teleport or instantiate into a trigger and call Trigger Stay commands
Trigger Exit	Unity Event	Empty	What to run when an object leaves the trigger collider
Trigger Once	Bool	False	Only activate the trigger once? Will deactivate the game object Quick Trigger is attached to after the trigger has fired
Input Type	Dropdown Enum	On Input Down	Specify what the player should be doing to register the input has been received. On Input Down = The frame the user pressed the input While Input Down = All frames in which the user is pressing the input On Input Release = The frame in which the user stopped pressing the input
Input Combination Mode	Dropdown Enum	OR	OR = Only one of any specified input requirements are needed to activate Trigger Stay() AND = All of the input requirements are needed, at the same time, to activate Trigger Stay()
Require Key Input	Bool	False	If ticked, the player needs to press a key while inside the Trigger to activate all Events specified in TriggerStay()
Input Key Name	String	Null	The name of the key. Ie 'e' or 'left ctrl' For all key names, see here: https://docs.unity3d.com/ScriptReference/KeyCode.html
Require Axis Input	Bool	False	If ticked, the player needs to press a button specified from the Axes (Edit/Project

			Settings/Input) while inside the Trigger to activate all Events specified in TriggerStay()
Input Axis Name	String	Null	The name of the Axis as specified in the Axes list in Project Settings/ Input. Ie ‘Fire1’
Require Mouse Input	Bool	False	If ticked, the player needs to press a mouse button while inside the Trigger to activate all Events specified in TriggerStay()
Input Mouse Button	Int	0	0 = Left Click 1 = Right Click 2 = Middle Click
Interactable Tags	List (String)	0	A list of tags determining what objects can activate the trigger when they enter the collider Note: Objects will require a Rigidbody
Edit Tags	Bool	False	If ticked, opens the tag editing options
Interactable Tag	Dropdown	Null	Select a current tag from this drop down list
Add Tag	Button	-	Press this to add the tag in ‘Interactable Tag’ to the list of ‘Interactable Tags’
Remove Tag	Button	-	Press this to remove the last entry of the tag specified in ‘Interactable Tag’ from the list of ‘Interactable Tags’
Remove Last	Button	-	Press this to remove the last entry in the list of ‘Interactable Tags’

Video Tutorials

Set Up A Quick Trigger To Open and Close Quick Doors

Set Up A Moving Platform With Quick Mover

Make A Quick Teleporter With Sound And Particle Effects

How To Correctly Set Up And Use The Quick Spawner

Hover, Rotate and Pendulum: Small but Effective Quick Scripts

Contact

Questions, Bug Reports or other requests can be sent to info@wanderlightgames.com

You can also contact Jack Wilson on Twitter: @t_jackwilson