User Guide

V 2.0

Last edit August 1st, 2021

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.

CONTENT

Here is a brief description of every feature included in Quick Scripts: Level Designer's Kit.
* Quick Scripts Lite: Level Designer's Kit only contains these asterisked items.

Quick Door	Add sliding or pivoting doors to your scene, with optional audio and the ability to fire Events on open and close.
Quick Follow *	Tell your game object to follow a target transform. You can specify if it should stick to it, lag behind it, and also set a minimum distance.
Quick Gizmo *	A useful component that can be added to any game object allowing you to draw a gizmo on it in the Scene View window in Editor. This can be a cube, a sphere, a mesh, or wireframe, or allow you to see the bounds of a box collider on the same object. Will not show in-game.
Quick Hover *	Allow your game object to hover on any axis using a sine wave.
Quick Light *	Easily add preset animations to Spot or Directional Lights such as campfire, storm, slow pulse, flicker, and more, or create your own animations using an Animation Curve. Also provides options to add colour changes during the animation.
Quick LookAt	Apply this to a game object to tell it to face toward a target transform. You can lock X, Y or Z axis rotations and tell it to face away from the target instead.
Quick Mover	Apply this to any game object and assign it a Quick Track to move along. You can control its speed, whether it will loop or ping pong along the track, and many other options.
Quick Pendulum	Put this on any object and it will rotate back and forth on any or all axes.
Quick Relay	A very useful component that fires all its Events after being called a specified number of times. You can add a delay to this which helps set up timed sequences.

Quick Rotate * Apply this to any object to add rotation on any axis. You can tell it to rotate constantly, speed up/ slow down, or rotate exponentially faster. You can also apply a curve to each axis if you want rotation speeds to fluctuate.

Quick Scale * Apply this to any object to make it scale up and down on a sine wave. You control the amount of scaling per axis.

Quick SetParent This is a trigger collider which will 'capture' objects with Rigidbodies and a specified tag and make them a child of the SetParent. It will return the child to it's previous parent when the child leaves the trigger collider. This is a must-have for setting up moving platforms or elevators.

Quick Shake Apply this to any object in your scene to shake it at any intensity and on any axis. You can also Ease in/ Ease out the shaking if you want a short-burst effect.

Quick Slide * Adding this to an object allows you to give it a start position and rotation, and an end position and rotation, and how long it takes to move between the two. A useful component for when you need a simple solution quickly.

Quick Spawner A 'brain' that spawns assigned prefabs at any of its destination nodes. There is added security where it will not spawn a prefab at a node until the previous one is moved away from the node or deleted.

Quick Teleport A trigger box that, when a Rigidbody with a specified tag enters it, will teleport the game object of that Rigidbody to an assigned or randomised destination. You can also easily add your own particle effect to play at both the origin and destination of the teleport.

Quick Track A highly flexible tool that allows you to set up a path for Quick Movers to travel along with options to override the Mover's speed, ease in/ease out of nodes, wait or stop at nodes, and more.

You can create Simple or Complex paths depending on your needs. Simple paths are direct node-to-node, whereas Complex paths use a spline for curves between nodes.

The Quick Track is by far the most complex tool of this kit so please watch tutorials or inspect its uses in the Demo Scene for instructions.

Quick Trigger * A highly useful tool, the Quick Trigger uses a trigger Box Collider to fire Events when a Rigidbody with a specified tag moves in or out of the collider. You can control Events for On Enter, On Stay and On Exit. You can add required input for On Stay, allowing you to create situations where the player must stand in the trigger and press a certain key/button combination to fire the Events.

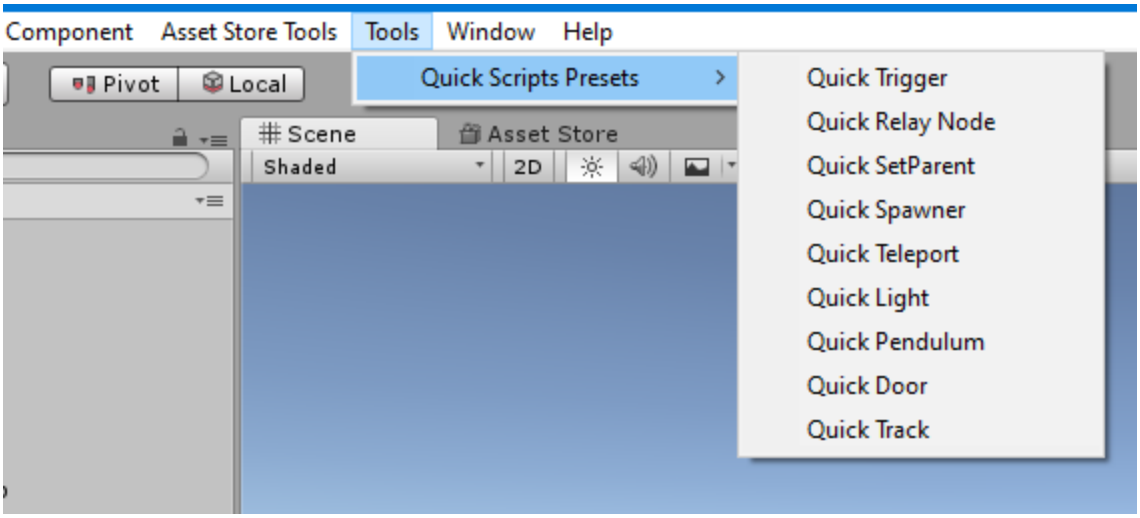
Presets and Prefabs

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

Presets

Presets are new game objects you can add to your scene using the Tools menu in the toolbar. You can access all Quick Scripts features this way.

To find these, go to your toolbar at the top of the screen and navigate to:
Tools → 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

Prefabs

There are some ready-made prefabs in your Project files, under Assets/Quick Scripts/Prefabs. Here you will find some useful prefabs that have had a bit more work put into them, they are more than just preset objects. For example, a Two-Way Door with Trigger Boxes already set up.

Features

QUICK DOOR

Overview:

The Quick Door is a very versatile component that allows users to make any game object act like a door. It can be used to make sliding doors (up/down, forward/back, left/right) or doors that rotate on a pivot.

If the game object being used for the Door has a mesh, the bounds of the mesh can be used to automatically determine how far the object needs to move (This is optional)

It also has extra functionality like playing audio clips or firing events on the opening and closing of the door, and time delays for cooldowns or automatic closing.

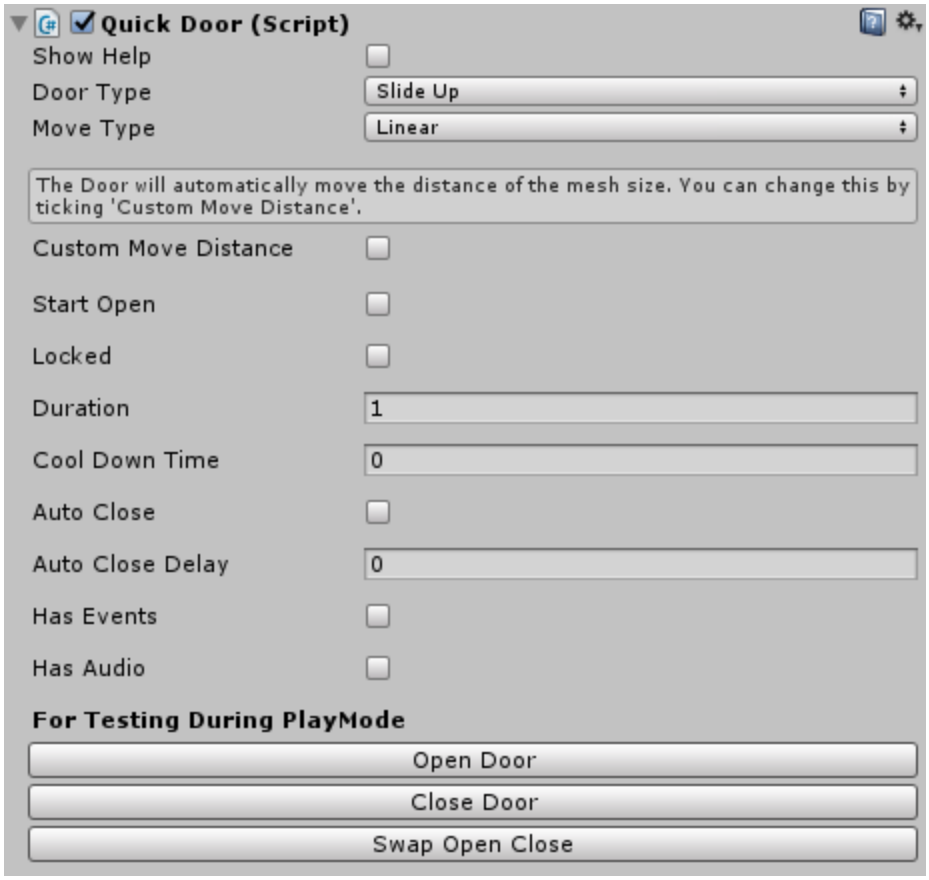
Quick Tips:

1. Pivot doors will need to be a child of another game object. The Door will then use the parent's transform position to be the pivot point it rotates around.
2. If you want a door that can open in/out from both sides, use the Two-Way Y Axis Door Type.
3. 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. Please note this distance won't update if you change the mesh during runtime. You can however override it by specifying a custom distance to move.

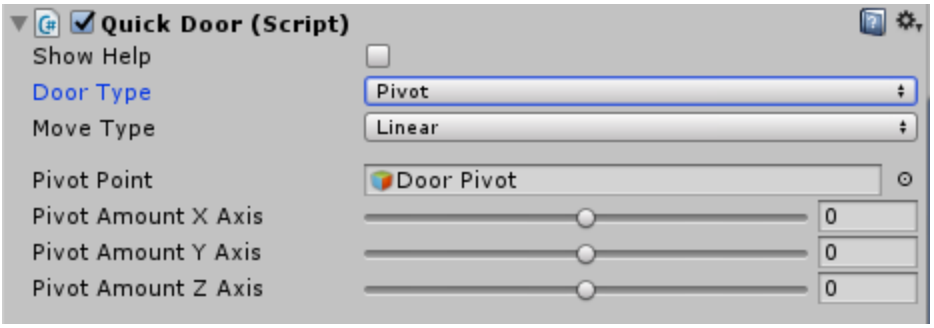
See Also

TODO Remake Video: Set up a Quick Trigger to Open a Quick Door

Inspector Breakdown

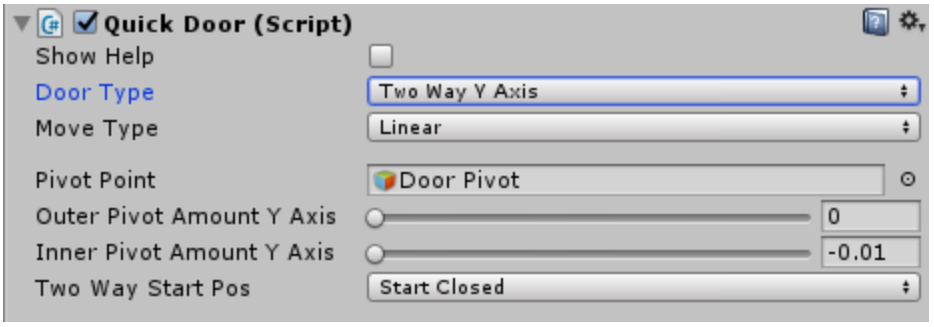


INSPECTOR FIELD	TYPE	DEFAULT	DESCRIPTION
Door Type	Enum Dropdown	Slide Up	Which direction will the door move when opened?
Move Type	Enum Dropdown	Linear	Linear: The door moves at a constant speed Smooth: The door slows down as it reaches its open/closed position
Custom Move Distance	Bool	Null	If the 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 Door Type = Pivot

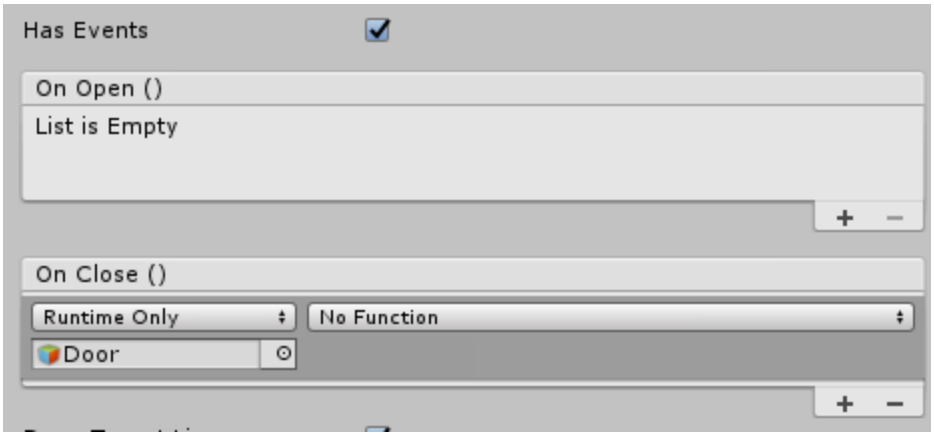
Pivot Point	Game Object	Null	This is used as the transform which the door pivots around. If using a pivot type door, it's important the game object the Quick Door is on is a child of the pivot game object.
Pivot Amount X Axis	Float Slider	0	How far to rotate on the X axis (Constrained to -179.9 to 179.9 due to Quaternion mathematics)
Pivot Amount Y Axis	Float Slider	0	How far to rotate on the Y axis (Constrained to -179.9 to 179.9 due to Quaternion mathematics)
Pivot Amount Z Axis	Float Slider	0	How far to rotate on the Z axis (Constrained to -179.9 to 179.9 due to Quaternion mathematics)



If Door Type = Two Way Y Axis

Outer Pivot Amount	Float Slider	0	How far to rotate on the Y Axis when opening outwards (Constrained to -179.9 to 179.9 due to Quaternion mathematics)
Inner Pivot Amount	Float Slider	-0.01	How far to rotate on the Y Axis when opening inwards

			(Constrained to -179.9 to 179.9 due to Quaternion mathematics)
Two Way Start Pos	Enum	Start Closed	Which position do you want the door to be in when the game starts?
Regular Properties			
Start Open	Bool	False	When ticked, the door will be in it's open position when the game starts
Duration	Float	0	How long (in seconds) should it take to finish it's opening/closing movement?
Cool Down Time	Float	0	How long (in seconds) should the door be unable to change from opening/closing? Ie if the Cool Down Time is set to 5 seconds, and the door takes 3 seconds to open, it will be unable to close until 2 seconds after reaching its open position (and the same for closing). Or, if the Cool Down Time is set to 5 seconds, and the door takes 10 seconds to open, after 5 seconds the opening/ closing movement can be reversed.
Auto Close	Bool	False	If ticked, the door will automatically close after reaching it's open position
Auto Close Delay	Float	0	If Auto Close is ticked, then how long (in seconds) should the door wait after reaching it's open position before it automatically closes?



If 'Has Events' is ticked

Has Events	Bool	False	If ticked, the user can create Unity Events to run when the door reaches its open position (On Open) and when the door reaches its closed position (On Close)
On Open()	Unity Event		Events to fire when the Door reaches its open position
On Close()	Unity Event		Events to fire when the Door reaches its closed position
Draw Target Lines	Bool	True	If ticked, the Scene View will show gizmo lines pointing to each game object in the Door's OnOpen and OnClose Events

If 'Has Audio' is ticked

Has Audio	Bool	False	If ticked, the door will play audio when opening and closing
Audio Source	Audio Source	Null	The Audio Source to play the audio through. Note: Quick Door will modify the loop settings of the linked Audio Source. Also, check 'Play on Awake' is set to false on the

			Audio Source
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.
Open Stop Audio Clip	Audio Clip	Null	The audio clip to play when the door reaches its open position.
Close Start Audio Clip	Audio Clip	Null	The audio clip to play when the door starts closing. Will loop if ‘Extend Sound FX’ is ticked.
Close Stop Audio Clip	Audio Clip	Null	The audio clip to play when the door reaches its closed position
Extend Sound FX	Bool	False	If ticked, it will loop Open Start Audio and Close Start Audio

PUBLIC FUNCTIONS	PARAMETERS	DESCRIPTION
SwapOpenClose		Change if the door is Open or Closed.
SwapOpenCloseImmediately		
SetDuration	Float	Set how quickly the door opens/ closes
SetLocked	Bool	
SetHasAudio	Bool	Set if the door will play audio when opening/closing
SetCustomMoveDistance	Float	Set how far the door moves when opening
OpenDoor		Open the door. Does not work for Two Way Doors
CloseDoor		Close the door.

OpenTwoWayIn		Open the Two Way Door on its inner swing angle
OpenTwoWayOut		Open the Two Way Door on its outer swing angle

QUICK FOLLOW

Overview

This component can be attached to any game object in the scene, and it will constantly move towards a target game object. You can specify some behaviours such as making it ‘stick to’ its target (meaning it will copy the target’s position every frame), or stop following if the target is within a set range, or slowly move towards the target over time.

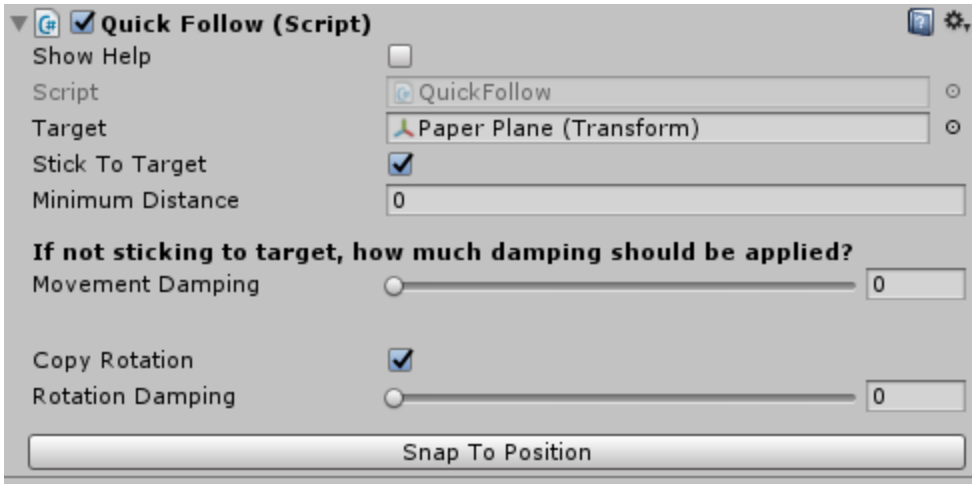
Quick Follow also allows the game object to copy the Target’s rotation for a nicer look when following along!

Quick Tips

1. Drag and drop any game object in the Scene into the Target field. This game object will now follow the Target.
2. If ‘Stick to Target’ is true, this game object will copy the world position of the Target game object every frame.
3. If you want this game object to slowly follow the Target, set a dampening speed. If the Target is also moving slowly you will need a very high damping value, for example 0.99.
4. Experiment with the rotation options if you want the following to look more natural!

See Also

Inspector Breakdown



INSPECTOR FIELD	TYPE	DEFAULT	DESCRIPTION
Target	Transform		The Transform of the game object in the Scene that this game object will follow
Stick To Target	Bool		If ticked, this game object will ignore all other customisation options in the Quick Follow component and instead apply itself to the exact same position as the Target on every frame
Minimum Distance	Float	0	If > 0, this game object will not move if the Target is closer than the minimum distance
Movement Damping	Float	0	If > 0, this game object will lag as it tries to reach the Target's position If 1, it will not move at all
Copy Rotation	Bool	True	If ticked, this object will copy the rotation of the Target game object
Rotation Damping	Float	0	If > 0, this game object will lag as it tries to match the Target's rotation If 1, it will not move at all
Snap To Position	Button		[Editor Only] If pressed, this game object will snap to the Target position

QUICK GIZMO

Overview

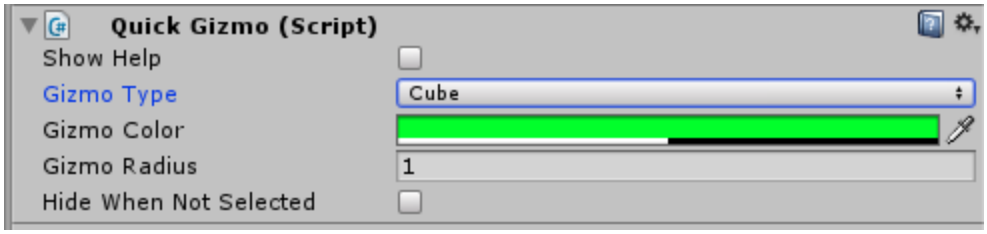
A Gizmo is only seen in the editor and never at runtime. You can apply this script to any game object and choose from a range of gizmo types and colour. They're useful for discerning the location of empty transforms (ie mover nodes or spawn points) or for visualising collider boundaries.

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.

Inspector Breakdown



INSPECTOR FIELD	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

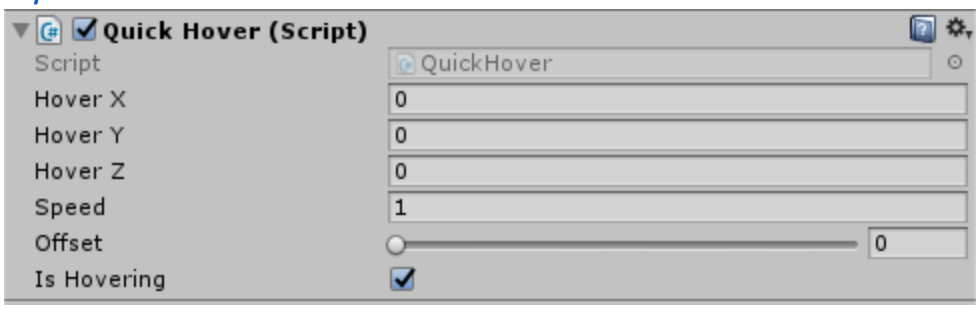
Overview

The Quick Hover can be applied to any game object to make it automatically move back and forward on one or more axes (uses a sin wave). 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

Inspector Breakdown



INSPECTOR FIELD	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

PUBLIC FUNCTIONS	PARAMETERS	DESCRIPTION
SetHovering	Bool	Set if the object is hovering or not

SetHoverX	Float	How far to move on X axis
SetHoverY	Float	How far to move on Y axis
SetHoverZ	Float	How far to move on Z axis
SetSpeed	Float	How fast to move
SetOffset	Float	Set the position 0 - 3 for the offset on the sin wave the hover uses for its movement

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

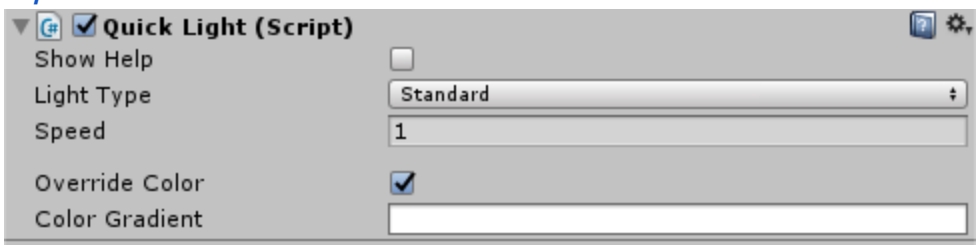
1. Adjust the intensity setting on the main Light component to set the maximum intensity for the Light.

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.

Inspector Breakdown



INSPECTOR FIELD	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	1	How long are the animation loops? All animations were made to a speed of 1, so change it if you want a faster or slower animation
(*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

PUBLIC FUNCTIONS	PARAMETERS	DESCRIPTION
SetLightType	Int (0-11)	Set the Light Type (see above)
SetSpeed	Float	Set how fast the light animation runs
SetOverrideColorBool	Bool	Sets if the lights colour will update with the override colour gradient during its animation

QUICK LOOKAT

Overview

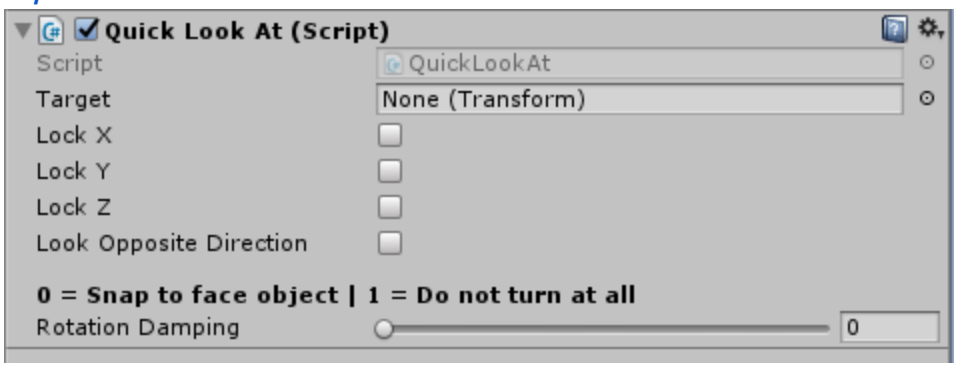
Add this to any game object in your scene to make it face towards or away from a set Target game object. This is useful for making elements like compasses, quest pointers, or making an NPC turn its head towards something.

Quick Tips

1. Add any game object in the Scene into the Target field and this game object will automatically rotate to face towards it. (Z axis forward)
2. You can slow the movement by modifying TODO: Parameter X
3. Tick the 'Face Away' box to make your game object look in the opposite direction to where the Target is

See Also

Inspector Breakdown



INSPECTOR FIELD	TYPE	DEFAULT	DESCRIPTION
Target			The Transform of the game object in the Scene that this game object will rotate to face
Lock X			If ticked, will not allow a change of rotation on X axis
Lock Y			If ticked, will not allow a change of rotation on Y axis
Lock Z			If ticked, will not allow a change of rotation on Z axis

Look Opposite Direction			If ticked, will look away from the Target game object
Rotation Damping			If > 0, this game object will lag as it tries to match the Target's rotation If 1, it will not move at all

QUICK MOVER

Overview

The Quick Mover is built for the intention of creating moving game objects that can move along a Quick Track with the option to loop or ping pong (which means to go forwards and then backwards along the track).

PLEASE NOTE: There is no functionality for carriages. The Quick Mover can only be applied to individual game objects, you cannot chain multiple Quick Movers together.

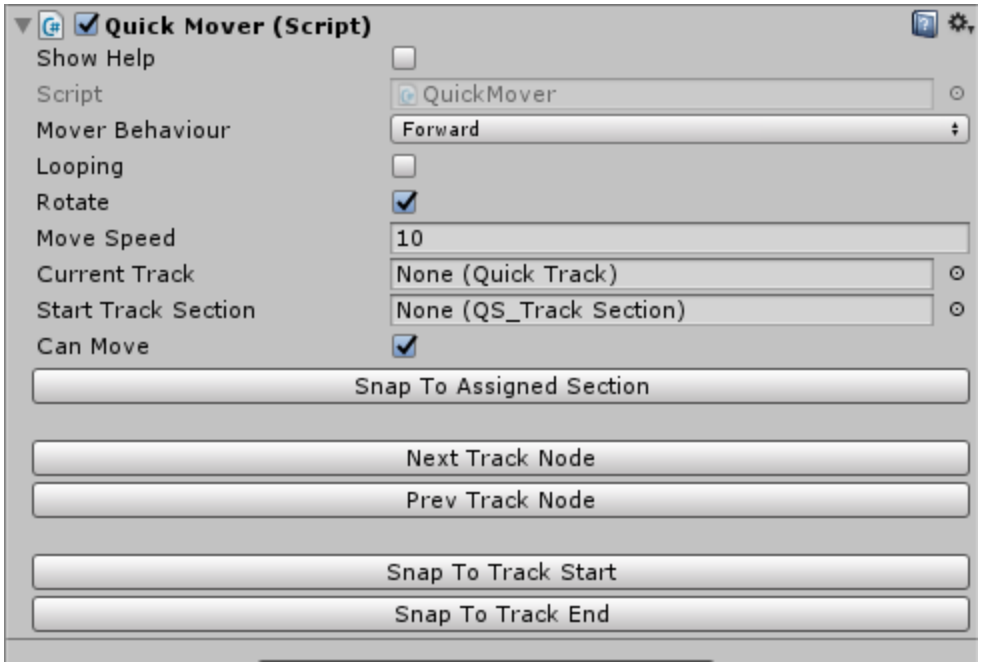
Quick Tips

1. Drag and drop any Quick Track from the Scene into the Quick Track field below, then press the ‘Go to Start Node’ button to move this Quick Mover into position.
2. If you want this Quick Mover to start at a different position along the track, you can use the Next Node or Previous Node buttons to move it.
3. Make sure the ‘Can Move’ box is ticked otherwise your Quick Mover will stay still.
4. There are a lot of useful Public Functions you can run on the Quick Mover from Unity Events (Quick Trigger, Relays, etc) so be sure to have a look at those in the documentation or in the script file itself.

See Also

TO DO: Video tutorial

Inspector Breakdown



INSPECTOR FIELD	TYPE	DEFAULT	DESCRIPTION
Mover Behaviour	Enum	Forward	Sets the direction the mover will go along the track to its end. Note: Ping Pong will always move Forward first.
Looping	Bool	False	If set to true, will loop on the track when it reaches the last node
Rotate	Bool	True	If true, will apply the Quick Track rotation to the Quick Mover game object
Move Speed	Float	10	How fast to move along the track Note: High values will create inaccurate movement
Current Track	Quick Track		The track the Quick Mover will move along
Start Track Section	QS_Track Section		The node on the Quick Track the Mover will start at

Can Move	Bool	True	If true, the Quick Mover will move. This bool is used by some features on the Quick Track such as wait nodes. Useful for linking with Quick Triggers to control when a Quick Mover will start moving
Buttons			
Snap To Assigned Section	Button		If a QS_TrackSection is assigned in the 'Start Track Section' field, this will move the Quick Mover to its position
Next Track Node	Button		Change the Start Track Section to the next section along the Quick Track
Prev Track Node	Button		Change the Start Track Section to the previous section along the Quick Track
Snap To Track Start	Button		Move the Quick Mover to the Start of the Track
Snap To Track End	Button		Move the Quick Mover to the End of the Track

PUBLIC FUNCTIONS	PARAMETERS	DESCRIPTION
StartMoving		Sets Can Move to false
StopMoving		Sets Can Move to true
DeccelerateToStop_InSeconds	Float 'time'	Slows the Mover from its current speed down to zero over the amount of seconds specified
AccelerateToMaxSpeed_InSeconds	Float 'time'	Speeds the Mover from its current speed up to its set Move Speed over amount of seconds specified
SetMoveSpeed	Float 'speed'	Immediately changes the Move Speed

SetLooping	Bool ‘b’	Sets Looping to true or false as specified
MoveForward		Sets the Mover Behaviour to Forward
MoveReverse		Sets the Mover Behaviour to Reverse
MovePingPong		Sets the Mover Behaviour to Ping Pong

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 (uses a sin wave).

Because Quick Pendulum rotates the attached transform, it is recommended you make your meshes a child of this transform and offset their positions so you can see the pendulum effect... *in full swing*.

Quick Tips

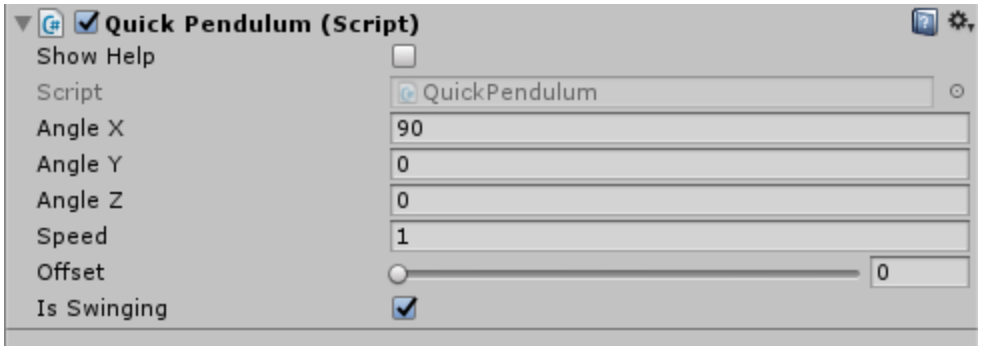
1. The pendulum will use this game object's transform position as the pivot point and it will override any rotation applied to this transform.

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

See Also

Hover, Rotate and Pendulum: Small but Effective Quick Scripts

Inspector Breakdown



INSPECTOR FIELD	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

PUBLIC FUNCTIONS	PARAMETERS	DESCRIPTION
AccelerateOverTime	Float	The value entered defines time in seconds for the pendulum to reach max angle **Recommended** to use if you need the pendulum to reach its max swing from no movement
DecelerateOverTime	Float	The value entered defines time in seconds for the pendulum to reach 0 swing angle on all axes **Recommended** to use if you need the pendulum to slow down to a stop.
SetAngleX	Float	Immediately change the swing angle x
SetAngleY	Float	Immediately change the swing angle y
SetAngleZ	Float	Immediately change the swing angle z
SetSpeed	Float	Immediately change the speed. Not recommended to use when the pendulum is swinging. Stop the pendulum first, change speed, then resume.
SetSwinging	Bool	Start or stop the pendulum swinging
SetOffset	Float	Value 0 - 3 for offsetting the pendulum's position on the sin wave use for the swing

QUICK RELAY

Overview

The Quick Relay is useful for firing Unity Events when called from another game object / script. A Quick Relay can also be given a set number of times to be called before it runs, and a delay before it runs.

'Run' = When the Quick Relay fires all of it's Events.

'Call' = When the Quick Relay is called by another object in the scene, as a request to Run. They do this by calling its Call() function. The Quick Relay can have any number of calls required before it runs. 1 call is the minimum = will Run as soon as it is called.

Quick Tips

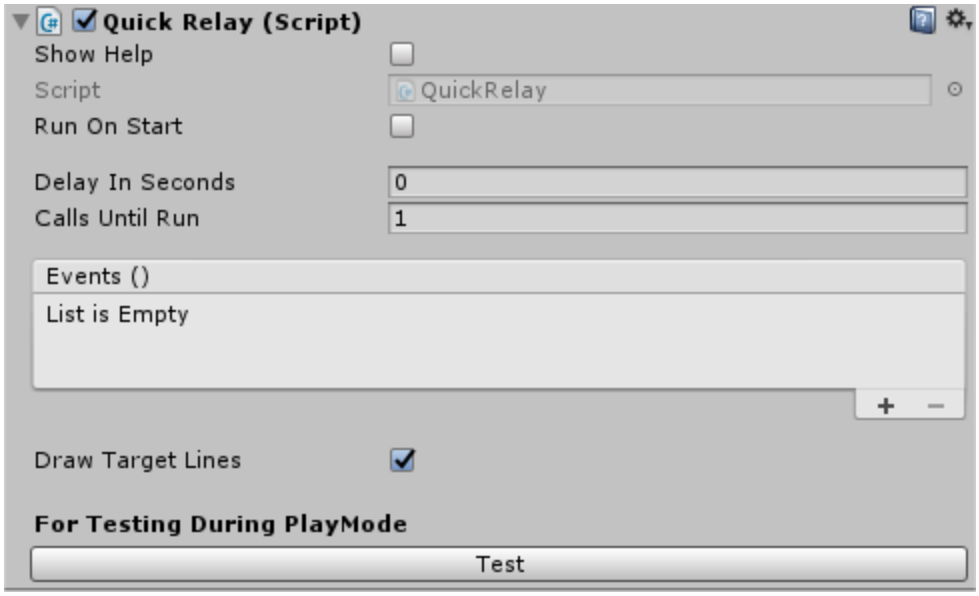
1. Test the Quick Relay by ticking 'Run On Start' to easily see if it is firing its Events when the game starts (It will fire its Events after the Delay In Seconds has passed).

2. Increase the Calls Until Run if you want the Quick Relay to require multiple Calls from objects in the scene before it fires its Events.

See Also

TO DO: Video Tutorial

Inspector Breakdown



INSPECTOR FIELD	TYPE	DEFAULT	DESCRIPTION
Run On Start	Bool	False	If ticked, the Relay will make on
Delay In Seconds	Float	0	How long (in seconds) between when the Quick Relay begins its Run and when it fires its Events
Calls Until Run	Int	1	How many times does this Relay need to be asked to Run before it begins its Run? (minimum is 1 = Run immediately when called) These other objects do this by calling the Quick Relay's publicly accessible Run() function. When Run() has been called the same number of times as Calls Until Run, the Quick Relay will fire all of its Events.
Events()	Unity Event		Will fire all Events in here when the Relay runs
Draw Target Lines	Bool	True	If ticked, the Scene View will draw a gizmo line to each game object the Quick Relay has an Event for

PUBLIC FUNCTIONS	PARAMETERS	DESCRIPTION
Call		The usual function to call when using the Relay node. Each call depletes the Relay's 'Calls Until Run' when it will fire
ForceRun		Skip the required calls and run the Relay after it's delay time passes
ForceRunImmediate		Run the Relay immediately, ignoring the required calls and the delay time

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 its maximum speed when 'Rotating' is ticked. It will decelerate when not ticked.

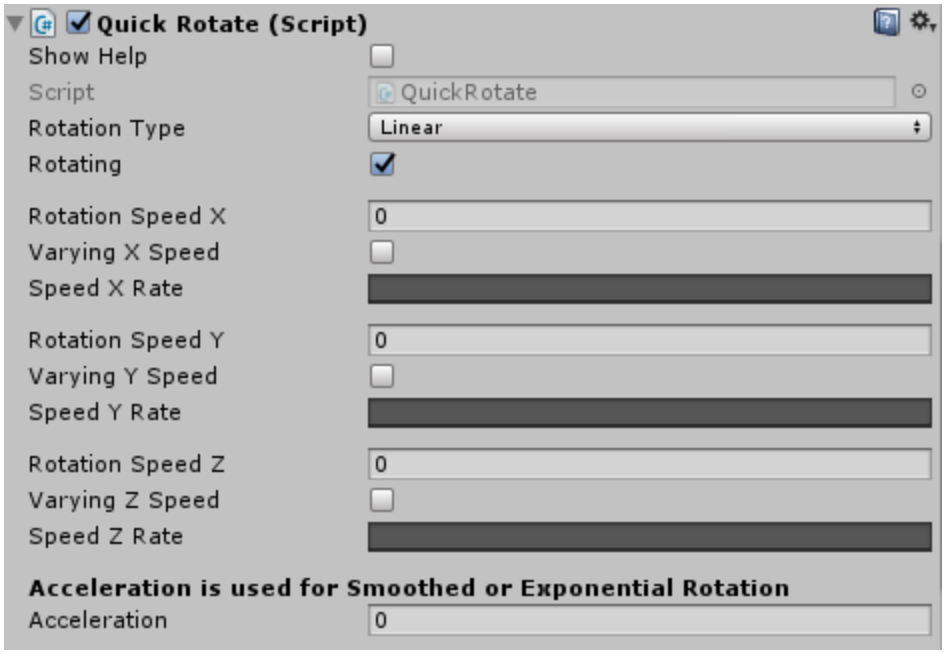
2. Only use Exponential rotation type if you plan on removing the object soon after. It is strongly recommended you do not leave it on exponential forever.

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

Inspector Breakdown



INSPECTOR FIELD	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 **WARNING**: Exponential should only be used if the object 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
Varying X Speed	Bool	False	If ticked, multiplies the X Rotation Speed by the Animation Curve over time

Speed X Rate	Animation Curve		An editable Animation Curve. Remember to set it to Loop for best results
Rotation Speed Y	Float	0	How fast to rotate on the Y Axis. Can be a negative number
Varying Y Speed	Bool	False	If ticked, multiplies the Y Rotation Speed by the Animation Curve over time
Speed Y Rate	Animation Curve		An editable Animation Curve. Remember to set it to Loop for best results
Rotation Speed Z	Float	0	How fast to rotate on the Z Axis. Can be a negative number
Varying Z Speed	Bool	False	If ticked, multiplies the Z Rotation Speed by the Animation Curve over time
Speed Z Rate	Animation Curve		An editable Animation Curve. Remember to set it to Loop for best results
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)

PUBLIC FUNCTIONS	PARAMETERS	DESCRIPTION
SetRotating	Bool	Set if the Rotator is rotating
SetRotateType	Int (0-2)	0 = Linear 1 = Accelerate 2 = Exponential
SetSpeedX	Float	Set the rotation speed on X Axis
SetSpeedY	Float	Set the rotation speed on Y Axis
SetSpeedZ	Float	Set the rotation speed on Z Axis

SetAcceleration	Float	Set the speed by which the rotator accelerates to max speed or decelerate to 0

QUICK SCALE

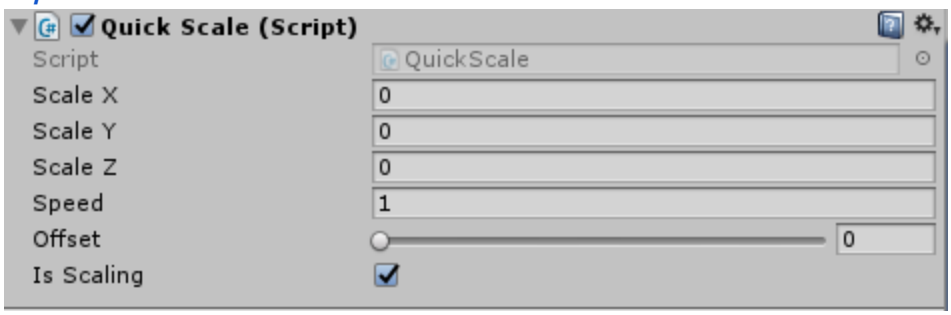
Overview

Applying this to a game object in your scene allows the object to scale up and down over time. It uses a sine wave to scale up and down over time, at the rate of Speed.

Quick Tips

1. Whatever value you put in any Scale X/Y/Z field will be both the positive and negative length of the scale distance
2. Ie if you have an object of Scale 1,1,1 and tell it to scale on X by 0.5, it will scale up to 1.5,1,1 and then down to 0.5,1,1
3. Because it uses a sine wave, you can increase the offset to make the timing different to other objects with the same settings

Inspector Breakdown



INSPECTOR FIELD	TYPE	DEFAULT	DESCRIPTION
Scale X	Float	0	The positive and negative distance this object will scale on the X axis
Scale Y	Float	0	The positive and negative distance this object will scale on the Y axis
Scale Z	Float	0	The positive and negative distance this object will scale on the Z axis

Speed	Float	1	How quickly should it scale up and down?
Offset	Float	0	Increasing this offsets its position on the sine wave function that controls the scaling, allowing it to be in or out of sync with other objects using the same settings
Is Scaling	Bool	True	If ticked, will scale up and down over time. If not, will not change its scale

QUICK SETPARENT

Overview

The Quick SetParent is the answer for keeping your characters and other physics objects attached to moving platforms in the scene. It requires a very specific set up, so if you are new to using it please watch the video under 'See Also'.

It is essentially a Trigger Volume that childs any object inside the trigger, and releases those objects back to their original parent when they leave the trigger. For this reason, the Quick SetParent *will require a Collider (is Trigger = ticked) on the same game object*.

Quick Tips

4. The Transform attached to the Quick SetParent should *always* have a scale of 1,1,1 unless you have set a custom Parent Transform (in which case the new Parent Transform's scale should be 1,1,1). If it is not a scale of 1,1,1 very strange things may happen to the objects being childed to the transform. You have been warned!

5. The Transform being used as the parent (in Parent Transform) needs a Collider set to 'is Trigger'. This collider is used as a trigger to catch / release all objects stepping in and out of the collider bounds.

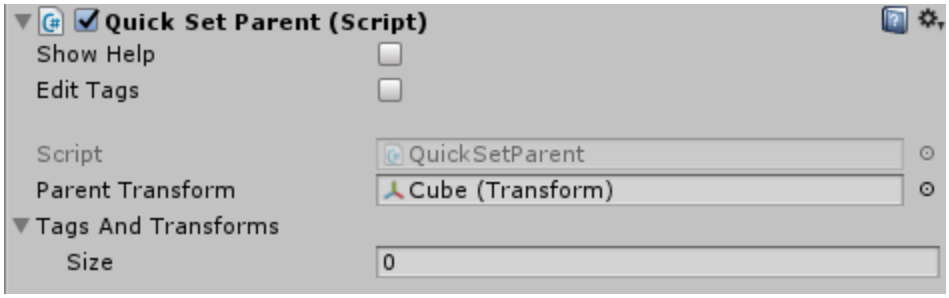
6. All objects you want the Quick SetParent to detect must have a Rigidbody

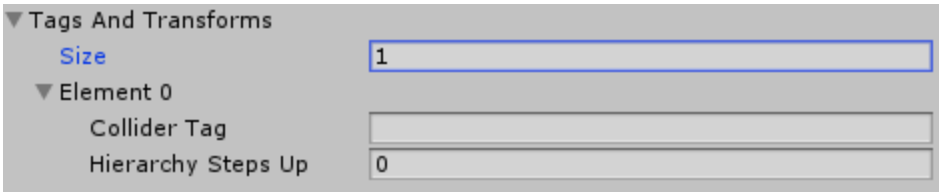
7. Tick 'Edit Tags' and add the relevant Tags you want the Quick SetParent to detect. Click Add Tag to automatically add them to the list under Tags and Transforms.

See Also

TO DO: Video Tutorial Link

Inspector Breakdown



INSPECTOR FIELD	TYPE	DEFAULT	DESCRIPTION
Edit Tags	Bool	False	Tick this to begin adding tags you want the SetParent's Collider to be able to detect
Parent Transform	Transform	The Transform the Quick SetParent is a Componen t of	The Transform here will be used to child all incoming objects to. Ie When a game object with a Rigidbody, Collider, and a matching Tag (under Tags and Transforms), steps into the Quick SetParent's Collider, that incoming game object's Transform will be made a child of the Parent Transform until it moves outside of the Collider.
			
Tags and Transform	List of QSParent _TagToTra nsform	Empty	Collider Tag = the Tag to look for of any Collider+Rigidbody entering the SetParent's Collider Hierarchy Steps Up = For complex game objects where the Collider+RB is not on the object's root transform, increase this number by how many steps to look up the hierarchy for the correct transform to use. Ie if you needed to re-parent a whole Character, but only the hand had the Collider and Rigidbody, and the hand was 9

			steps down in the Character's hierarchy of bones and joints, you would set this number to 9, so it would grab the root Transform for the whole Character instead of just the hand!

PUBLIC FUNCTIONS	PARAMETERS	DESCRIPTION
ExternalAddTag	String	Add a tag to the SetParent's list of tracked tags
ExternalRemoveLast		Remove the last tag in the SetParent's tracked tag list
ExternalRemoveTag	String	Remove a specific tag in the SetParent's tracked tag list

QUICK SHAKE

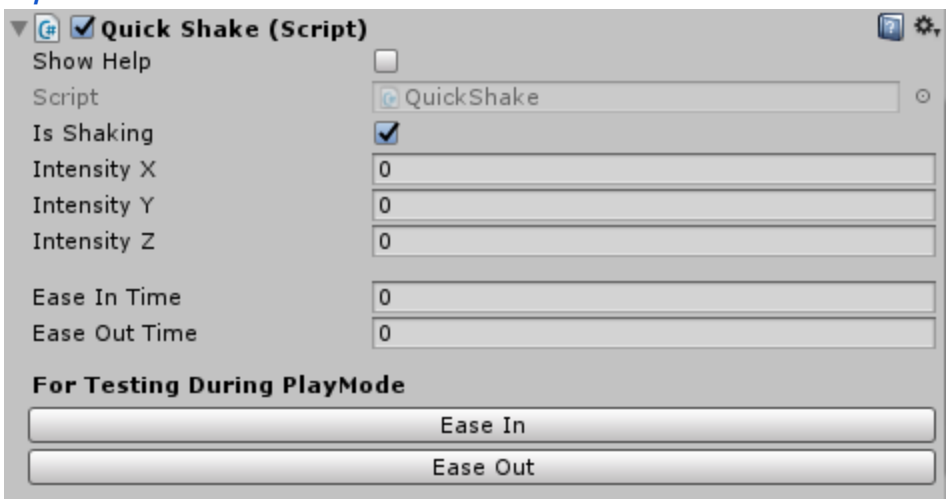
Overview

This is a useful component you can add to any object to make it shake or vibrate, with the option for it to ramp up or down from zero. There is also an inbuilt Public Function “Pulse” which will ramp it up and then back down over a designated number of seconds.

Quick Tips

1. If you want this object to Ease In or Out over time, you will need to call the relevant public function either from a Unity Event or through your own code.

Inspector Breakdown



INSPECTOR FIELD	TYPE	DEFAULT	DESCRIPTION
Is Shaking	Bool	True	During playmode, if true, the object will shake at the values set
Intensity X	Float	0	Amount to shake on the X axis
Intensity Y	Float	0	Amount to shake on the Y axis
Intensity Z	Float	0	Amount to shake on the Z axis
Ease In Time	Float	0	How many seconds it should take for this object to go from not shaking to the full amount as set in the Intensity values

Ease Out Time	Float	0	How many seconds it should take for this object to go from shaking at full Intensity to zero
For Testing During Play Mode			
Ease In	Button		Press this during Play Mode to make the object ramp up shaking from 0 to the Intensity values over Ease In Time
Ease Out	Button		Press this during Play Mode to make the object ramp down its shaking from the Intensity values to 0 over Ease Out Time

PUBLIC FUNCTIONS	PARAMETERS	DESCRIPTION
Ease In		Call this during Play Mode to make the object ramp up shaking from 0 to the Intensity values, over Ease in Time
Ease Out		Call this during Play Mode to make the object ramp up shaking from 0 to the Intensity values, over Ease Out Time
EaseInAndOut		Call this during Play Mode to make the object do the Ease In function, then immediately Ease Out. Useful for making pulse-shake effects

QUICK SLIDE

Overview

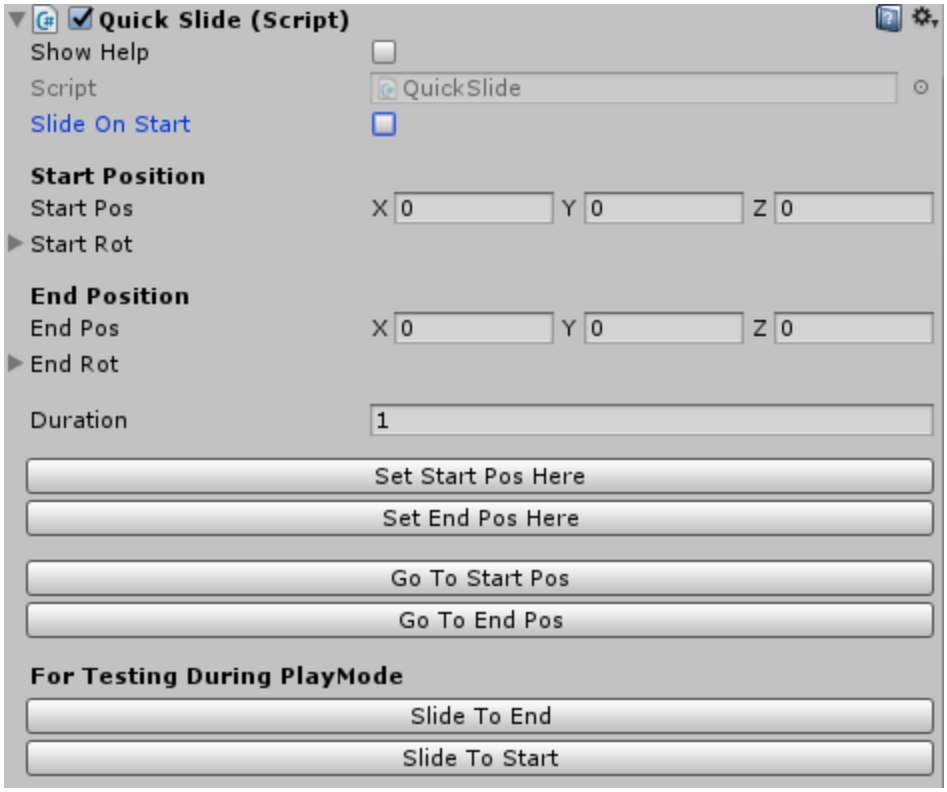
The Quick Slide allows an object to move between two set positions and rotations, over a designated number of seconds. It is one of the more ‘quick and easy’ components of the Quick Scripts pack.

Think of it as a highly simplified sliding Quick Door.

Quick Tips

1. Make sure you set both a Start Position and an End Position by moving the object where you want it and then pressing the relative button below.
2. You can make use of Public Functions via any Event System (Quick Trigger, Relays, etc) or through your own custom code to set new Start and End positions at runtime.

Inspector Breakdown



INSPECTOR FIELD	TYPE	DEFAULT	DESCRIPTION
Slide On Start	Bool	False	If ticked, will run as soon as game starts or as soon as this object is instantiated

Start Pos	Vector 3		The World Position where this game object will start
Start Rot	Quaternion		The local Rotation this game object will have at the start of its slide
End Pos	Vector 3		The World Position where this game object will be at the end of its slide
End Rot	Quaternion		The local Rotation this game object will have at the end of its slide
Duration	Float	1	How many seconds it takes to complete the slide
Set Start Pos Here	Button		Will set the Start Pos and Start Rotation at the game object's current position
Set End Pos Here	Button		Will set the End Pos and End Rotation at the game object's current position
Go To Start Pos	Button		Snap to the set Start Pos and Rotation
Go To End Pos	Button		Snap to the set End Pos and Rotation
For Testing During Play Mode			
Slide To End	Button		If pressed during Play Mode, will slide the object from Start to End
Slide To Start	Button		If pressed during Play Mode, will slide the object from End to Start

PUBLIC FUNCTIONS	PARAMETERS	DESCRIPTION
Slide To End		If called during Play Mode, will slide the object from Start to End
Slide To Start		If called during Play Mode, will slide the object from Start to End

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.

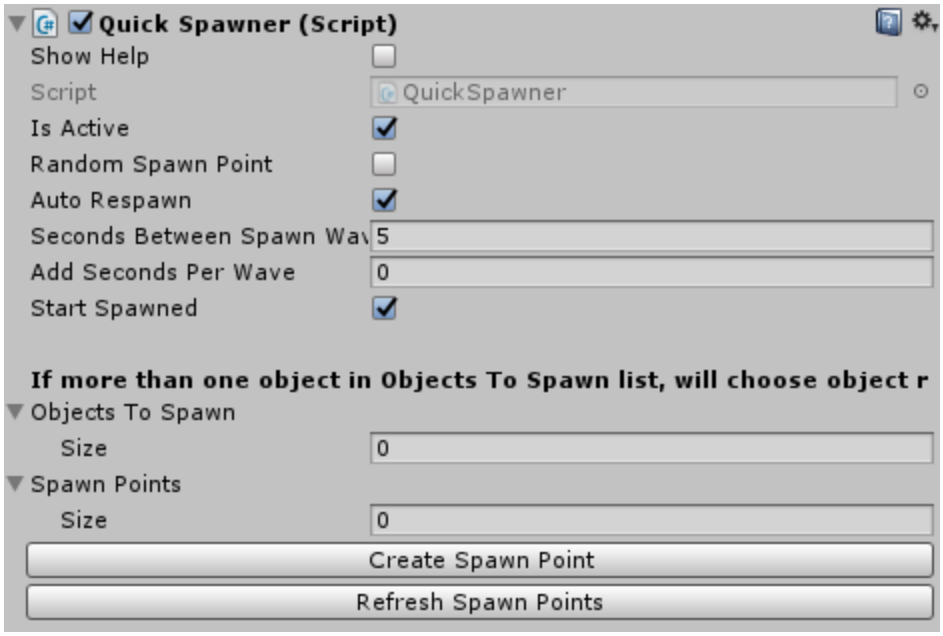
6. If you click the dropdown arrow next to 'Objects To Spawn' and change 'Size' 0 to 1 or more, you can drag and drop objects into these slots. These are the objects the Quick Spawner will spawn at the Spawn Points. If you add multiple objects to the list of Objects To Spawn, the Quick Spawner will spawn 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. Object pooling is not currently implemented.

See Also

How To Correctly Set Up And Use The Quick Spawner

Inspector Breakdown



INSPECTOR FIELD	TYPE	DEFAULT	DESCRIPTION
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	0	Increase the size above zero and fill the fields with any game objects you want to

	Object)		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

PUBLIC FUNCTIONS	PARAMETERS	DESCRIPTION
Spawn		Immediately run the Spawn function, the Spawner must be active for this to happen
SetSpawnerActive	Bool	Set if the Spawner is active and able to spawn
SetSecondsBetweenWaves	Float	Set how many seconds default are between spawn waves
SetAddSecondsPerWave	Float	Set how many seconds are added to each spawn wave
SetRandomSpawnPoint	Bool	Set whether or not the Spawner chooses a random spawn point from its list
SetAutoRespawn	Bool	Set whether or not the Spawner continues to respawn on its own
SetStartSpawned	Bool	Set whether or not the Spawner starts and immediately spawns

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

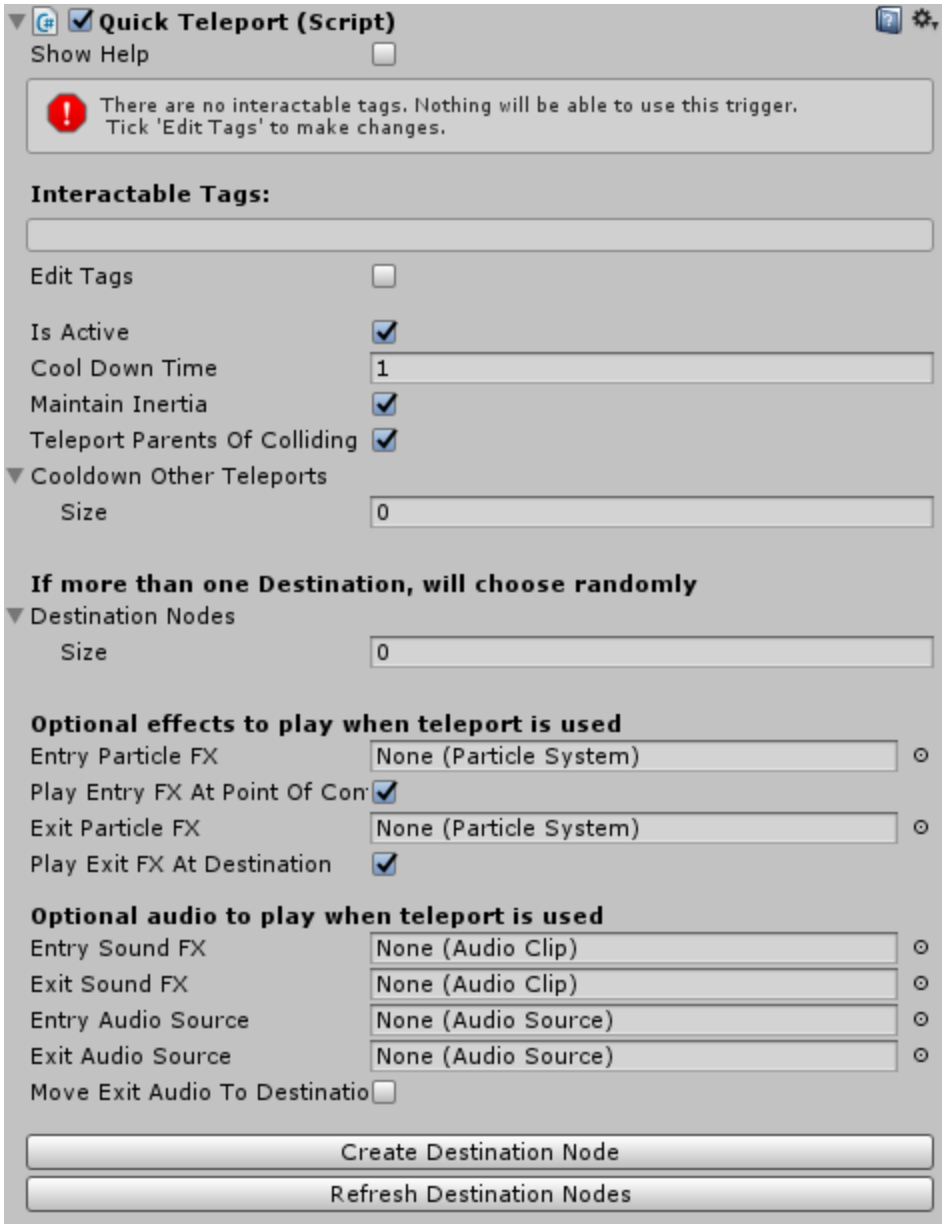
1. Click 'Create Destination Node' to make a point to teleport to.
2. Add more than one destination node to create a teleporter that sends to random destinations.
3. Remember to fill out the list of interactable tags for what game objects can be teleported.
4. Teleported objects will face the same way 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

Inspector Breakdown



INSPECTOR FIELD	TYPE	DEFAULT	DESCRIPTION
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’
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.
Other Teleport Cooldowns	List (Quick Teleports)		Any teleports added here will have their cooldown automatically triggered when you use this teleport. Useful for 2-way teleport setups where the destinations are inside the other teleports’ triggers
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!
Create Destination Nodes	Button		If pressed, will make a new destination node
Refresh Destination Nodes	Button		If pressed, will clean up the list of destination nodes to make sure there are no missing nodes

PUBLIC FUNCTIONS	PARAMETERS	DESCRIPTION
SetCooldownTime	Float	Set the time in seconds for the cooldown before the Teleporter can be reused

SetTeleportActive	Bool	Set if the Teleporter can be used
SetMaintainIntertia	Bool	Set whether or not the Teleport maintains the inertia of objects entering it

QUICK TRACK

Overview

The Quick Track is by far the most complex feature of the Quick Scripts pack, due to it also providing the most customisation and flexibility for your use case.

Most important is to be familiar with its two modes: Simple and Complex.

Simple Quick Tracks create a path that moves directly node-to-node with no curvature or rotation along the path.

Complex Quick Tracks use bezier splines between nodes to allow you to create tracks that bend, turn, twist, rotate, go upside down, etc, with in-editor visualisation to help understand what direction the track is facing at any point.

There are three levels of detail in a Quick Track: the Quick Track itself, Track Sections, and Track Nodes.

Quick Track → *contains* → Track Sections → *contains* → Track Nodes

Each Quick Track, Track Section and Track Nodes contain customisation options that will affect all Quick Movers travelling along the track, so be sure to check them if you are looking for specific behaviours (or debugging a track!)

NOTE: See Below for Inspector Breakdowns for both Track Section and Track Node

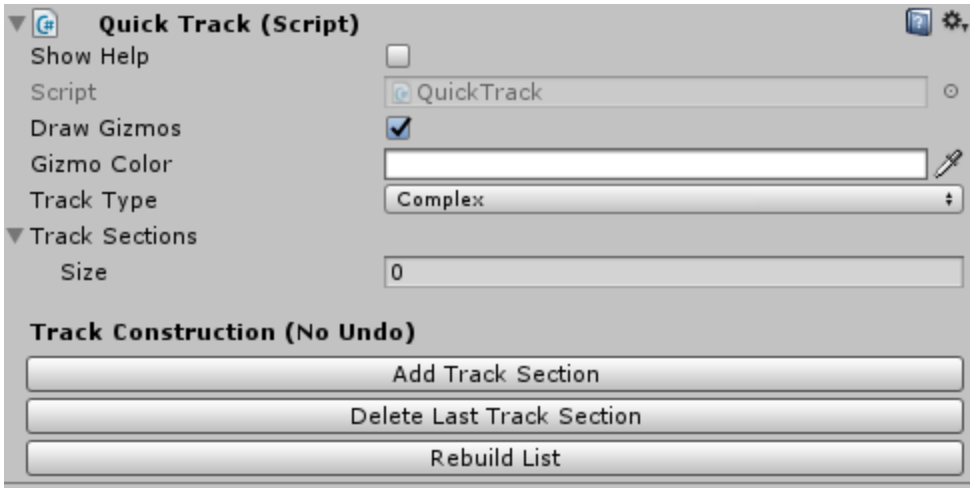
Quick Tips

1. Always add or delete your Quick Tracks using the buttons below or the buttons on the Track Sections. Do not delete Track Sections or sub game objects from the Hierarchy or you risk breaking the whole track!
2. There is no undo when using the Construction / Positioning buttons. Please keep this in mind.
3. Quick Tracks contain Track Sections, which contain Track Nodes. Tracks, Sections and Nodes have their own customisation options so be sure to check them all!
4. If the editor is lagging and you are using many tracks or have one very very long track, try unticking ‘Draw Gizmos’

See Also

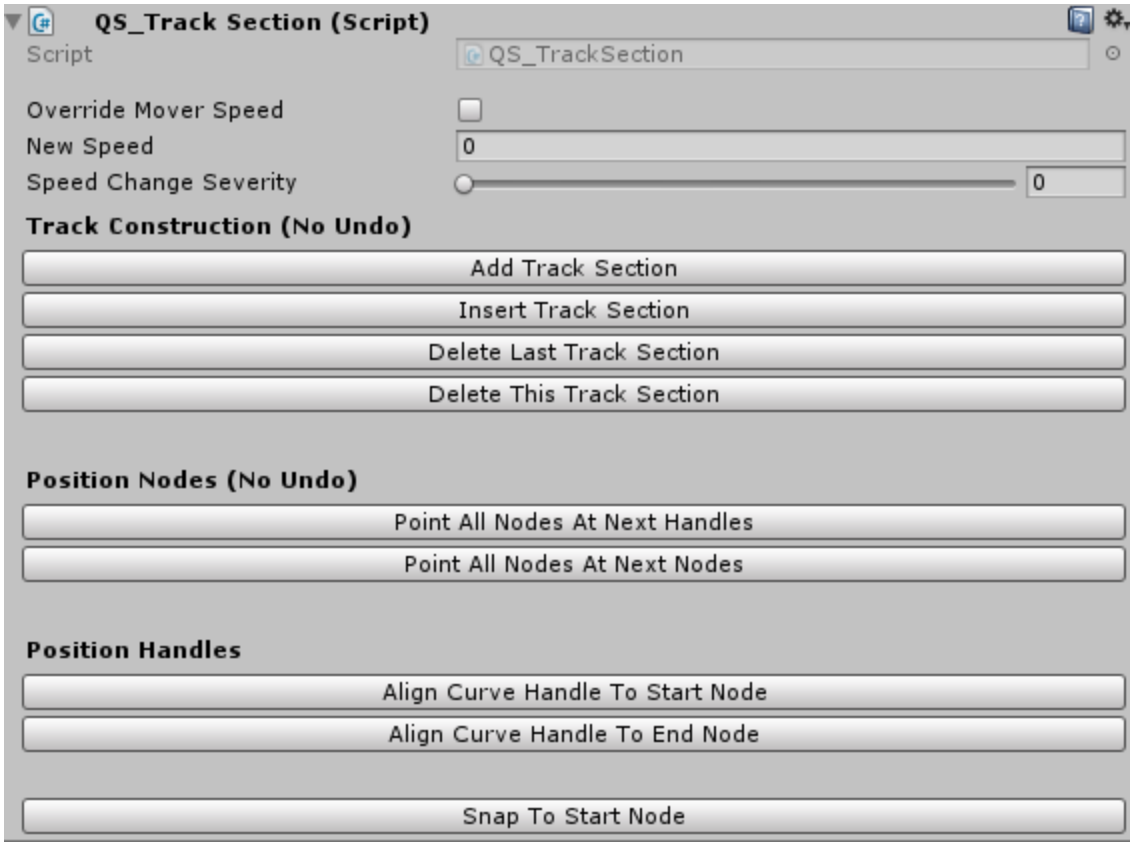
TO DO: Video Tutorial

Inspector Breakdown - Quick Track



INSPECTOR FIELD	TYPE	DEFAULT	DESCRIPTION
Draw Gizmos	Bool	True	Shows the track gizmos in the Editor Scene window
Gizmo Colour	Color	White	Sets the colour of the track gizmos in the Editor Scene window. Useful if you have multiple tracks in one scene
Track Type	Enum	Complex	Is this a Complex track (bezier curves between nodes) or Simple? (direct node to node)
Track Sections	List (QS_TrackSection)		The collection of Track Sections that make up this one Track. Do not edit directly
Track Constriction (No Undo)			
Add Track Section	Button		Adds a new Track Section at the end of the track
Delete Last Track Section	Button		Deletes the Track Section at the end of the track
Rebuild List	Button		Rebuilds the Track Sections List incase something has been deleted

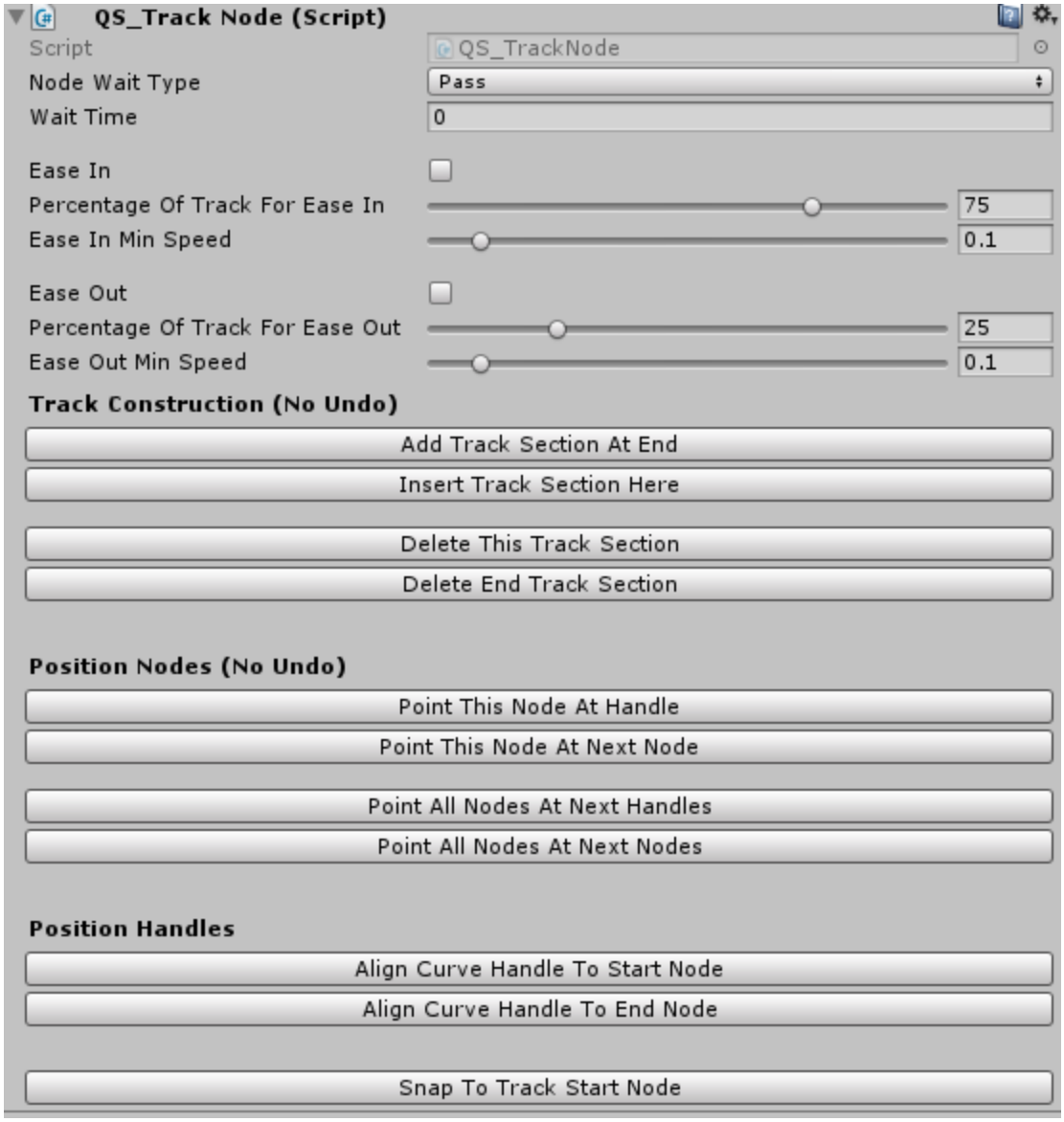
Inspector Breakdown - QS Track Section



INSPECTOR FIELD	TYPE	DEFAULT	DESCRIPTION
Override Mover Speed	Bool	False	Allow this Track Section to change the Move Speed of any Quick Mover as it travels along it?
New Speed	Float	0	If 'Override Mover Speed' is true, this is the new speed the Mover will be set to as it reaches this Track Section
Speed Change Severity	Float (0 - 1)	1	How immediate does the speed change come into effect? 1 = immediate Less than 1 = Lerps the speed using this value as a percentage of distance along the track. Ie a value closer to 0 = the speed will lerp to the new target closer to the end of this Track Section. 0 = will change to the new speed as soon as the Mover reaches the end of this Track Section
Track Construction (No Undo)			
Add Track Section	Button		Adds a new Track Section at the end of the track
Insert Track Section	Button		Adds a new Track Section directly after the currently selected Track Section
Delete Last Track Section	Button		Deletes the Track Section at the end of the track
Delete This Track Section	Button		Deletes the currently selected Track Section
Position Nodes (No Undo)			
Point All Nodes At Next Handles	Button		Rotates all Track Nodes to look at the next handle in each Track Section
Point All Nodes at Next Nodes	Button		Rotates all Track Nodes to look at the next Track Node along the track

Position Handles			
Align Curve Handle to Start Node	Button		Move the curve handle in this Track Section to be in line with the section's Start Node
Align Curve Handle to End Node	Button		Move the curve handle in this Track Section to be in line with the section's End Node
Snap To Start Node	Button		Move this track section to the Start Node of the whole track

Inspector Breakdown - QS Track Node



INSPECTOR FIELD	TYPE	DEFAULT	DESCRIPTION
Node Wait Type	Enum	Pass	Pass or Wait. Should Quick Movers pass through this node or wait here?
Wait Time	Float	0	If Node Wait Type is 'Wait', how long should the Mover wait here? (In Seconds)
Ease In	Bool	False	Should Quick Movers smoothly slow down to a minimum speed as they

			approach this node?
Percentage Of Track For Ease In	Float (Range 1 - 100)	75	What percentage of the Track Section should be used to slow the Mover down? Ie 50% will use the second half of the Track Section, 25% will use the last quarter, 10% will use the final tenth, etc
Ease In Min Speed	Float (Range 0.01 - 1)	0.1	What is the speed the Mover will slow down to as it completes the Ease In?
Ease Out	Bool	False	Should Quick Movers smoothly speed up to its regular speed as they move away from this node?
Percentage Of Track For Ease Out	Float (Range 1 - 100)	75	What percentage of the **NEXT** Track Section should be used to speed the Mover up? Ie 50% will use the first half of the next Track Section, 25% will use the first quarter, 10% will use the first tenth, etc
Ease Out Min Speed	Float (Range 0.01 - 1)	0.1	What is the speed the Mover will start at as it completes the Ease Out?
Track Construction (No Undo)			
Add Track Section	Button		Adds a new Track Section at the end of the track
Insert Track Section	Button		Adds a new Track Section directly after the currently selected Track Section
Delete End Track Section	Button		Deletes the Track Section at the end of the track
Delete This Track Section	Button		Deletes the currently selected Track Section
Position Nodes (No Undo)			
Point This Node At Handle	Button		Rotates this Track Node to look at the curve handle in its Track Section

Point This Node at Next Node	Button		Rotates this Track Nodes to look at the next Track Node along the track
Point All Nodes At Next Handles	Button		Rotates all Track Nodes to look at the next handle in each Track Section
Point All Nodes at Next Nodes	Button		Rotates all Track Nodes to look at the next Track Node along the track
Position Handles			
Align Curve Handle to Start Node	Button		Move the curve handle in this Track Section to be in line with the section's Start Node
Align Curve Handle to End Node	Button		Move the curve handle in this Track Section to be in line with the section's End Node
Snap To Track Start Node	Button		Move this track node to the Start Node of the whole track. Useful for setting up a clean looping track.

QUICK TRIGGER

Overview

The Quick Trigger is a script applied to a collider which has ‘is Trigger’ ticked. 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.

PLEASE NOTE: In order for a Trigger to work, itself or the game object entering it MUST have a collider and rigidbody.

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, tick 'Edit Tags'. Then, select a tag from the dropdown field. Then press Add Tag. It will be added to the 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.

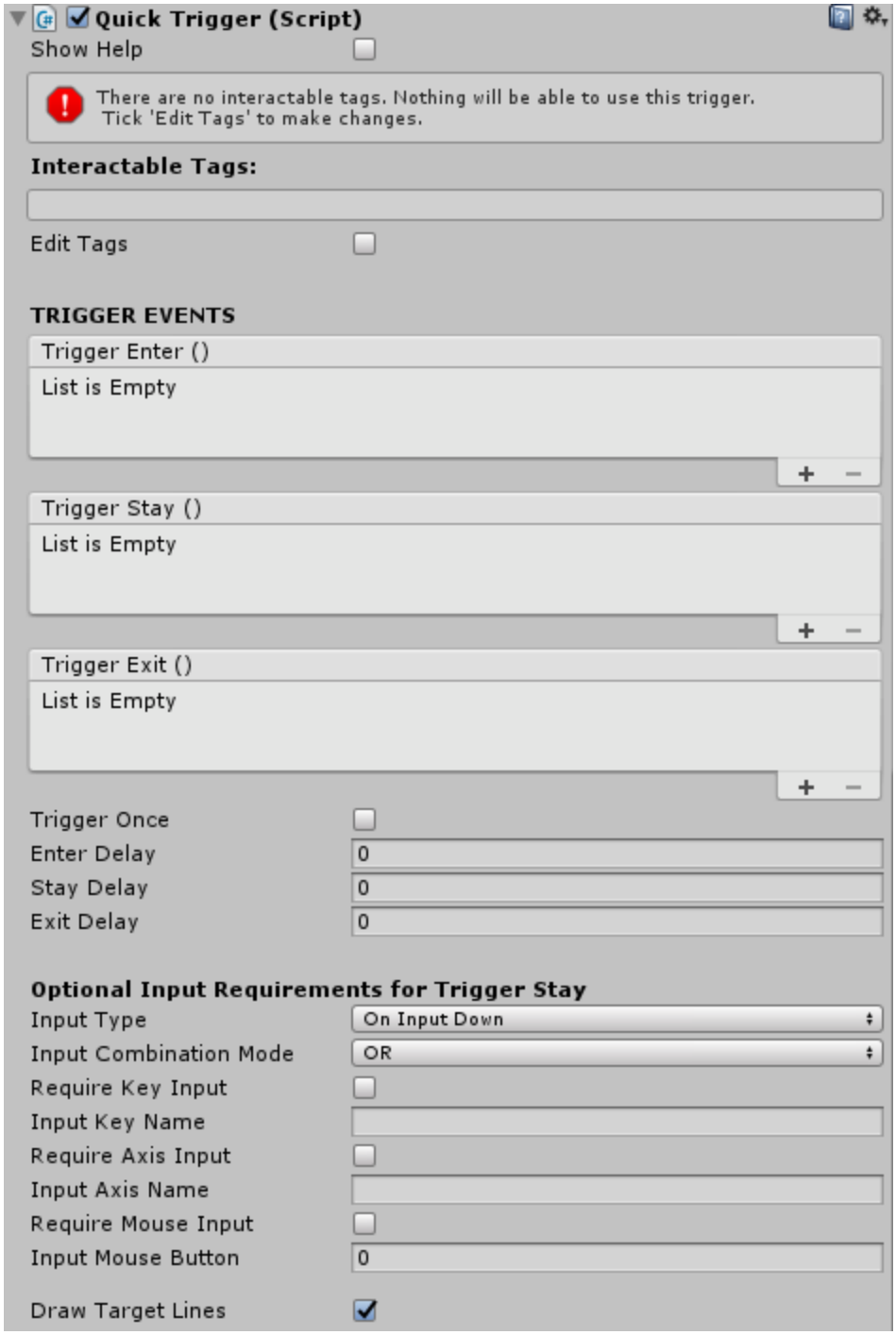
3. To set up events to happen when something enters, stays within, or exits the Trigger follow these steps:
- Click the + sign underneath TriggerEnter, TriggerStay or TriggerExit to add an event.
- An empty object field will appear. You can drag and drop any object from the Hierarchy window into this 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'. Here is a useful link to Unity 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

TO DO: Video Tutorial

Inspector Breakdown



INSPECTOR FIELD	TYPE	DEFAULT	DESCRIPTION
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’
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? After firing it’s Events once, it will never be able to fire again.
Enter Delay	Float	0	The delay (in seconds) before the Trigger Enter() Events fire
Stay Delay	Float	0	The delay (in seconds) before the Trigger Stay() Events fire
Exit Delay	Float	0	The delay (in seconds) before the Trigger Exit() Events fire
Input Type	Dropdown Enum	On Input	Specify what the player should be doing to register the input has been received.

		Down	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
Draw Target Lines	Bool	True	If ticked, the Scene View will show gizmo lines pointing to each game object in the Trigger Enter, Stay and Exit Events

PUBLIC FUNCTIONS	PARAMETERS	DESCRIPTION
SetTriggerOnce	Bool	Set whether or not the Trigger will only trigger once per play session
DeActivateTrigger	Bool	Set whether or not the Trigger can be activated
RequireKeyInput	Bool	Set if the Trigger requires Key Input to be activated
RequireMouseInput	Bool	Set if the Trigger requires Mouse Button Input to be activated
RequireAxisInput	Bool	Set if the Trigger requires a project-specified Axis Input to be activated
SetKeyInputName	String	Takes a string which is the name of a key that needs to be pressed to activate the Trigger
SetAxisInputName	String	Takes a string which is the name of an existing project-specified Input Axis which is required
SetMouseInputButton	Int (0 - 2)	Set which mouse button needs to be pressed to activate the Trigger. (See above)
SetInputType	Int (0-2)	Set which Input method the Trigger takes in order to be activated (See above)
SetInputCombination	Int (0-1)	Set what method of Input Combination must be used to activate the Trigger (See above)

VIDEO TUTORIALS

TODO: Remake all these videos. Outdated!

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 mail.jackwilson@gmail.com

You can also contact me on Twitter: @t_jackwilson