Documentation

Version 1.0

1. Introduction

SwitchNTrap is a compact, state based controlling system for traps, elevators, switches, levers, doors and much more. The main workflow is based primarily on rotation and movement of objects. You can easily create chaining events for example if something is touching a specific object it opens a door somewhere, it moves a trap and after it's finished it does something else. SwitchNTrap comes with a pack of modular 3D PRB assets (that you can use standalone) for your dungeons or other types of environment.

Create your own combination of materials for your custom lever in the demo scene with the Lever generator.

The system supports any rotations without limits, movement on multiple position nodes, swinging of objects and manipulation of some extra options such as material switching.

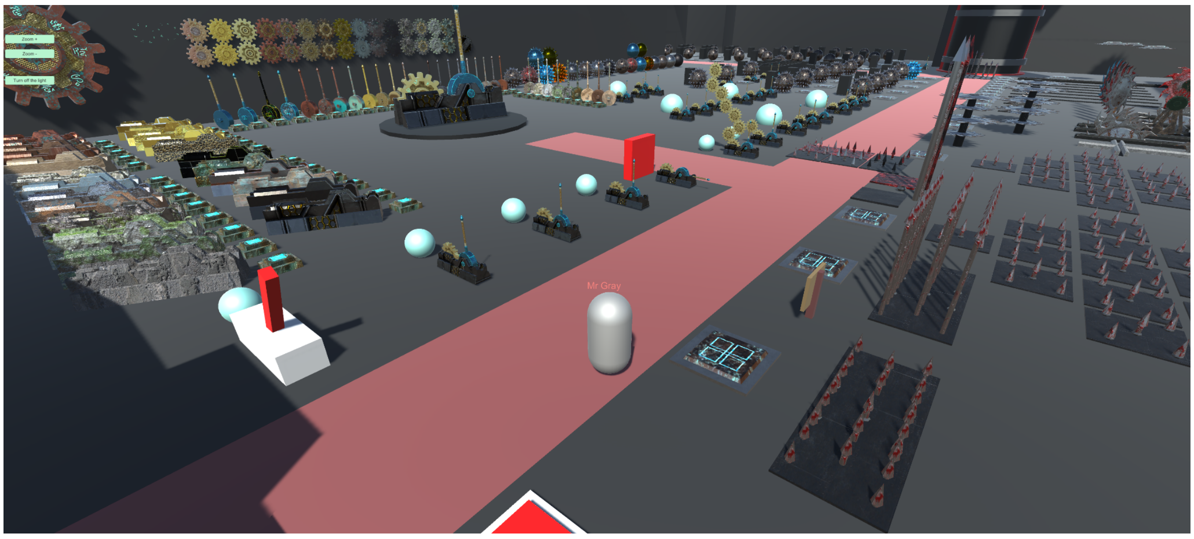
It provides visual support for better control of things in your scene. Drag and drop your sounds. You can build in minutes the whole interactive level after some practice. It can enhance your work in any kind of game greatly.

Let your fantasy free!

Latest documentation link

If you need help use my discord channel:

Help on Discord



2. Installation

Please do not move the package to other asset folders or the icons will not work. It will be supported in later versions.
But you can safely move or copy prefabs and materials and place your custom scripts outside.

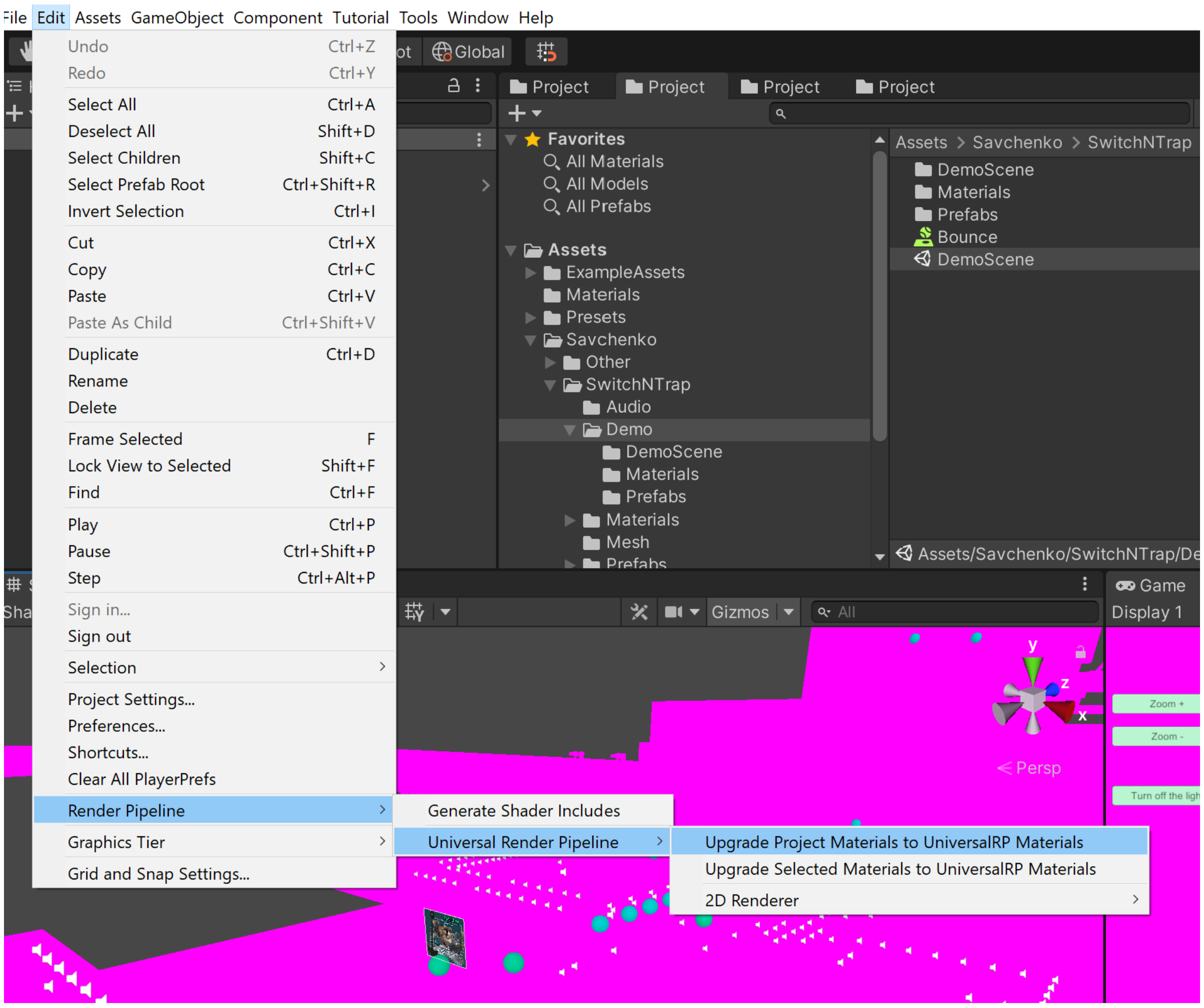
For better understanding and visual experience it's best to test the demo scene in the build-in rendering pipeline project.

Build-in

1. Import the package.
2. Open the demo scene “Assets\Savchenko\SwitchNTrap\DemoScene.unity” and look at the examples.

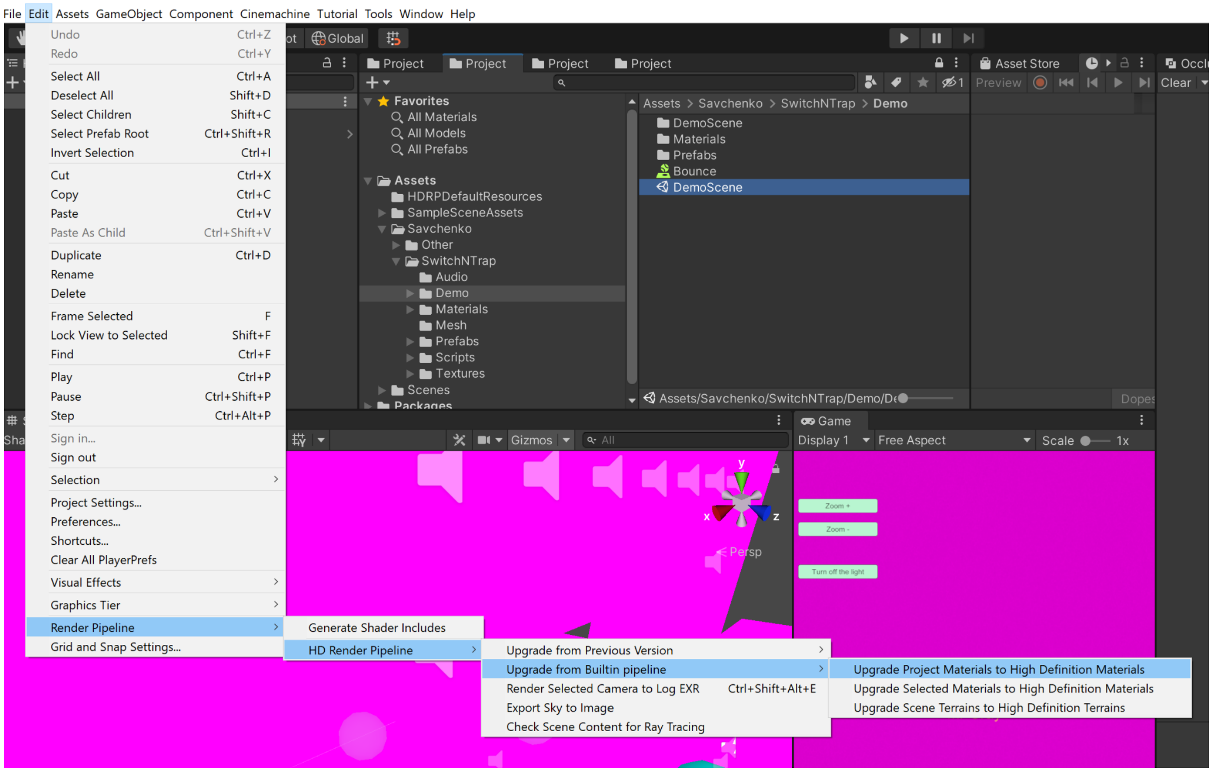
URP

1. Import the package.
2. Open the demo scene “Assets\Savchenko\SwitchNTrap\DemoScene.unity”.
3. Convert the materials/shaders to URP (Make copy of your project in case)
4. (Optional) adjust Skybox and light for the demo scene. The Demo scene was made on Built-In RP and will look darker on URP on default.
5. Look at the examples.

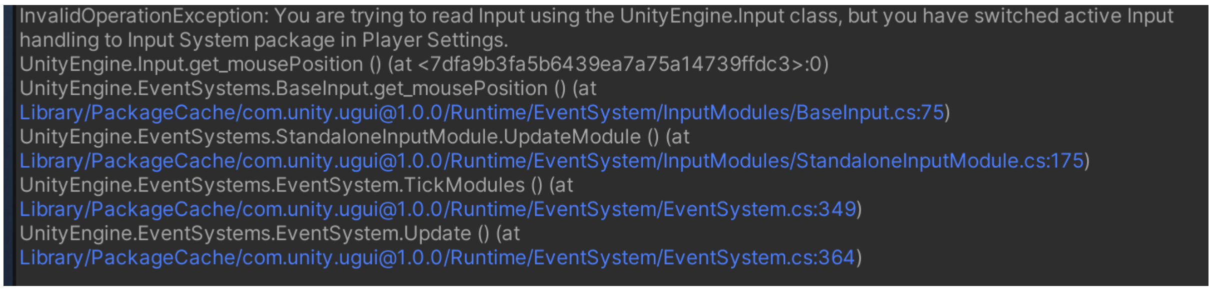


HDRP

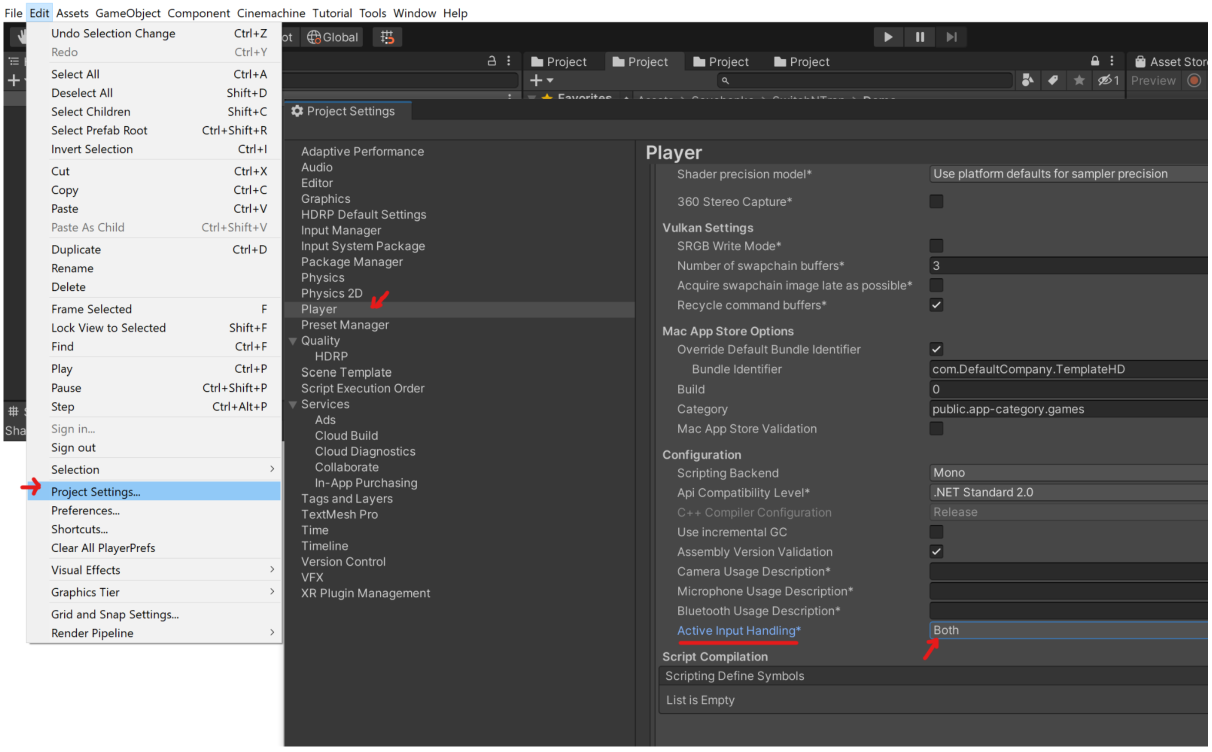
1. Import the package.
2. Open the demo scene “Assets\Savchenko\SwitchNTrap\DemoScene.unity”.
3. Convert the materials to HDRP (Make copy of your project in case)
4. (Optional) adjust Skybox and light for the demo scene. The Demo scene was made on Built-In RP and will look darker on HDRP on default.
5. Look at the examples.



6. If InputSystem error occurs. Fix it by setting “Active Input Handling” to “Both” in player settings.



```
InvalidOperationException: You are trying to read Input using the UnityEngine.Input class, but you have switched active Input handling to Input System package in Player Settings.
UnityEngine.Input.get_mousePosition () (at <7dfa9b3fa5b6439ea7a75a14739ffdc3>:0)
UnityEngine.EventSystems.BaseInput.get_mousePosition () (at Library/PackageCache/com.unity.ugui@1.0.0/Runtime/EventSystem/InputModules/BaseInput.cs:75)
UnityEngine.EventSystems.StandaloneInputModule.UpdateModule () (at Library/PackageCache/com.unity.ugui@1.0.0/Runtime/EventSystem/InputModules/StandaloneInputModule.cs:175)
UnityEngine.EventSystems.EventSystem.TickModules () (at Library/PackageCache/com.unity.ugui@1.0.0/Runtime/EventSystem/EventSystem.cs:349)
UnityEngine.EventSystems.EventSystem.Update () (at Library/PackageCache/com.unity.ugui@1.0.0/Runtime/EventSystem/EventSystem.cs:364)
```

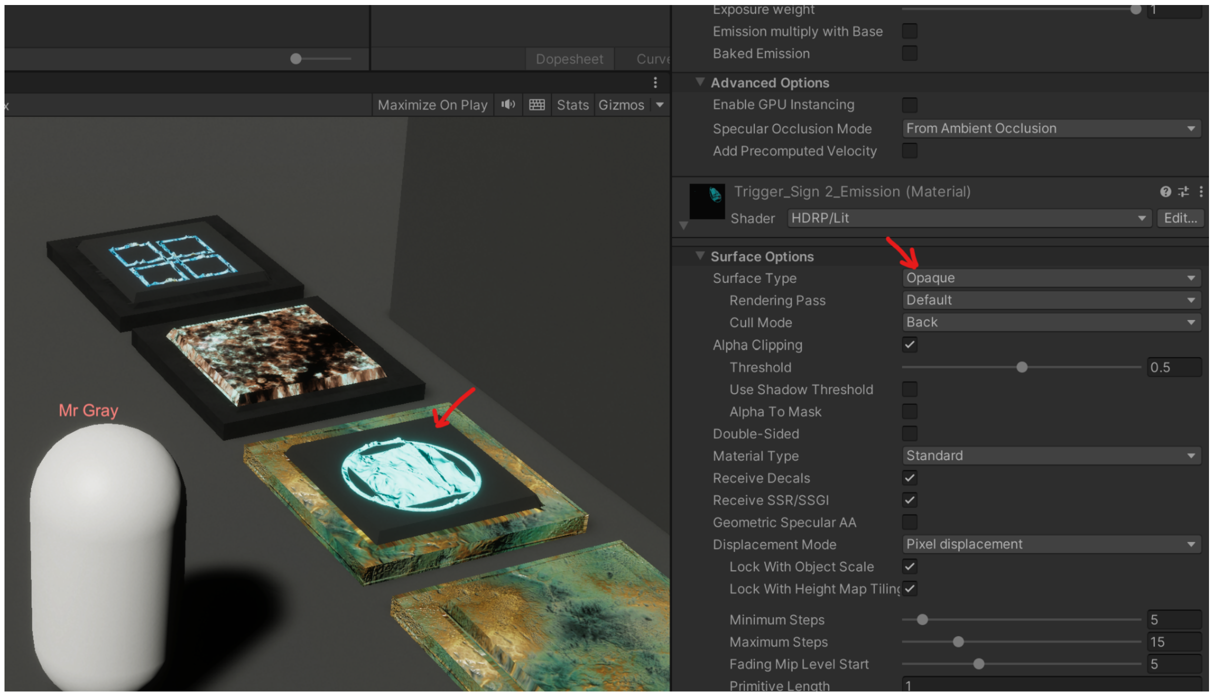


Player

Shader precision model*	Use platform defaults for sampler precision
360 Stereo Capture*	☐
Vulkan Settings	
SRGB Write Mode*	☐
Number of swapchain buffers*	3
Acquire swapchain image late as possible*	☐
Recycle command buffers*	✓
Mac App Store Options	
Override Default Bundle Identifier	✓
Bundle Identifier	com.DefaultCompany.TemplateHD
Build	0
Category	public.app-category.games
Mac App Store Validation	☐
Configuration	
Scripting Backend	Mono
Api Compatibility Level*	.NET Standard 2.0
C++ Compiler Configuration	Release
Use incremental GC	☐
Assembly Version Validation	✓
Camera Usage Description*	
Microphone Usage Description*	
Bluetooth Usage Description*	
Active Input Handling*	Both

Script Compilation
Scripting Define Symbols
List is Empty

7. If some of the transparent materials turn black, give it some time(~5 minutes). If it wont help, select light in the scene and if it still wont help, check those few materials and set “Surface Type” to transparent. I will include separate packages in later versions for all rendering pipelines.



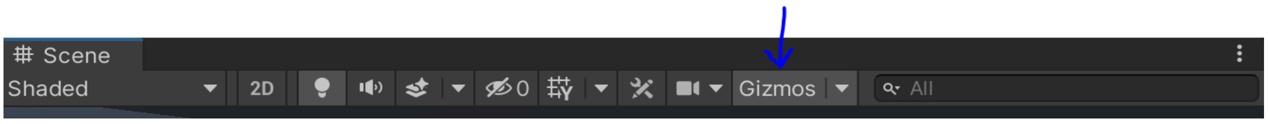
Exposure weight	1
Emission multiply with Base	☐
Baked Emission	☐
Advanced Options	
Enable GPU Instancing	☐
Specular Occlusion Mode	From Ambient Occlusion
Add Precomputed Velocity	☐

Trigger_Sign 2_Emission (Material)
Shader HDRP/Lit Edit...

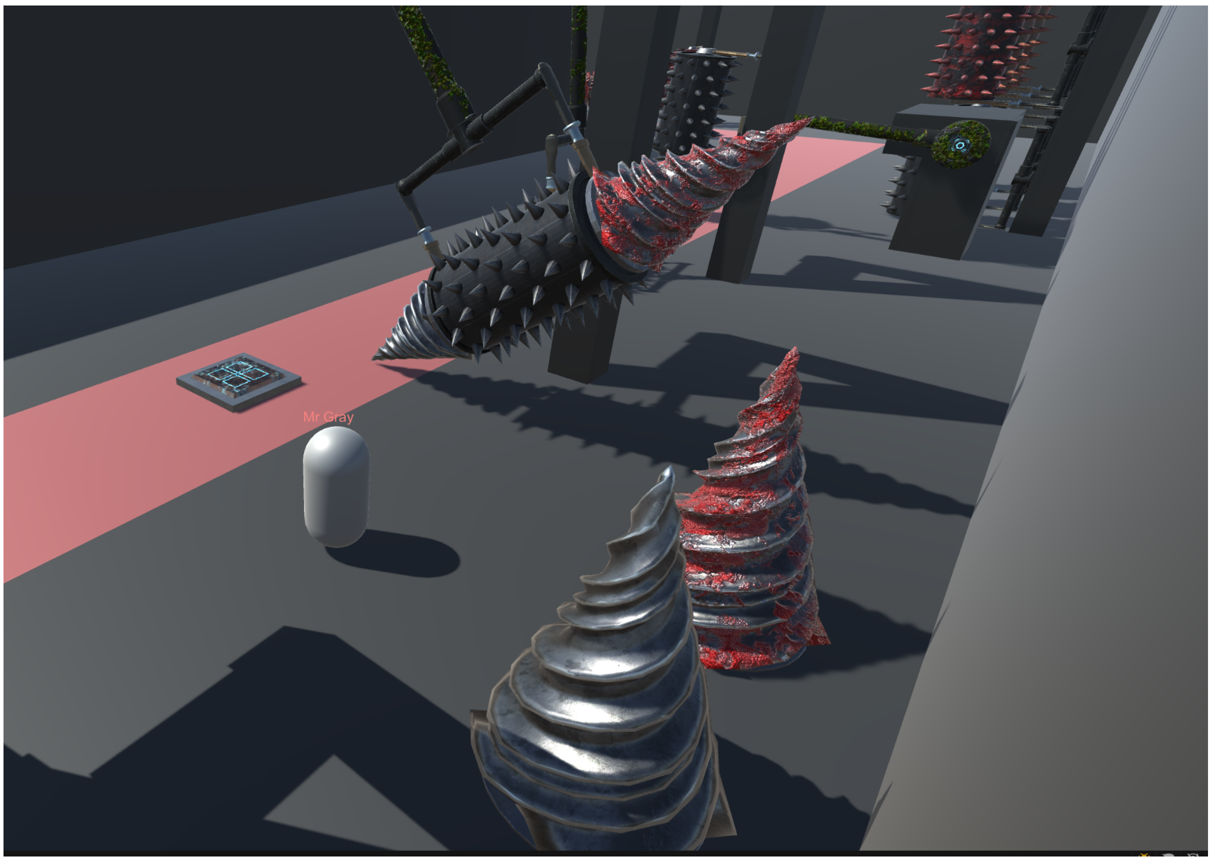
Surface Options	
Surface Type	Opaque
Rendering Pass	Default
Cull Mode	Back
Alpha Clipping	✓
Threshold	0.5
Use Shadow Threshold	☐
Alpha To Mask	☐
Double-Sided	☐
Material Type	Standard
Receive Decals	✓
Receive SSR/SSGI	✓
Geometric Specular AA	☐
Displacement Mode	Pixel displacement
Lock With Object Scale	✓
Lock With Height Map Tiling	✓
Minimum Steps	5
Maximum Steps	15
Fading Mip Level Start	5
Primitive Length	1

3. Notes

- For scene-view gizmos, button “F” is your best friend. Focus on objects with SwitchController to have a better visualization on it.
- For gizmos visualization MainSwitchController or NodeWalker must be selected and gizmos must be turned on.



- Do not move SwitchNTrap to other directories in the assets folder. It's not supporting it right now, but will change in a later version.
- While using movement or rotation manipulating SwitchControllers do not change the rotation or rotation by other scripts or events (RigidBody physics) on the same ActionTarget or it could lead to not expected results.
- The hierarchy icons use lazy load to not hit on performance of the editor. By selecting items, icons will reload.
- Use demo scene for examples
- Make a copy of prefabs if you want to modify them or else the demo scene can break.
- For easier configuration of target objects for movement or rotation, place them into parents.
- For use on older mobiles: The assets have mid-poly count and could be used in older mobile devices as well. In case the performance hits, the materials must be reduced and the maximum texture size must be decreased in configuration for that. (duplicate material, remove Normal- Map, heightmap… til it fits your needs)



4. Usage SwitchControllers

SwitchControllers are working as state machines. There are basic four states:

- Deactivated: Deactivating task is done.
- Activating: Task is doing job and is transferring itself on finish to Activated.
- Activated: Activating task is done.
- Deactivating: Task is doing job and is transferring itself on finish to Deactivated.

Depending on the script the tasks vary.
Some scripts are reduced to Activated/Deactivated state instead of Activating->Activated.
All positions used in the configuration are "localPositions" so you might put TargetObjects into parents to easier setup the positions from 0.

The scripts that implement ISwitchController can be connected in chains. You can easily write your own scripts that will work with others in a chain.
Look at the demo scene for examples or you can just drag and drop combinations to the asset folder to create prefabs for your game from it.

All SwitchControllers implement ISwitchController interface.
You can easily set the state of any SwitchController inside any of your scripts by calling for example (for Activating):

```
var switchController = targetGameObject.GetComponent<ISwitchController>();
switchController.OnSetStateByCaller(TriggerStateEnum.Activating);
```

look for example on OnCollision_SwitchControllerStateSetter.

You can get current state by:

```
switchController.OnGetStateByCaller();
```

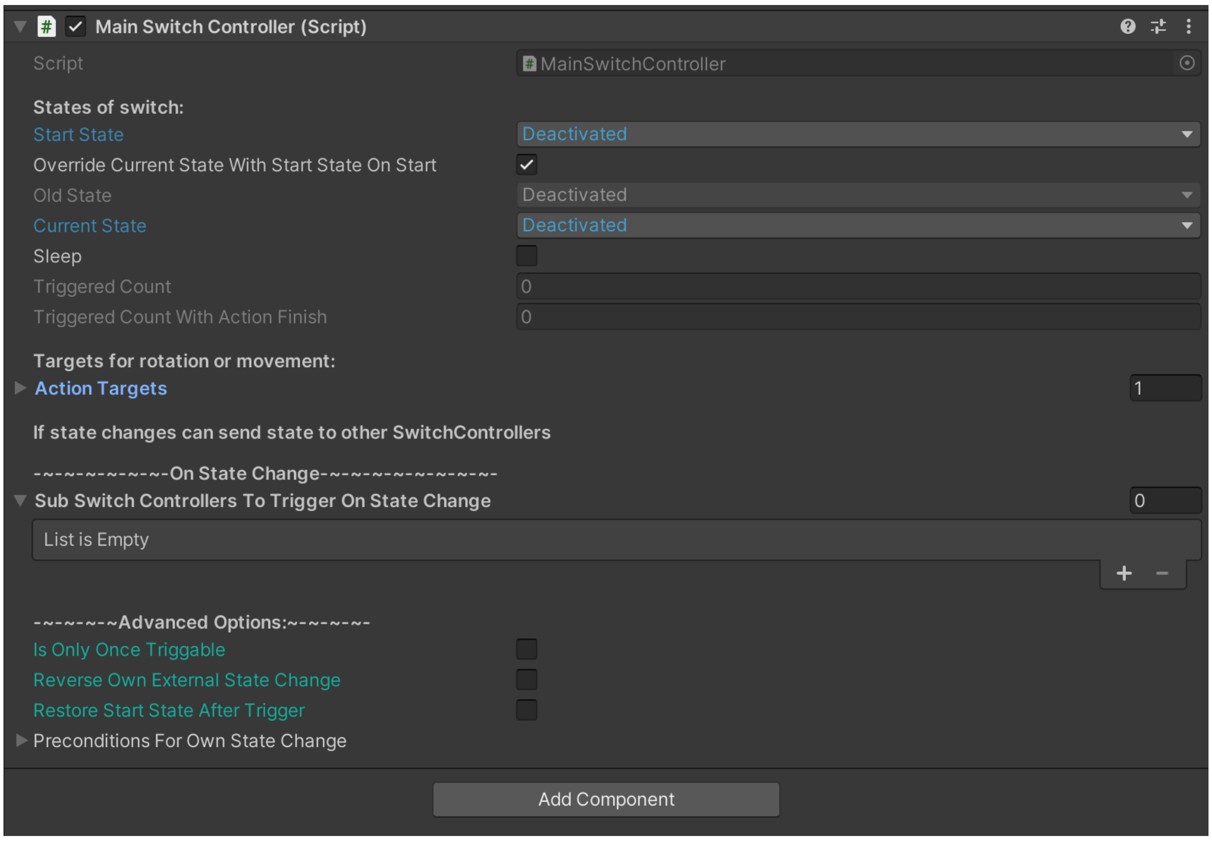
There are also a lot of public fields which you can access depending on the script.

4.1 MainSwitchController (Please read before others, reusing explanations)

Handles:

- movement between two positions.
- rotation from an angle to an angle. Can create swinging movement with a third angle. Supports negative rotations if you want to rotate in specific direction and there is no limit in rotation angle(You can rotate for example 3600° or -3600° 10 times around)
- SetActive (Enable/Disable) object. Note: Do not put the script for SetActive on the same object as the action target, because else it will disable the script on Deactivated state.
- adding sounds for each state and situation.

The concept is to have “ActionTarget” which should be moved rotated ot SetActive.
After the task is done depending on configuration the state of the SwitchController will change and notify “SubSwitchControllersToTriggerOnStateChange” if needed. You can add multiple “ActionTargets” and multiple “SubSwitchControllers”.


Main Switch Controller (Script)

Script: MainSwitchController

States of switch:

Start State: Deactivated
Override Current State With Start State On Start: ✓
Old State: Deactivated
Current State: Deactivated
Sleep: ☐
Triggered Count: 0
Triggered Count With Action Finish: 0

Targets for rotation or movement:

Action Targets: 1

If state changes can send state to other SwitchControllers

-~-~-~-~-~-~-On State Change-~-~-~-~-~-~-~-~-~-
Sub Switch Controllers To Trigger On State Change: 0
List is Empty

-~-~-~-~-~Advanced Options:~-~-~-~-~-
Is Only Once Triggable: ☐
Reverse Own External State Change: ☐
Restore Start State After Trigger: ☐
Preconditions For Own State Change

Add Component

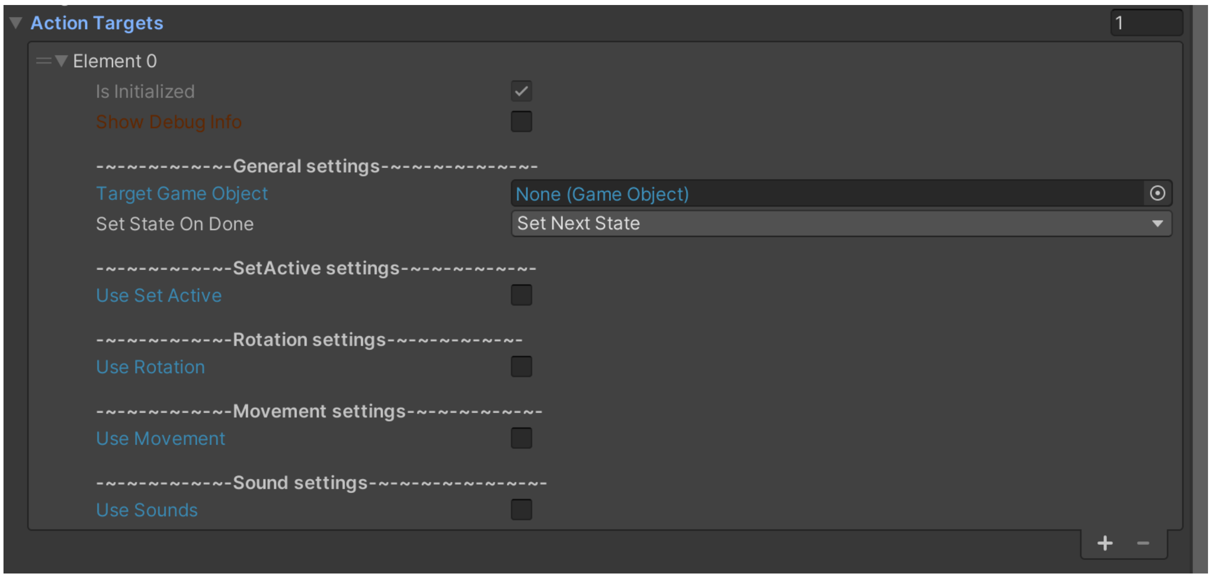
Field Definitions

StartState	State on load of the script. For example on MainSwitchController, on start will set rotation or position given in rotation and movement settings if set.
OverrideCurrentStateWithStartStateOnStart	Should the current state be instantly overloaded by the start state on start without executing state changing tasks?
OldState (ReadOnly)	Previous state.
CurrentState	Current state of SwitchController.
Sleep	Pauses all update actions. The state can still be set from outside.
TriggeredCount (ReadOnly)	How many state changing events had been received.
TriggeredCountWithActionFinish (ReadOnly)	Count of how many times triggered with successful finished action. Useful if for example a switch should trigger something if exactly 3 times triggered.
ActionTargets	Configuration for GameObjects to perform

	action on.
SubSwitchControllersToTriggerOnStateCha nge	Other GameObjects with SwitchControllers that are connected to the state change of this one. This is needed to create chaining events. If state of this SController changes it will trigger the configuration to change state on SubSwitchControllers(connected SwitchControllers)
IsOnlyOnceTriggable	Only one state changing event is allowed, others after will be blocked.
InvertOwnExternalStateChange	If activated the state changing attempts from outside will be inverted. Activating->Deactivating,Deactivating->Acti vating.
RestoreStartStateAfterTrigger	Automatically restores the start state after state changing action was received and finished.
PreconditionsForOwnStateChange	Conditions when to execute state changing actions or rewrite depending on other SControllers.

Action Targets

The Action Target contains definitions for tasks. The field “TargetGameObject” is any “GameObject” that should be moved for example. With “SetStateOnDone” you can let the task change the state only on specific conditions. For example if you want to rotate something but it should not be relevant for state change you can mute it.



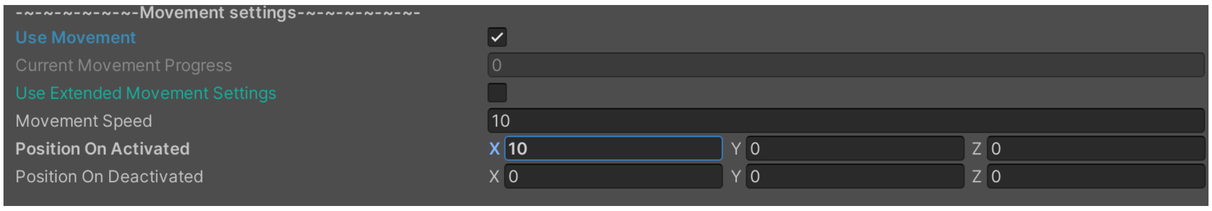
Action Targets 1
Element 0
Is Initialized ✓
Show Debug Info
-~-~-~-~-~-~-General settings-~-~-~-~-~-~-~-
Target Game Object: None (Game Object)
Set State On Done: Set Next State
-~-~-~-~-~-~-SetActive settings-~-~-~-~-~-~-
Use Set Active
-~-~-~-~-~-~-Rotation settings-~-~-~-~-~-~-
Use Rotation
-~-~-~-~-~-~-Movement settings-~-~-~-~-~-~-
Use Movement
-~-~-~-~-~-~-Sound settings-~-~-~-~-~-~-~-~-
Use Sounds

SetActive

To deactivate on Deactivated state or activate on Activated state the TargetGameObject mark “SetActive”. It disables other options.

Movement

For basic movement mark “UseMovement”.
For example, given the data on the following snippet, “PositionOnActivated” means that on “Activating” it will move to x:10,y:0,z:0 and switch controllers state to “Activated” if no extra tasks are given to wait for. On “Deactivating” the reverse to “.localPosition=Vector3(0,0,0)”.



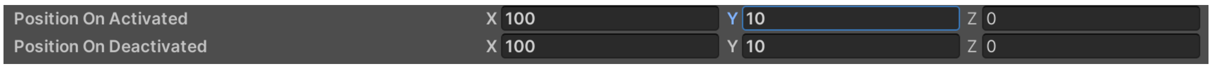
-~-~-~-~-~-~-Movement settings-~-~-~-~-~-~-

Use Movement	✓		
Current Movement Progress	0		
Use Extended Movement Settings			
Movement Speed	10		
Position On Activated	X 10	Y 0	Z 0
Position On Deactivated	X 0	Y 0	Z 0

Teleportation

To Teleport on any stateChange the TargetGameObject set PositionOnActivated=PositionOnDeactivated but not x:0y:0z:0.

For example:

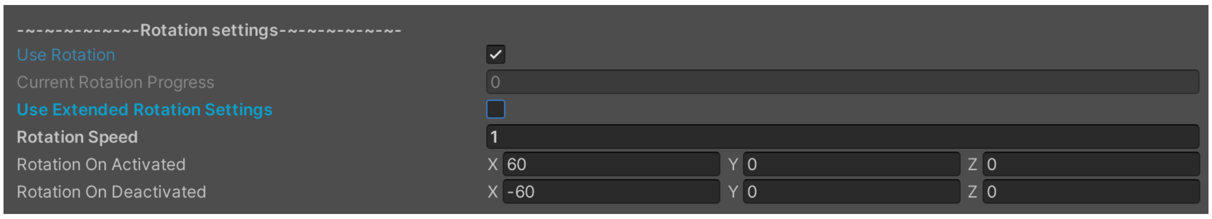


Position On Activated	X 100	Y 10	Z 0
Position On Deactivated	X 100	Y 10	Z 0

will teleport the TargetGameObject to “localPosition” x:100,y:10,z:0 if the state changes.

Rotation

For basic Rotation mark “UseRotation”.
The same game goes here. Set which rotation you want the TargetObject to have on each state. The Rotation direction can go in a positive or negative direction. The rotations can go over (+/-)360° degrees. By “ShowDebugInfo” check you can see what exactly is done with rotation chunks.

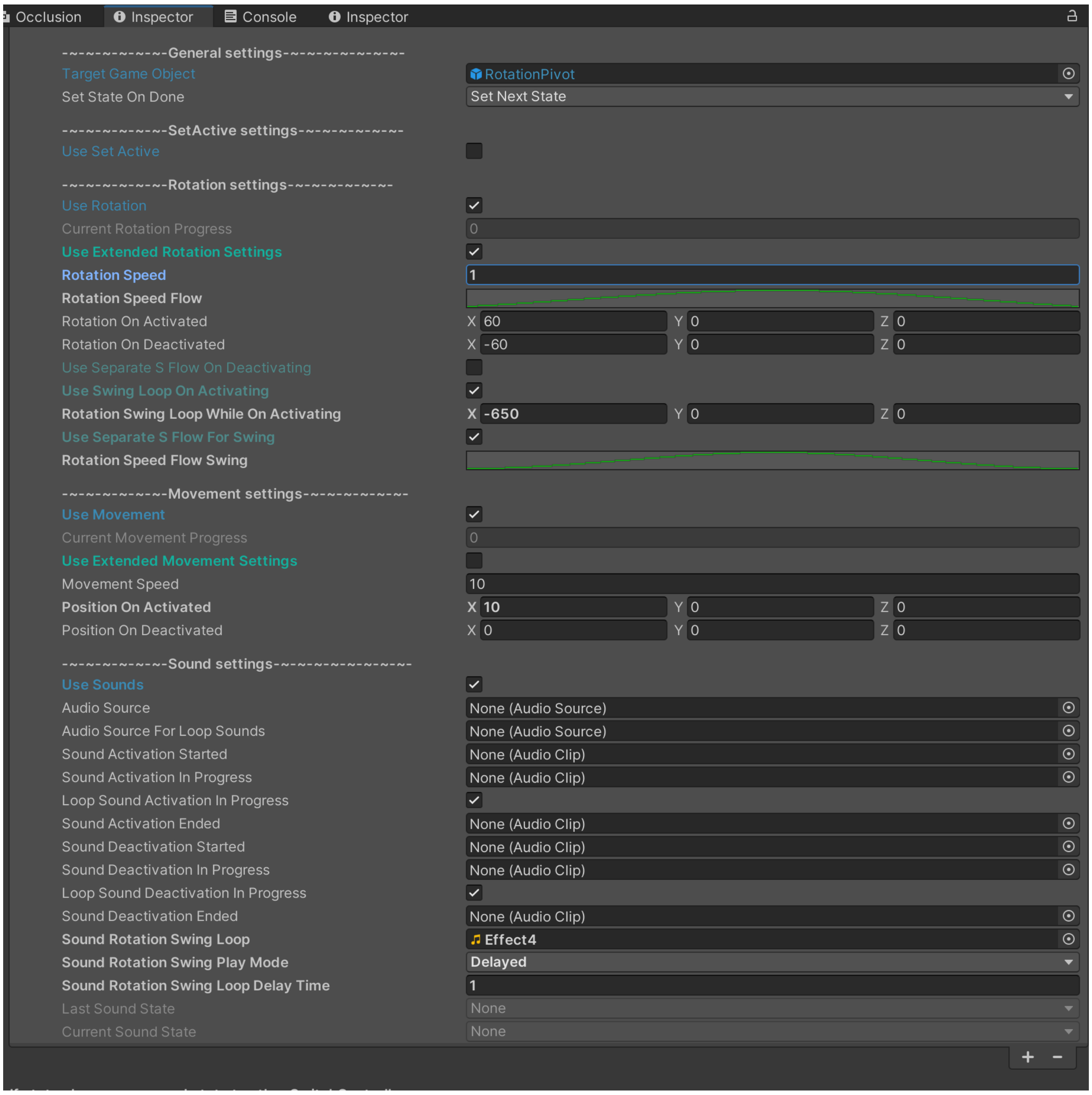


-~-~-~-~-~-~-Rotation settings-~-~-~-~-~-~-

Use Rotation	✓		
Current Rotation Progress	0		
Use Extended Rotation Settings			
Rotation Speed	1		
Rotation On Activated	X 60	Y 0	Z 0
Rotation On Deactivated	X -60	Y 0	Z 0

ExtendedOptions

By expanding “ExtendedOptions” you can do even more.



-~-~-~-~-~-~-General settings-~-~-~-~-~-~-~-

Target Game Object	RotationPivot
Set State On Done	Set Next State

-~-~-~-~-~-~-SetActive settings-~-~-~-~-~-~-

Use Set Active	☐

-~-~-~-~-~-~-Rotation settings-~-~-~-~-~-~-

Use Rotation	✓		
Current Rotation Progress	0		
Use Extended Rotation Settings	✓		
Rotation Speed	1		
Rotation Speed Flow			
Rotation On Activated	X 60	Y 0	Z 0
Rotation On Deactivated	X -60	Y 0	Z 0
Use Separate S Flow On Deactivating	☐		
Use Swing Loop On Activating	✓		
Rotation Swing Loop While On Activating	X -650	Y 0	Z 0
Use Separate S Flow For Swing	✓		
Rotation Speed Flow Swing			

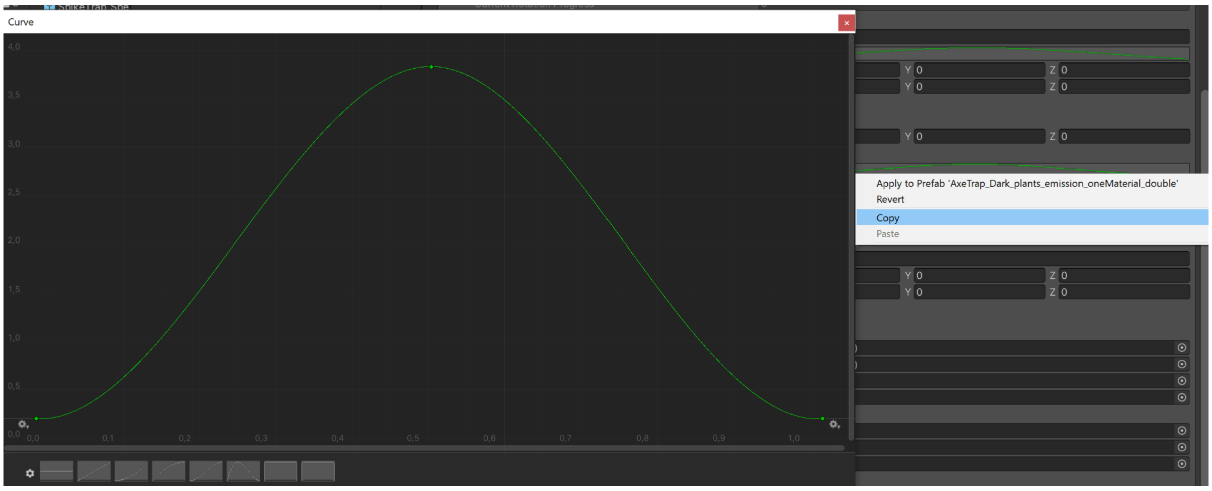
-~-~-~-~-~-~-Movement settings-~-~-~-~-~-~-

Use Movement	✓		
Current Movement Progress	0		
Use Extended Movement Settings	☐		
Movement Speed	10		
Position On Activated	X 10	Y 0	Z 0
Position On Deactivated	X 0	Y 0	Z 0

-~-~-~-~-~-~-Sound settings-~-~-~-~-~-~-~-~-

Use Sounds	✓
Audio Source	None (Audio Source)
Audio Source For Loop Sounds	None (Audio Source)
Sound Activation Started	None (Audio Clip)
Sound Activation In Progress	None (Audio Clip)
Loop Sound Activation In Progress	✓
Sound Activation Ended	None (Audio Clip)
Sound Deactivation Started	None (Audio Clip)
Sound Deactivation In Progress	None (Audio Clip)
Loop Sound Deactivation In Progress	✓
Sound Deactivation Ended	None (Audio Clip)
Sound Rotation Swing Loop	Effect4
Sound Rotation Swing Play Mode	Delayed
Sound Rotation Swing Loop Delay Time	1
Last Sound State	None
Current Sound State	None

For example, instead of using linear speed you can use “SpeedFlow”. This chart represents for X-axis the lifetime of a task from 0 to 1 (1=100% done) and on Y-axis the speed multiplier. By using it you can make for example that something moves faster or slower on certain progress.


Curve
Apply to Prefab 'AxeTrap_Dark_plants_emission_oneMaterial_double'
Revert
Copy
Paste

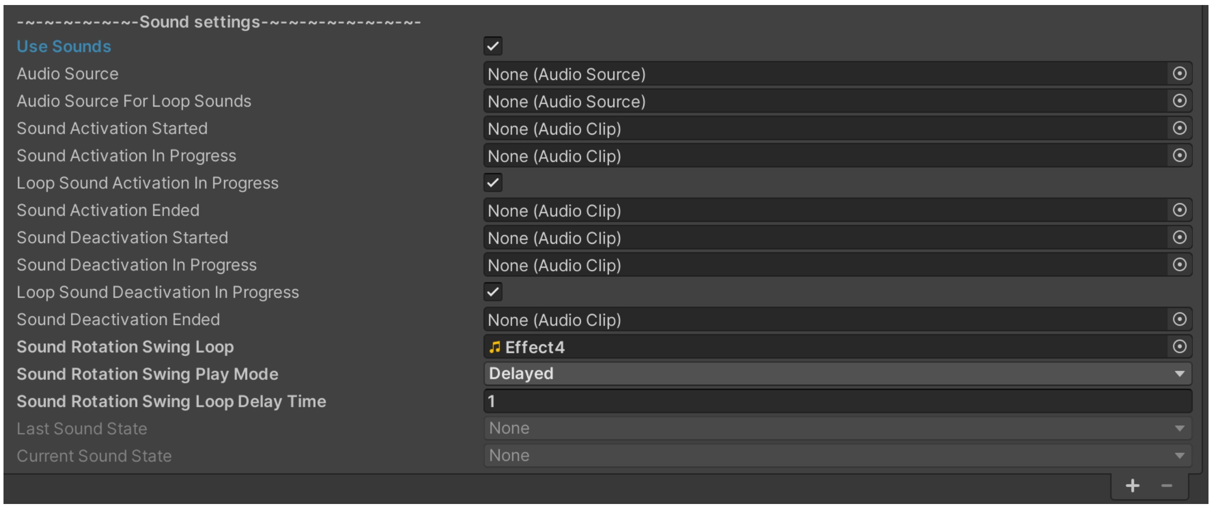
Rotation supports rotating of the “TargetGameObject” in any direction (-/+) and any Angle. If the delta angle is used over 360° it will spin over.

SwingLoop

If it's desired to create a swinging object, in rotation settings the “useSwingLoop” option can be used. It adds to “Activating” state another axis configuration. So on Activating the “TargetObject” rotates from OnDeactivated Rotation axis-> onActivatedRotationAxis and from there to SwingLoop and from there goes back to Activated axis repeating(SwingLoop->Activated, Activated->SwingLoop, SwingLoop->Activated…).

Sounds

You can add sounds for progress states of rotation and movement.



-~-~-~-~-~-~-~-Sound settings-~-~-~-~-~-~-~-~-~-~-

Use Sounds	✓
Audio Source	None (Audio Source)
Audio Source For Loop Sounds	None (Audio Source)
Sound Activation Started	None (Audio Clip)
Sound Activation In Progress	None (Audio Clip)
Loop Sound Activation In Progress	✓
Sound Activation Ended	None (Audio Clip)
Sound Deactivation Started	None (Audio Clip)
Sound Deactivation In Progress	None (Audio Clip)
Loop Sound Deactivation In Progress	✓
Sound Deactivation Ended	None (Audio Clip)
Sound Rotation Swing Loop	**Effect4**
Sound Rotation Swing Play Mode	**Delayed**
Sound Rotation Swing Loop Delay Time	1
Last Sound State	None
Current Sound State	None

If it's desired to use looping repeating sounds for progress mark “Loop” checkbox. SwitchConroller automatically generates AudioSources for “OneShot” and “Looping” sounds, if the component has Audio source it will be used or copied. So if extended control over sound is needed please add AudioSource on the TargetObject with custom configuration.

Swing Loop sounds have some extra sound options to repeat the sounds delayed, progress based or by direction change. More options for sounds will be added in later releases to other configurations.

Field Definitions

ActionTargets

ShowDebugInfo	Show the debug info, for example rotation chunks.
TargetGameObject	GameObject on which to perform desired action.
SetStateOnDone	When the action of the current GameObject is finished, should it set the desired next state of the SwitchController? SetNextState: Set the desired next state. DoNotSetNextState: Action of this object is not relevant for state change. SetNextStateCombined: Only if other ActionTargets with the same state are finished as well, the desired next state will be set.
IsDone	The action of the current GameObject was finished.
UseSetActive	On ‘Activated’ will enable GameObject, on ‘Deactivated’ disable.
UseRotation	Rotate the target object?
CurrentRotationProgress(ReadOnly)	Progress in percent of the delta rotation.
UseExtendedRotationSettings	Enable extended settings.
RotationSpeed	Speed of rotation steps.
RotationSpeedFlow	Chart for more control of speed. X-Axis: from 0-1 lifetime of action, Y-Axis: Speed multiplier.
InternalRotationModel(Debug only)	Shows extended data of rotation steps.
InternalPreviousRotationModel(Debug only)	Shows extended data of previous rotation steps.
RotationOnActivated	Rotation angles which should be reached on Activated state.

RotationOnDeactivated	Rotation angles which should be reached in the Deactivated state.
UseSeparateSFlowOnDeactivating	Separate settings for Deactivating state, if not used, will use RotationSpeedFlow.
RotationSpeedFlowOnDeactivating	Chart for more control of speed. X-Axis: from 0-1 lifetime of action, Y-Axis: Speed multiplier.
UseSwingLoopOnActivating	Should third angle be used with constant bidirectional rotation from ‘Activated’ axis to third axis. If it's desired to create a swinging object, in rotation settings the “useSwingLoop” option can be used. It adds to “Activating” state another axis configuration. So on Activating the “TargetObject” rotates from OnDeactivated Rotation axis-> onActivatedRotationAxis and from there to SwingLoop and from there goes back to Activated axis repeating(SwingLoop->Activated, Activated->SwingLoop, SwingLoop->Activated…).
RotationSwingLoopWhileOnActivating	Rotation angles which should go from Activated angles on Activating state to this angles and reverse.
UseSeparateSFlowForSwing	Separate setting for SwingLoop, if not used, will use RotationSpeedFlow.
RotationSpeedFlowSwing	Chart for more control of speed. X-Axis: from 0-1 lifetime of action, Y-Axis: Speed multiplier.

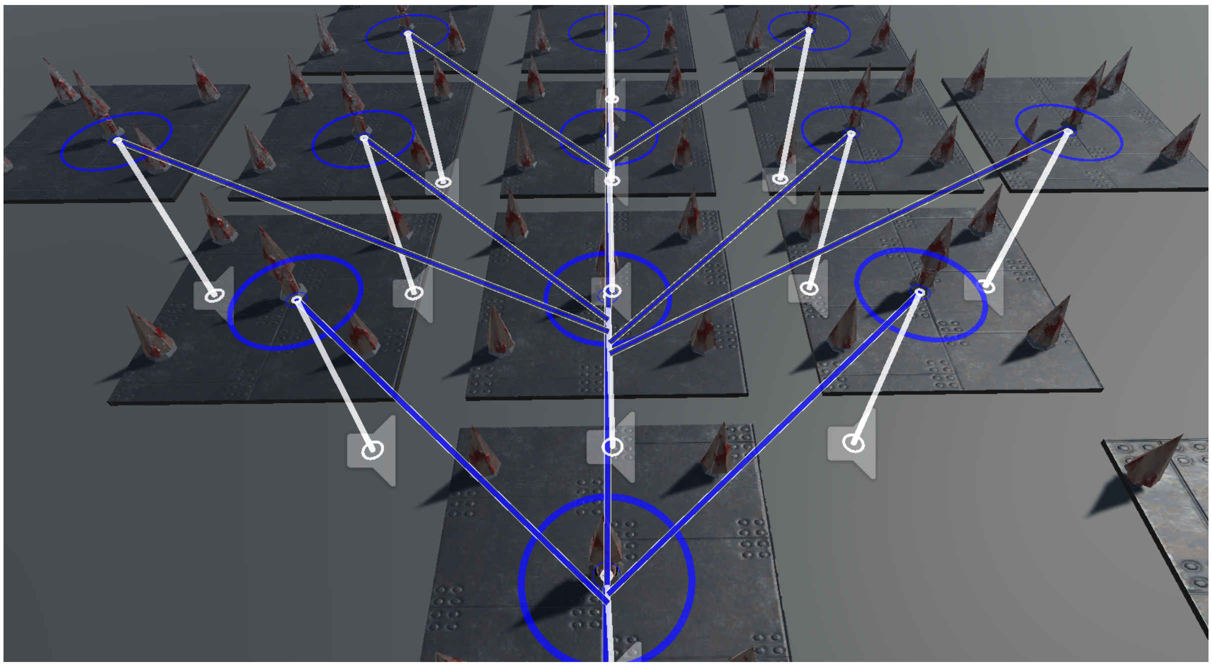
UseMovement	Move the target object between two ‘localPositions’?
CurrentMovementProgress	Progress in percent of the delta movement.
UseExtendedMovementSettings	Enable extended settings.
MovementSpeed	Speed of movement steps.
PositionOnActivated	localPosition which should be reached on Activated state.
PositionOnDeactivated	localPosition which should be reached on

	Deactivated state.
MovementSpeedFlow	Chart for more control of speed. X-Axis: from 0-1 lifetime of action, Y-Axis: Speed multiplier.
UseSeparateMSFlowOnDeactivating	Separate settings for Deactivating state, if not used, will use MovementSpeedFlow.
MovementSpeedFlowOnDeactivating	Chart for more control of speed. X-Axis: from 0-1 lifetime of action, Y-Axis: Speed multiplier.
UseSounds	Should sounds be used?
AudioSource	Audio source for ‘OneShot’ sounds. If not set, will be created automatically on the target GameObject. For more control of audio settings, create one and configure it.
AudioSourceForLoopSounds	Audio source for ‘Looping’ sounds. If not set, will be copied automatically from ‘AudioSource’ to the target GameObject. If ‘AudioSource is not set, it will be created automatically.’ For more control of audio settings, create one and configure it.
SoundActivationStarted	AudioClip to play on Activating state beginning.
SoundActivationInProgress	Second audioClip to play on Activating state beginning, supports looping.
LoopSoundActivationInProgress	AudioClip should be looping.
SoundActivationEnded	AudioClip to play on Activating state finish.
SoundDeactivationStarted	AudioClip to play on Deactivating state beginning.
SoundDeactivationInProgress	Second audioClip to play on Deactivating state beginning, supports looping.
LoopSoundDeactivationInProgress	AudioClip should be looping?
SoundDeactivationEnded	AudioClip to play on Deactivating state finish.
SoundRotationSwingLoop	AudioClip to play on swing loop action.
SoundRotationSwingPlayMode	PlayAtDirectionSwitch: Play sound each time SwingLoop changes its direction. Delayed: Play with delay RepeatPercentPassed: Play the sound

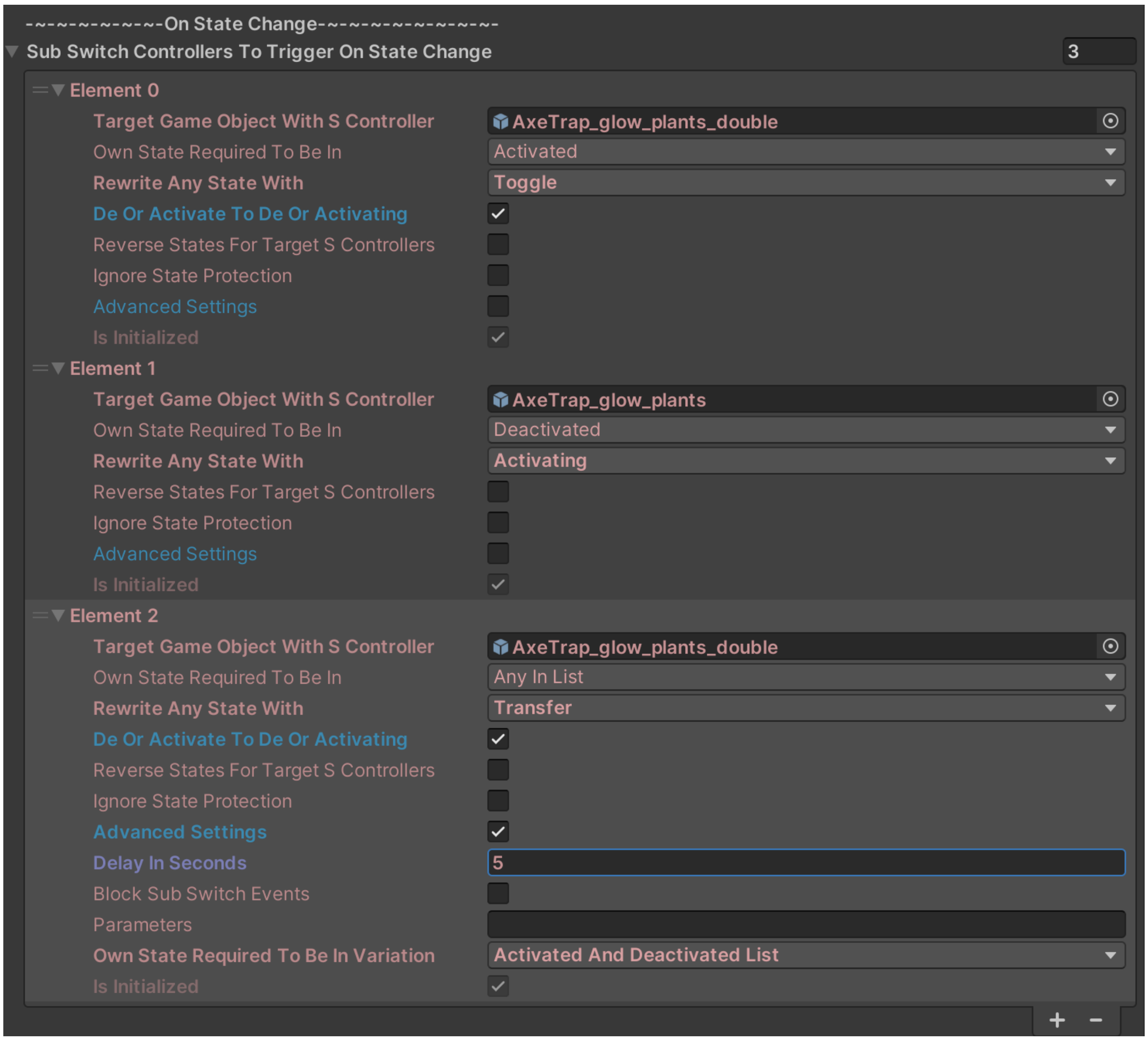
	cumulative percent passed. For example 5-> 5%,10%,15%..
SoundRotationSwingLoopDelayTime	Delay time in seconds
RepeatForPercentPassed	Repeat each percent
LastSoundState(ReadOnly)	Internal last sound state
CurrentSoundState(ReadOnly)	Internal current sound state

SubSwitchControllers

When operation has finished on ActionTarget after the state is set to new one its possible to send state changing events to other objects that have SwitchController attached to them. This way it's easily possible to create chaining events. For example some switch has been triggered, the door is opening, after that ground starts to move when its finished, trap disappears.



This is done by adding “SubSwitchControllers” to the “SubSwitchControllerToTriggerOnStateChange” list. Is possible to delegate state only if the owner SwitchController has a specific state or/and rewrite the current state to another one for SubSwitchController.



-~-~-~-~-~-~-On State Change-~-~-~-~-~-~-~-~-

Sub Switch Controllers To Trigger On State Change: 3

Element 0
- Target Game Object With S Controller: AxeTrap_glow_plants_double
- Own State Required To Be In: Activated
- Rewrite Any State With: Toggle
- De Or Activate To De Or Activating: ✓
- Reverse States For Target S Controllers: ☐
- Ignore State Protection: ☐
- Advanced Settings: ☐
- Is Initialized: ✓

Element 1
- Target Game Object With S Controller: AxeTrap_glow_plants
- Own State Required To Be In: Deactivated
- Rewrite Any State With: Activating
- Reverse States For Target S Controllers: ☐
- Ignore State Protection: ☐
- Advanced Settings: ☐
- Is Initialized: ✓

Element 2
- Target Game Object With S Controller: AxeTrap_glow_plants_double
- Own State Required To Be In: Any In List
- Rewrite Any State With: Transfer
- De Or Activate To De Or Activating: ✓
- Reverse States For Target S Controllers: ☐
- Ignore State Protection: ☐
- Advanced Settings: ✓
- Delay In Seconds: 5
- Block Sub Switch Events: ☐
- Parameters:
- Own State Required To Be In Variation: Activated And Deactivated List
- Is Initialized: ✓

For example ObjA has finished, Set its own state from Activating to Activated, then iterate through SubSwitchControllers in the list and set Activating on them. Other SwitchControllers will get state Activating and do actions. Switch stepped on->moving itself to ground->finished->open door1, open door2.

Field Definitions

SubSwitchControllersToTriggerOnStateChange

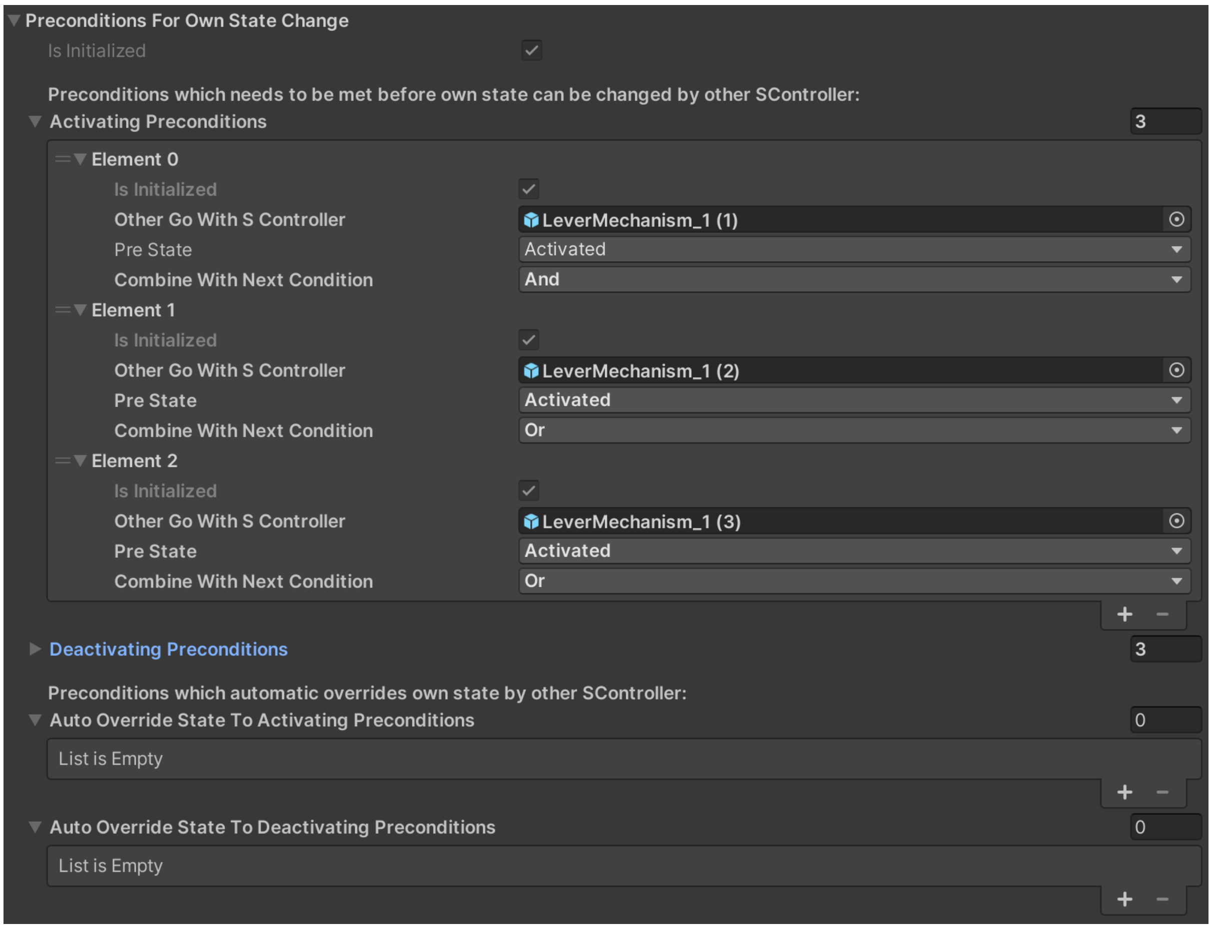
TargetGameObjectWithSController	Other GameObjects with SwitchControllers to set new state on.
OwnStateRequiredToBeIn	Precondition needed its own state to set state on target.
RewriteAnyStateWith	Transfer: Sends its own state to target. Toggle: Sets target state to opposite state. Others: Set the given state.
DeOrActivateToDeOrActivating	If the own state is in Activated, set it to

	Activating on target, same for Deactivated->Deactivating.
InvertStatesForTargetSControllers	Activating->Deactivating, Deactivating->Activating. Same can be done with using rules with ‘RewriteAnyStateWith’
IgnoreStateProtection	Disables preconditions on target SwitchController
AdvancedSettings	Enables advanced settings
DelayInSeconds	Set a new state with delay after given seconds.
BlockSubSwitchEvents	Set state on target SwitchController but block it from changing the state on its SubSwitchControllers when actions on it are done.
Parameters	Parameters are string combinations which can be sent to give extra options. You can add your own parameters on your custom SwitchControllers inside OnSetStateByCaller method. Parameters can be combinated ’paramOne, paramTwo’ Current build in parameters: —--- MainSwitchController: ‘Ignorestatechangenow:true’ : ignores the given state change ‘sleep:true’/’sleep:false’ : sets SwitchController to sleep —--- NodeWalkerSwitchController: ‘Ignorestatechangenow:true’ : ignores the given state change ‘sleep:true’/’sleep:false’ : sets SwitchController to sleep ‘nodewalker:reversedir’ :reverse movement direction —---

OwnStateRequiredToBeInVariation	Change this flag only in your custom SwitchControllers depending on your implementation, it changes the options in the 'OwnStateRequiredToBeIn' dropdown.

Preconditions

Sometimes it is needed to create preconditions to block state change of SwitchController or rewrite state by specific constellation. For example Ian leaver can be only activated if other leavers are set with specific states. To block the leaver from activating preconditions can be used.


Preconditions For Own State Change
Is Initialized ✓

Preconditions which needs to be met before own state can be changed by other SController:
Activating Preconditions 3

Element 0
Is Initialized ✓
Other Go With S Controller: LeverMechanism_1 (1)
Pre State: Activated
Combine With Next Condition: And

Element 1
Is Initialized ✓
Other Go With S Controller: LeverMechanism_1 (2)
Pre State: Activated
Combine With Next Condition: Or

Element 2
Is Initialized ✓
Other Go With S Controller: LeverMechanism_1 (3)
Pre State: Activated
Combine With Next Condition: Or

Deactivating Preconditions 3

Preconditions which automatic overrides own state by other SController:
Auto Override State To Activating Preconditions 0
List is Empty

Auto Override State To Deactivating Preconditions 0
List is Empty

The preconditions are to read from top to down. “CombineWithNextCondition” combines expression with the next condition if it's set to “And”. In Case of “Or” it's acting on its own without combining to the next expression.

“... Preconditions”: Preconditions which needs to be met before own state can be changed by another SController

“AutoOverrideState..”: Automatically sets the SwitchControllers state to desired if targets change. It ignores the internal state check.

Field Definitions

Precondition

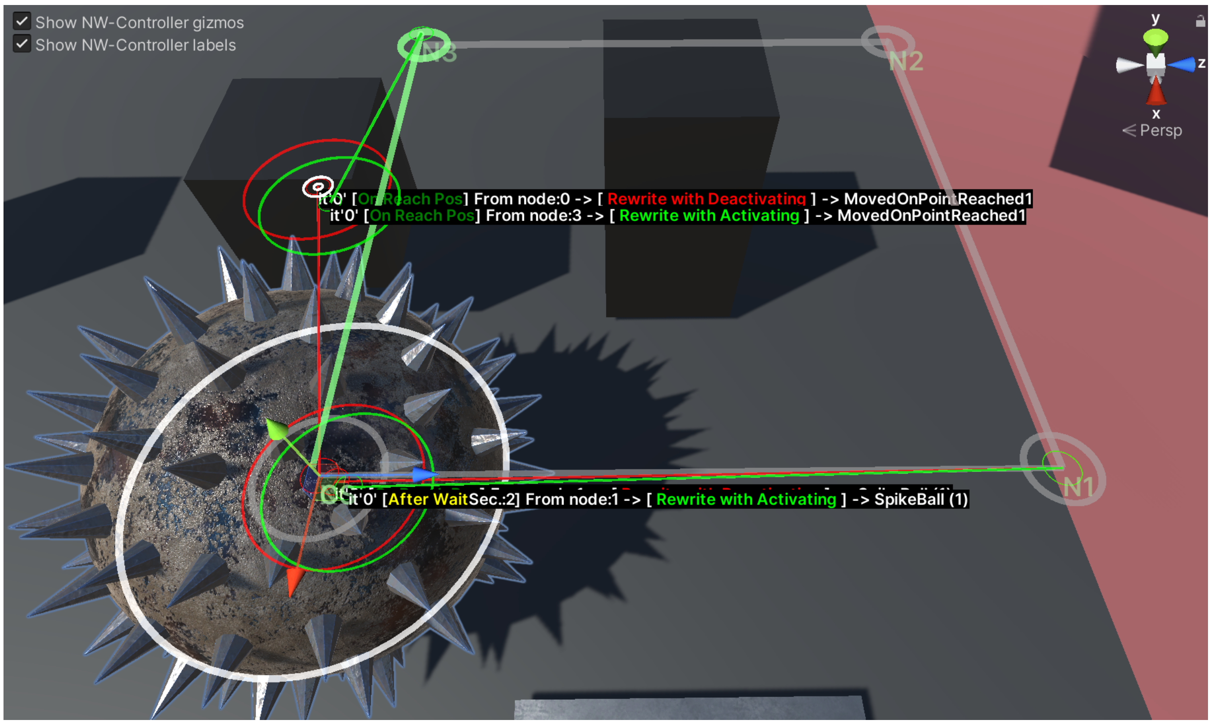
OtherGoWithSController	Depending on the state of given GameObjects SwitchController.
PreState	Given objects state needs to be in.
CombineWithNextCondition	‘And’ combines with the next condition, ‘Or’ works as a standalone condition.

4.2 NodeWalkerSwitchController

Handles:

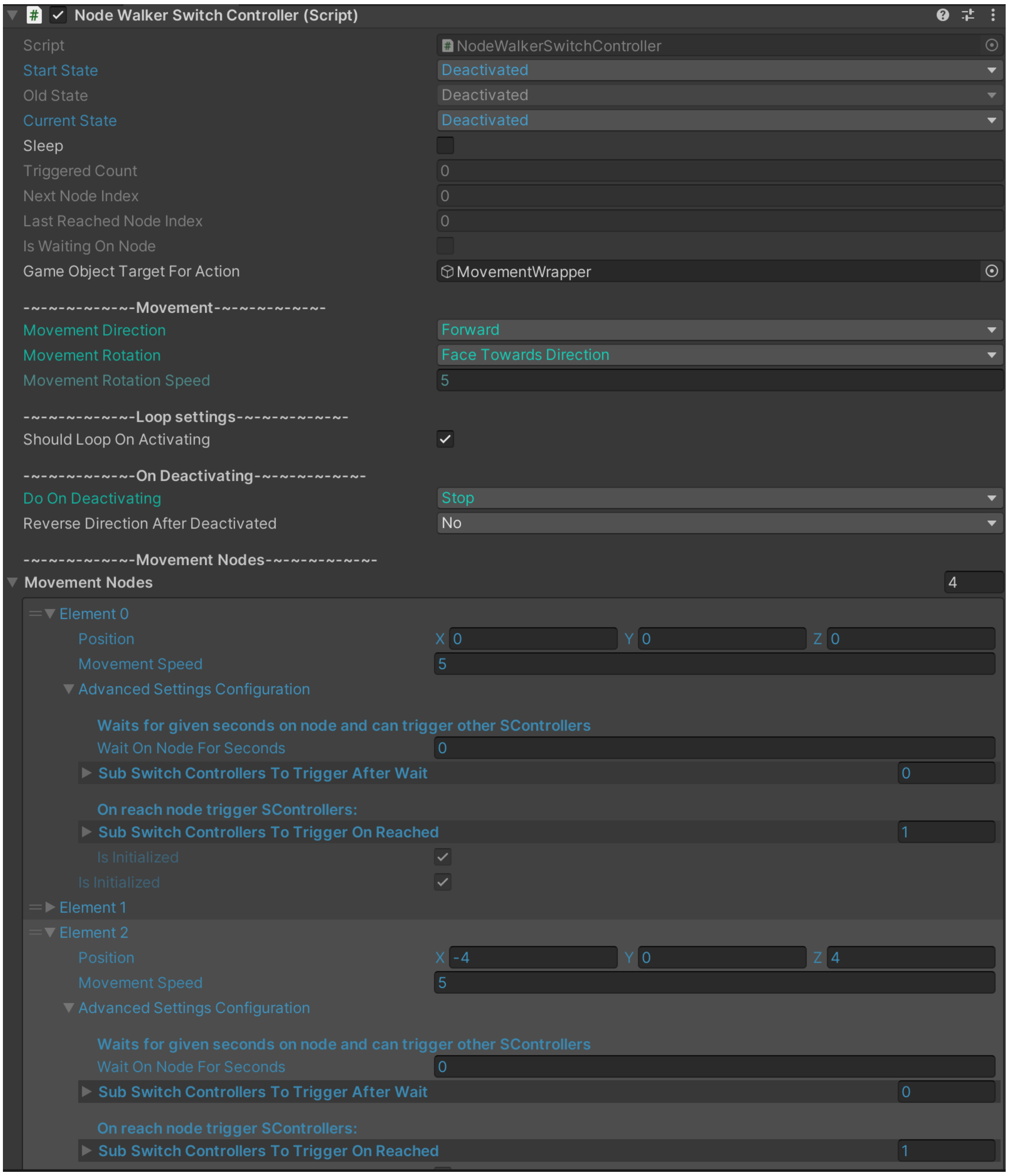
- Movement between multiple positions(nodes).
- Looping on default on the nodes.
- Can stop and trigger other SwitchControllers passing on nodes.
- Moves on ‘Activating’ state, custom configuration on Deactivating state.

With MainSwitchController you can set movement between two positions but if you want to create something that can run between multiple positions or loop on them the NodeWalkerSwitchController can be used. Do not use MainSwitchController with movement options and NodeWalker on the same TargetGameObject they will conflict.



```
it'0' [On Reach Pos] From node:0 -> [ Rewrite with Deactivating ] -> MovedOnPointReached1
it'0' [On Reach Pos] From node:3 -> [ Rewrite with Activating ] -> MovedOnPointReached1
it'0' [After WaitSec.:2] From node:1 -> [ Rewrite with Activating ] -> SpikeBall (1)
```

NodeWalkerSwitchController uses part of the same settings as MainSwitchController. Please refer to the missing pieces to the explanation above!


Node Walker Switch Controller (Script)

```
Script                                  NodeWalkerSwitchController
Start State                             Deactivated
Old State                               Deactivated
Current State                           Deactivated
Sleep                                   ☐
Triggered Count                         0
Next Node Index                         0
Last Reached Node Index                 0
Is Waiting On Node                      ☐
Game Object Target For Action           MovementWrapper

-~-~-~-~-~-~-Movement-~-~-~-~-~-~-
Movement Direction                      Forward
Movement Rotation                       Face Towards Direction
Movement Rotation Speed                 5

-~-~-~-~-~-~-Loop settings-~-~-~-~-~-~-
Should Loop On Activating               ✓

-~-~-~-~-~-~-On Deactivating-~-~-~-~-~-~-
Do On Deactivating                      Stop
Reverse Direction After Deactivated     No

-~-~-~-~-~-~-Movement Nodes-~-~-~-~-~-~-
Movement Nodes                                                          4
  Element 0
    Position                            X 0        Y 0        Z 0
    Movement Speed                      5
    Advanced Settings Configuration
      Waits for given seconds on node and can trigger other SControllers
      Wait On Node For Seconds          0
      Sub Switch Controllers To Trigger After Wait                      0
      On reach node trigger SControllers:
      Sub Switch Controllers To Trigger On Reached                      1
      Is Initialized                    ✓
    Is Initialized                      ✓
  Element 1
  Element 2
    Position                            X -4       Y 0        Z 4
    Movement Speed                      5
    Advanced Settings Configuration
      Waits for given seconds on node and can trigger other SControllers
      Wait On Node For Seconds          0
      Sub Switch Controllers To Trigger After Wait                      0
      On reach node trigger SControllers:
      Sub Switch Controllers To Trigger On Reached                      1
```

Unlike the MainSwitchController you cannot add a list of ActionTargets, you are moving around only one object which is TargetForAction. It's best practice to add the target object into a parent object and set its localPosition to 0. then you can move it around and add movement nodes where you want the object to move. On Activated state the object will move to the nodes. In AdvancedSettingsConfiguration you can set WaitOnNodeForSeconds time if you want the object to wait on the Node. you can also trigger other SubSwitchControllers when you reach the Node or after waiting time has finished.

It's possible to set movement rotation to the direction of which the object is facing when it reaches the nodes. In the next updates I will add more functions for rotation on Nodes.

you can also define which actions the object should take when it is getting into state of Deactivating.

NodeWalkerSwitchController

For missing fields, see above 'MainSubSwitchController' explanation	
NextNodeIndex	Node index, where TargetObject is moving to.
LastReachedNodeIndex	Last reached Node index.
IsWaitingOnNode	Is the targetObject waiting on Node currently?
GameObjectTargetForAction	TargetObject to move on nodes.
MovementDirection	Movement direction of TargetObject.
MovementRotation	Movement rotation option.
MovementRotationSpeed	Speed of movement rotation.
ShouldLoopOnActivating	Should move in a loop, from last node to first node when reached?
DoOnDestinationReached	Options for actions to take when reaching the last node in a non looping context.
DoOnDeactivating	Which actions should be taken when the NodeWalker state is state to Deactivating?
DeactivatedDestinationNodeIndex	Node index for DoOnDeactivating action.
ReverseDirectionAfterDeactivated	Should the direction be reversed/inverted after the Deactivated state has been set.
MovementNodes	List of nodes to move to with configurations.

MovementNodes

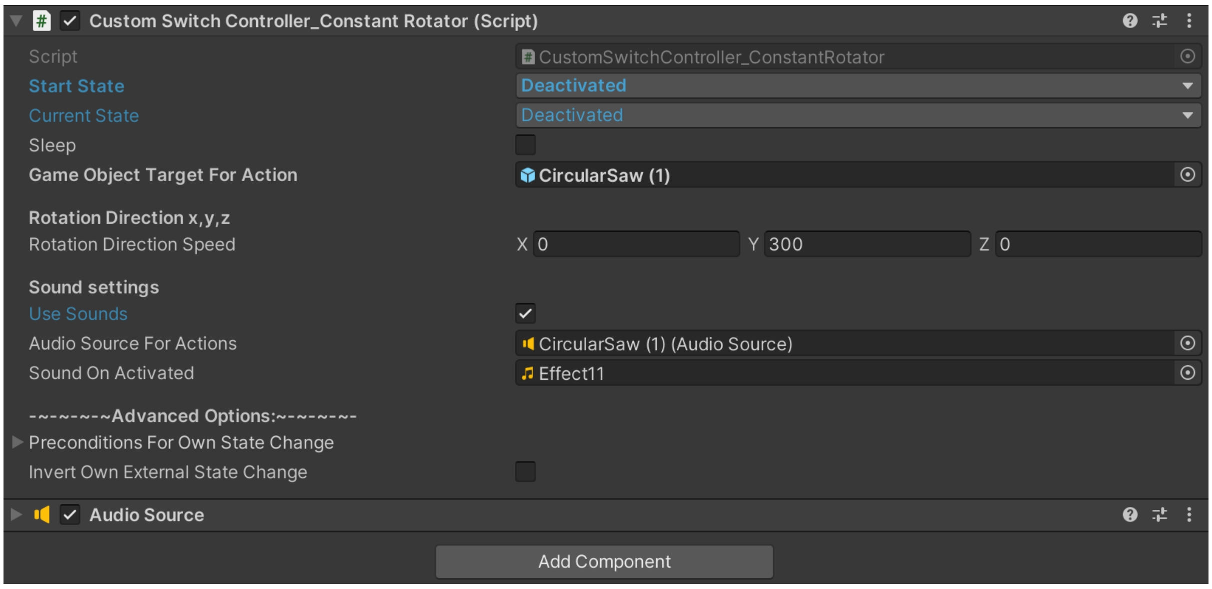
Position	Position of node to where the TargetObject should move to.
MovementSpeed	Speed of the movement. (Will add SpeedFlow chart next updates)
AdvancedSettingsConfiguration	Expand advanced settings.
WaitOnNodeForSeconds	Wait an amount of seconds on the node and then keep moving.

SubSwitchControllersToTriggerAfterWait	After waiting has finished on the node, set state on given SubSwitchControllers. see above 'MainSubSwitchController' explanation
SubSwitchControllersToTriggerOnReached	When reached the node, set state on given SubSwitchControllers. see above 'MainSubSwitchController' explanation

4.3 CustomSwitchController_ConstantRotator

Handles:
- Constant rotation of TargetObject by given speed on an axis.

Simple SwitchController that is used to constantly rotate objects.


Custom Switch Controller_Constant Rotator (Script)

Script: CustomSwitchController_ConstantRotator
Start State: Deactivated
Current State: Deactivated
Sleep: ☐
Game Object Target For Action: CircularSaw (1)

Rotation Direction x,y,z
Rotation Direction Speed: X 0 Y 300 Z 0

Sound settings
Use Sounds: ✓
Audio Source For Actions: CircularSaw (1) (Audio Source)
Sound On Activated: Effect11

-~-~-~-~Advanced Options:~-~-~-~-
Preconditions For Own State Change
Invert Own External State Change: ☐

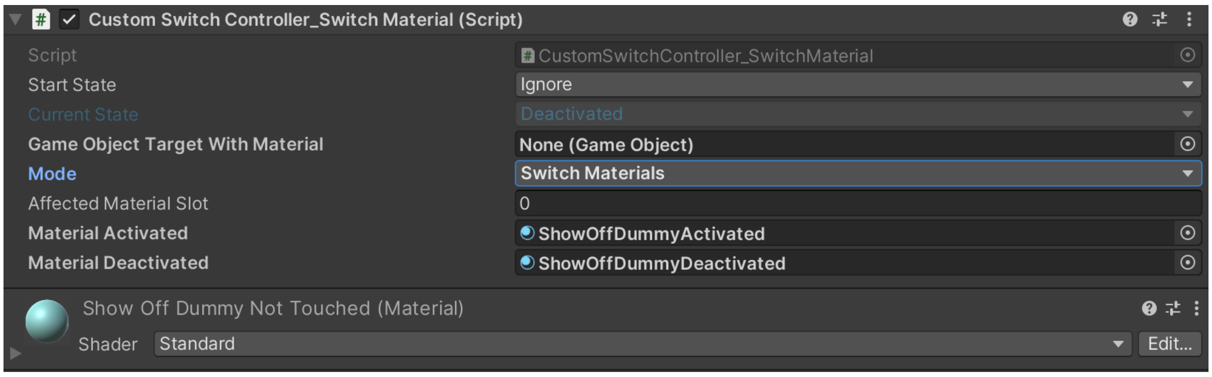
For missing fields, see above 'MainSubSwitchController' explanation	
GameObjectTargetForAction	TargetObject to rotate.
RotationDirectionSpeed	Rotation speed for each axis to rotate. Use -x, -y,-z to Rotate opposite.

4.4 CustomSwitchController_SwitchMaterial

Handles:

- Switches/Replaces materials by given material slot/index.
- Sets material by given material slot/index.

By given materials this SwitchController can depending on the state switch/replace materials.



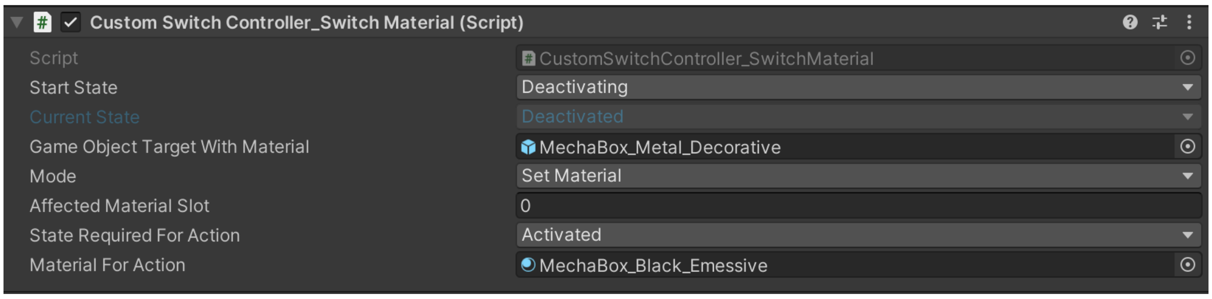
Custom Switch Controller_Switch Material (Script)

Script	CustomSwitchController_SwitchMaterial
Start State	Ignore
Current State	Deactivated
Game Object Target With Material	None (Game Object)
Mode	Switch Materials
Affected Material Slot	0
Material Activated	ShowOffDummyActivated
Material Deactivated	ShowOffDummyDeactivated

Show Off Dummy Not Touched (Material)

Shader Standard Edit...

Or it can set a material by index. If the renderer has less materials, the list will be increased.



Custom Switch Controller_Switch Material (Script)

Script	CustomSwitchController_SwitchMaterial
Start State	Deactivating
Current State	Deactivated
Game Object Target With Material	MechaBox_Metal_Decorative
Mode	Set Material
Affected Material Slot	0
State Required For Action	Activated
Material For Action	MechaBox_Black_Emessive

For missing fields, see above 'MainSubSwitchController' explanation	
GameObjectTargetWithMaterial	Target GameObject to switch materials on.
Mode	SwitchMaterials: Switch one material to another one and back depending on state. SetMaterial: Set material to desired on Activated or Deactivated.
AffectedMaterialSlot	Affected material slot/index of the mesh renderer.
StateRequiredForAction	On which state to perform the desired action.

MaterialActivated	Material to set on Activated state.
MaterialDeactivated	Material to set on Deactivated state.
MaterialForAction	Material to set on action.

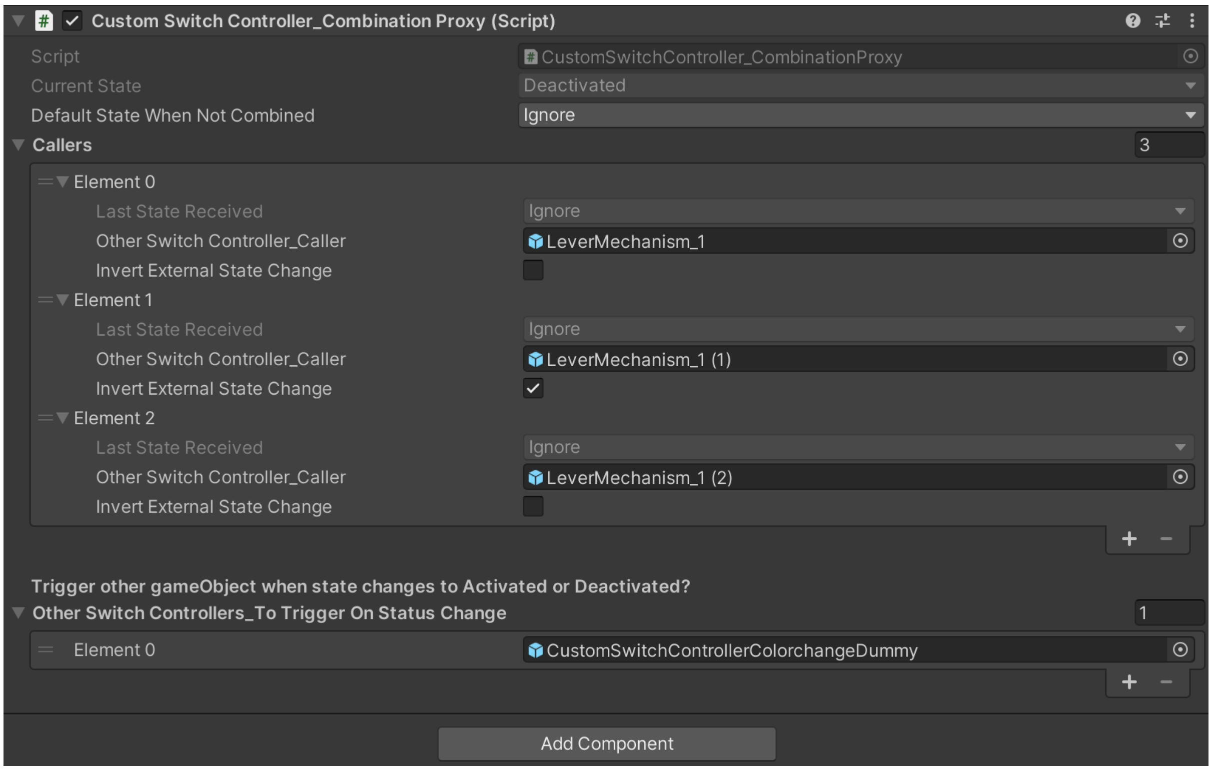
4.5 CustomSwitchController_CombinationProxy

Handles:

- Simple combine of states on multiple SwitchControllers to set other SubSwitchControllers depending on them.

When all callers match the same state ‘Activated’ (or by inverting with ‘InvertExternalStateChange’ -> Deactivated) the SubSwitchControllers will set to Activating. The same or even more advanced effect can be reached by using preconditions on MainSwitchController and NodeWalker! It's just a weaker all-around version of that, which can be used with any SwitchController.

By setting DefaultStateWhenNotCombined to something else than ‘ignore’ it will emit to SubSwitchControllers chosen state if the caller SwitchControllers does not match same combination.

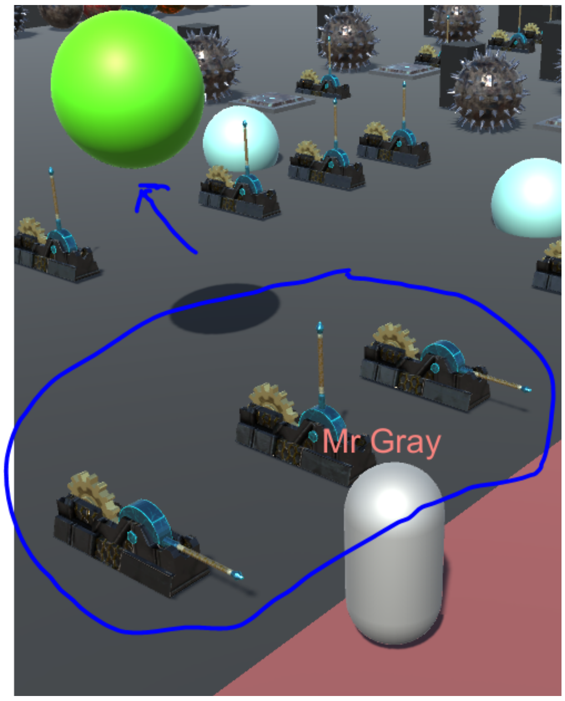

Custom Switch Controller_Combination Proxy (Script)

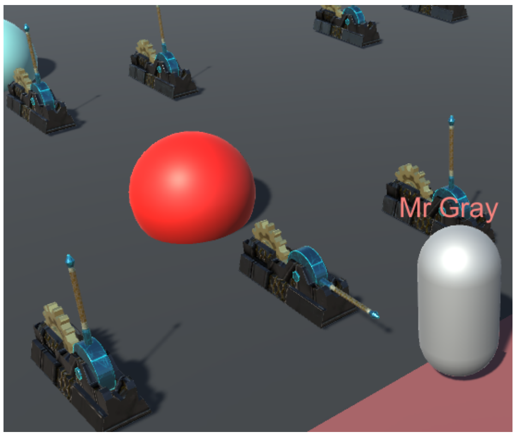
Script	CustomSwitchController_CombinationProxy
Current State	Deactivated
Default State When Not Combined	Ignore
Callers	3
Element 0	
Last State Received	Ignore
Other Switch Controller_Caller	LeverMechanism_1
Invert External State Change	
Element 1	
Last State Received	Ignore
Other Switch Controller_Caller	LeverMechanism_1 (1)
Invert External State Change	✓
Element 2	
Last State Received	Ignore
Other Switch Controller_Caller	LeverMechanism_1 (2)
Invert External State Change	

Trigger other gameObject when state changes to Activated or Deactivated?

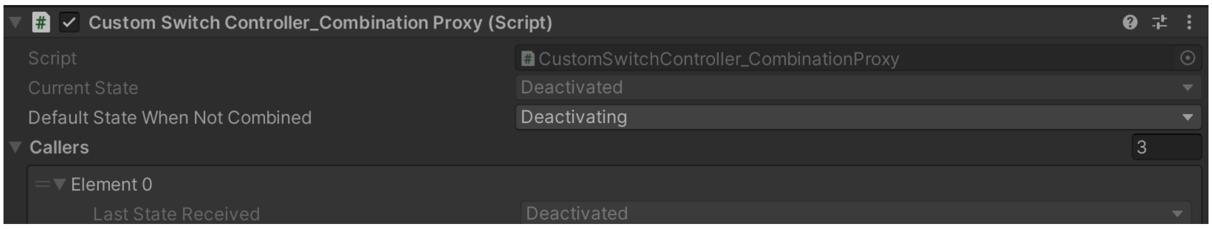
Other Switch Controllers_To Trigger On Status Change	1
Element 0	CustomSwitchControllerColorchangeDummy

On Example screenshot you can see the following setting: If caller element0 and 2 is set to Activated and element1(inverted) to Deactivated, the SubSwitch-Dummy will be set to Activating. If the Caller constellation is opposite, the subSwitchController will be set to Deactivating.

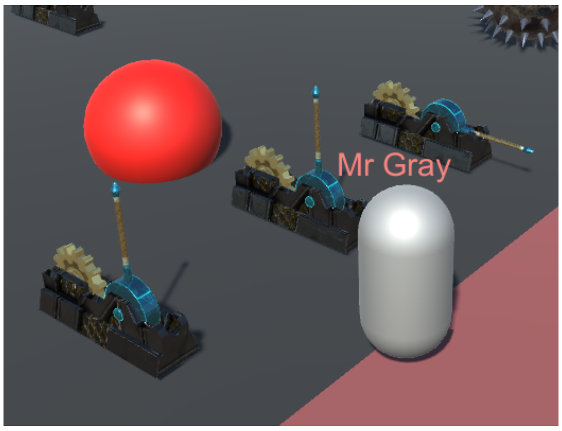




By using DefaultStateWhenNotCombined -> Deactivated for example it needs only one wrong caller to set the SubSwitchControllers to Deactivated state.


Custom Switch Controller_Combination Proxy (Script)

Script	CustomSwitchController_CombinationProxy
Current State	Deactivated
Default State When Not Combined	Deactivating
Callers	3
Element 0	
Last State Received	Deactivated



CurrentState	Current state of the SwitchController
DefaultStateWhenNotCombined	When the caller combination is not in same state (or by inverting) this default state will be set.

Callers	
OtherSwitchControllers_ToTriggerOnStatus Change	SubSwitchControllers to set state when this controller's state changes. Simpler version of previously explained field.

Callers

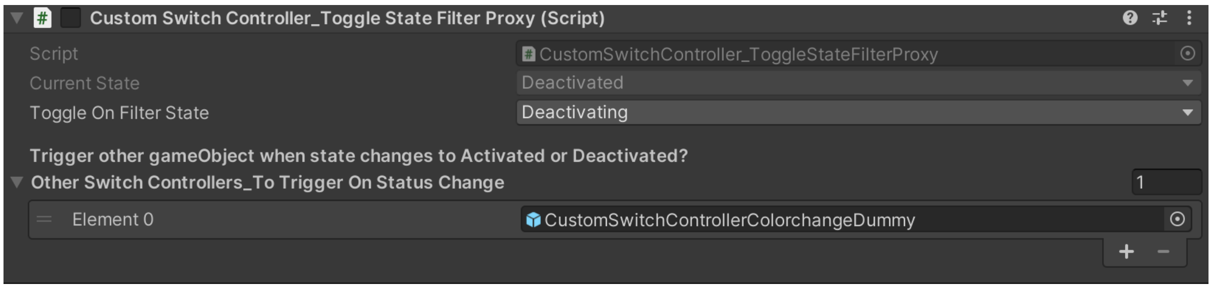
LastStateReceived(ReadOnly)	Last state of the caller that is known
OtherSwitchController_Caller	The GameObject with SwitchController that has an impact on the SubSwitchControllers state.
InvertExternalStateChange	To change the state of SubSwitch controllers all callers must have the same state or they can invert by this field the condition. For example with three callers: 0: Activated, 1: Deactivated(Inverted) 2: Activated SubSwitchControllers = Activating

4.6 CustomSwitchController_ToggleStateFilterProxy

Handles:

- Sets state of target Switch controller only if certain state is received

Standalone weaker version to do the same as the 'RewriteAnyStateWith' field on the SubSwitchController list.' Useful in custom scripts without implementing SubSwitchControllerSettingModel.



Custom Switch Controller_Toggle State Filter Proxy (Script)

Script	CustomSwitchController_ToggleStateFilterProxy
Current State	Deactivated
Toggle On Filter State	Deactivating

Trigger other gameObject when state changes to Activated or Deactivated?

Other Switch Controllers_To Trigger On Status Change: 1

Element 0	CustomSwitchControllerColorchangeDummy

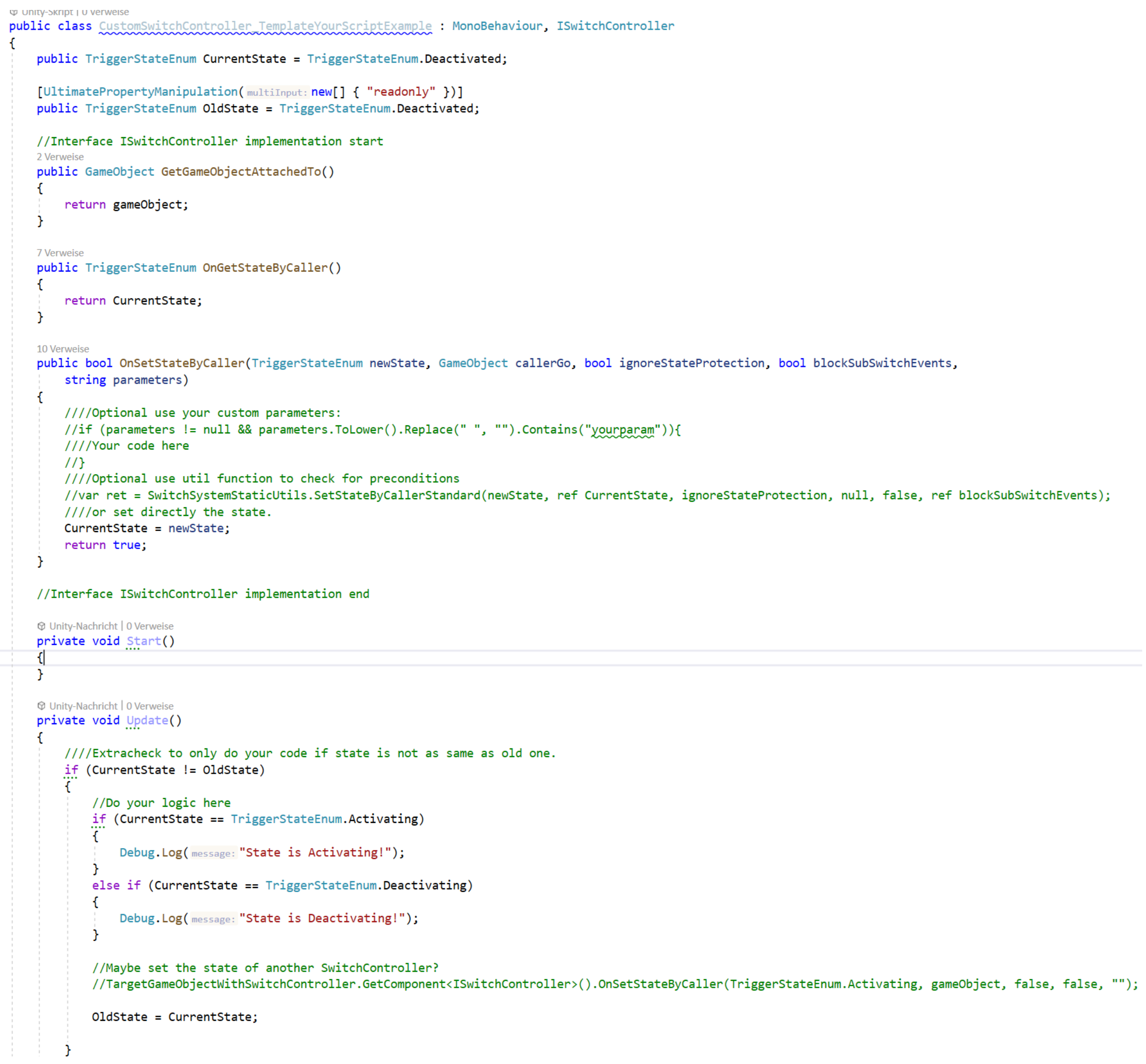
4.6 Create your own SwitchControllers

You can easily connect your scripts as SubSwitchControllers to other SwitchControllers by implementing the ISwitchController interface. They will automatically receive icons in the hierarchy window and you will be able to change their state by other SControllers.

Copy, rename the file CustomSwitchController_TemplateYourScriptExample.cs and add your logic. After that you can add them to any SubSwitchController field.

ISwitchController interface wants you to implement those methods:

- GetGameObjectAttachedTo: Used by hierarchy window icon draw. Do not change it!
- GetCurrentStateByCaller: To read the current state. Do not change it!
- OnSetStateByCaller: Other SwitchControllers or StateSetter where you attach your SwitchController will write to this Method a new state. You can write your full logic here if you don't want to use Update Method.



```
Unity-Skript | 0 Verweise
public class CustomSwitchController_TemplateYourScriptExample : MonoBehaviour, ISwitchController
{
    public TriggerStateEnum CurrentState = TriggerStateEnum.Deactivated;

    [UltimatePropertyManipulation(multiInput: new[] { "readonly" })]
    public TriggerStateEnum OldState = TriggerStateEnum.Deactivated;

    //Interface ISwitchController implementation start
    2 Verweise
    public GameObject GetGameObjectAttachedTo()
    {
        return gameObject;
    }

    7 Verweise
    public TriggerStateEnum OnGetStateByCaller()
    {
        return CurrentState;
    }

    10 Verweise
    public bool OnSetStateByCaller(TriggerStateEnum newState, GameObject callerGo, bool ignoreStateProtection, bool blockSubSwitchEvents,
        string parameters)
    {
        ////Optional use your custom parameters:
        //if (parameters != null && parameters.ToLower().Replace(" ", "").Contains("yourparam")){
        ////Your code here
        //}
        ////Optional use util function to check for preconditions
        //var ret = SwitchSystemStaticUtils.SetStateByCallerStandard(newState, ref CurrentState, ignoreStateProtection, null, false, ref blockSubSwitchEvents);
        ////or set directly the state.
        CurrentState = newState;
        return true;
    }

    //Interface ISwitchController implementation end

    Unity-Nachricht | 0 Verweise
    private void Start()
    {
    }

    Unity-Nachricht | 0 Verweise
    private void Update()
    {
        ////Extracheck to only do your code if state is not as same as old one.
        if (CurrentState != OldState)
        {
            //Do your logic here
            if (CurrentState == TriggerStateEnum.Activating)
            {
                Debug.Log(message: "State is Activating!");
            }
            else if (CurrentState == TriggerStateEnum.Deactivating)
            {
                Debug.Log(message: "State is Deactivating!");
            }

            //Maybe set the state of another SwitchController?
            //TargetGameObjectWithSwitchController.GetComponent<ISwitchController>().OnSetStateByCaller(TriggerStateEnum.Activating, gameObject, false, false, "");

            OldState = CurrentState;

        }
```

5. Usage SwitchControllerStateSetter

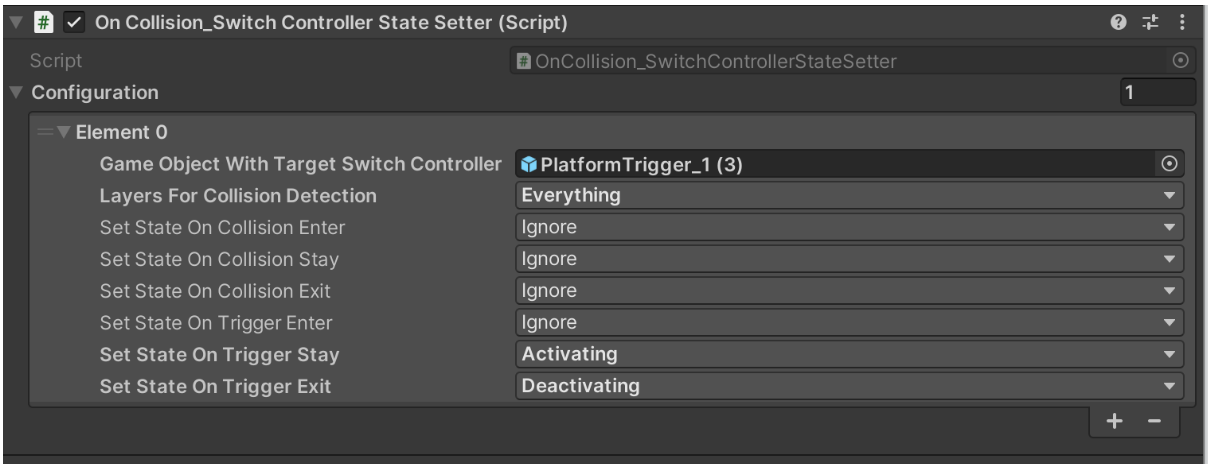
To set SwitchController state it is needed to call its OnSetStateByCaller(...) method. Scripts which are doing it and are not SwitchControllers themself, which dont have their own state, are defined here as "StateSetter".

5.1 OnCollision_SwitchControllerStateSetter

Handles:

- Sets a state of SwitchControllers depending on collision events.

This script can be used to activate switches if you step on them, changing the state of SwitchControllers by colliding with a collider.

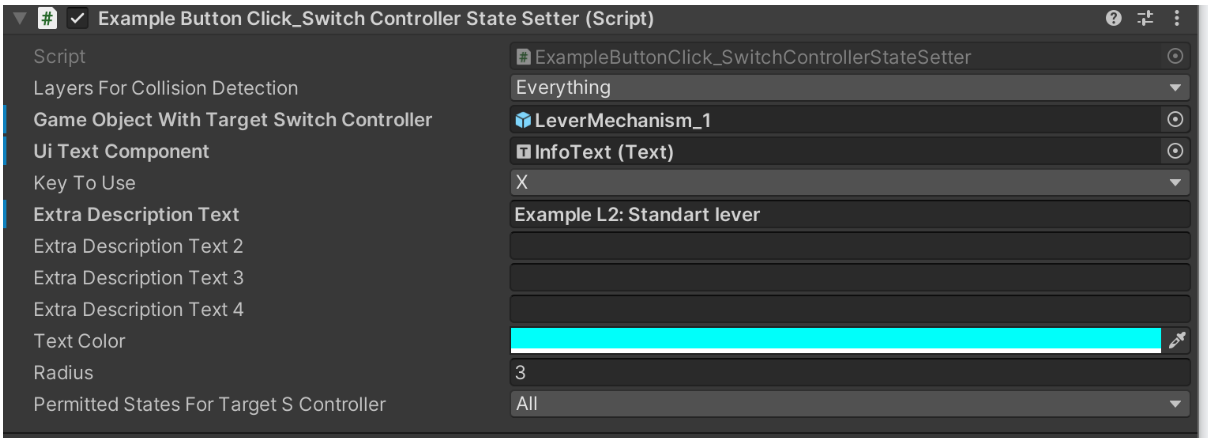


On Collision_Switch Controller State Setter (Script)

Script	OnCollision_SwitchControllerStateSetter
Configuration	1
Element 0	
Game Object With Target Switch Controller	PlatformTrigger_1 (3)
Layers For Collision Detection	Everything
Set State On Collision Enter	Ignore
Set State On Collision Stay	Ignore
Set State On Collision Exit	Ignore
Set State On Trigger Enter	Ignore
Set State On Trigger Stay	Activating
Set State On Trigger Exit	Deactivating

You can add multiple target objects with SwitchControllers into the configuration list.
Define the "LayersForCollisionDetection" or else unwanted GameObjects will trigger the state change on SwitchControllers.

Similar implementation is used by the ExampleButtonClick_SwitchControllerStateSetter.cs but adds InputEvent for old EventSystem. So after Player collides with collider he needs to press the right key to switch state of the attached GameObject.
Look into them to create your own StateSetters depending on your needs.
The UiTextComponent is optional in the demo script.



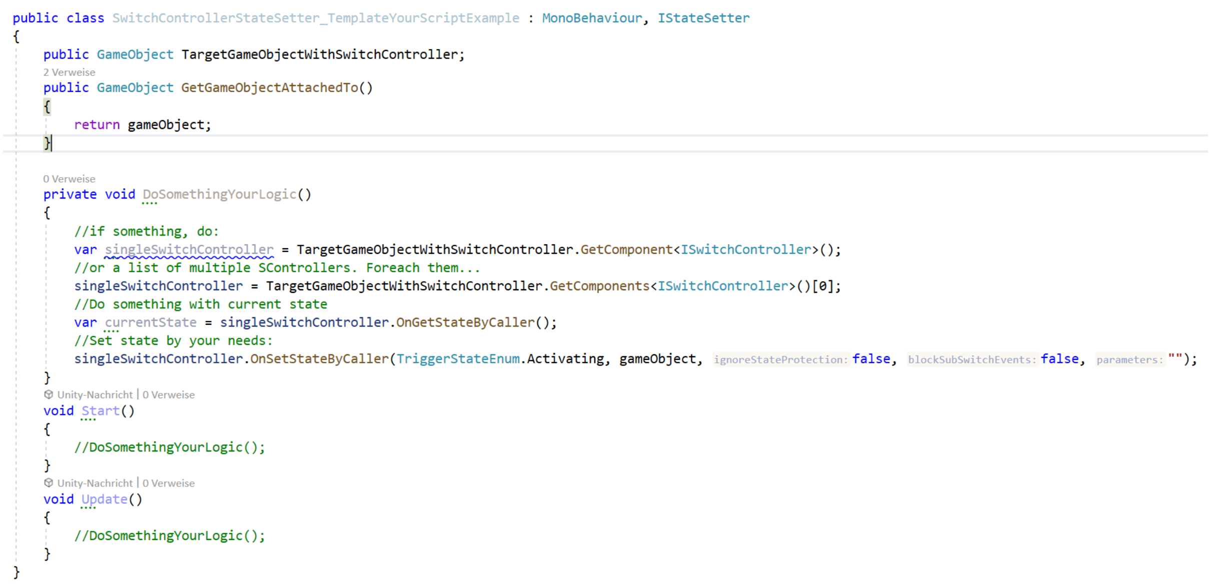
Example Button Click_Switch Controller State Setter (Script)

Script	ExampleButtonClick_SwitchControllerStateSetter
Layers For Collision Detection	Everything
Game Object With Target Switch Controller	LeverMechanism_1
Ui Text Component	InfoText (Text)
Key To Use	X
Extra Description Text	Example L2: Standart lever
Extra Description Text 2	
Extra Description Text 3	
Extra Description Text 4	
Text Color	🟦
Radius	3
Permitted States For Target S Controller	All

5.2 Create your own SwitchControllerStateSetter

To create your own StateSetter scripts simply get objects with SwitchControllers (That implements ISwitchController) and call OnSetStateByCaller on them. Add implementation of IStateSetter interface to your script to make it receive an icon in the hierarchy view.

You can use the example to simply create your StateSetter.
Copy, rename the file SwitchControllerStateSetter_TemplateYourScriptExample.cs and add your logic or look at OnCollision_SwitchControllerStateSetter.cs or ExampleButtonClick_SwitchControllerStateSetter.cs. After that you can set the states of SwitchControllers.



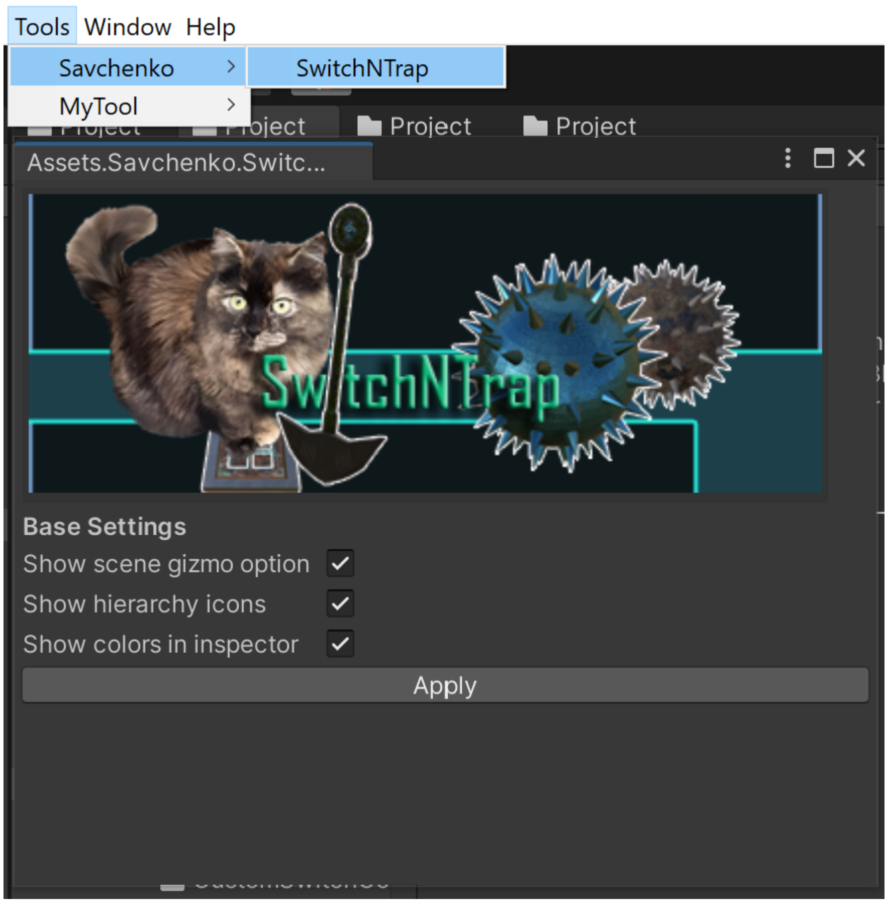
```
public class SwitchControllerStateSetter_TemplateYourScriptExample : MonoBehaviour, IStateSetter
{
    public GameObject TargetGameObjectWithSwitchController;
    2 Verweise
    public GameObject GetGameObjectAttachedTo()
    {
        return gameObject;
    }

    0 Verweise
    private void DoSomethingYourLogic()
    {
        //if something, do:
        var singleSwitchController = TargetGameObjectWithSwitchController.GetComponent<ISwitchController>();
        //or a list of multiple SControllers. Foreach them...
        singleSwitchController = TargetGameObjectWithSwitchController.GetComponents<ISwitchController>()[0];
        //Do something with current state
        var currentState = singleSwitchController.OnGetStateByCaller();
        //Set state by your needs:
        singleSwitchController.OnSetStateByCaller(TriggerStateEnum.Activating, gameObject, ignoreStateProtection: false, blockSubSwitchEvents: false, parameters: "");
    }
    Unity-Nachricht | 0 Verweise
    void Start()
    {
        //DoSomethingYourLogic();
    }
    Unity-Nachricht | 0 Verweise
    void Update()
    {
        //DoSomethingYourLogic();
    }
}
```

6. Visualization and Configuration

Switch and trap provides three types of helpers for your project :

- The coloring in the inspector
- The icons in hierarchy window that indicates which script is attached
- The gizmos in the scene window which shows connection of SwitchControllers. They support MainSwitchController and NodeWalker the best.

If you don't want to use any of these you can turn the options off:



Base Settings

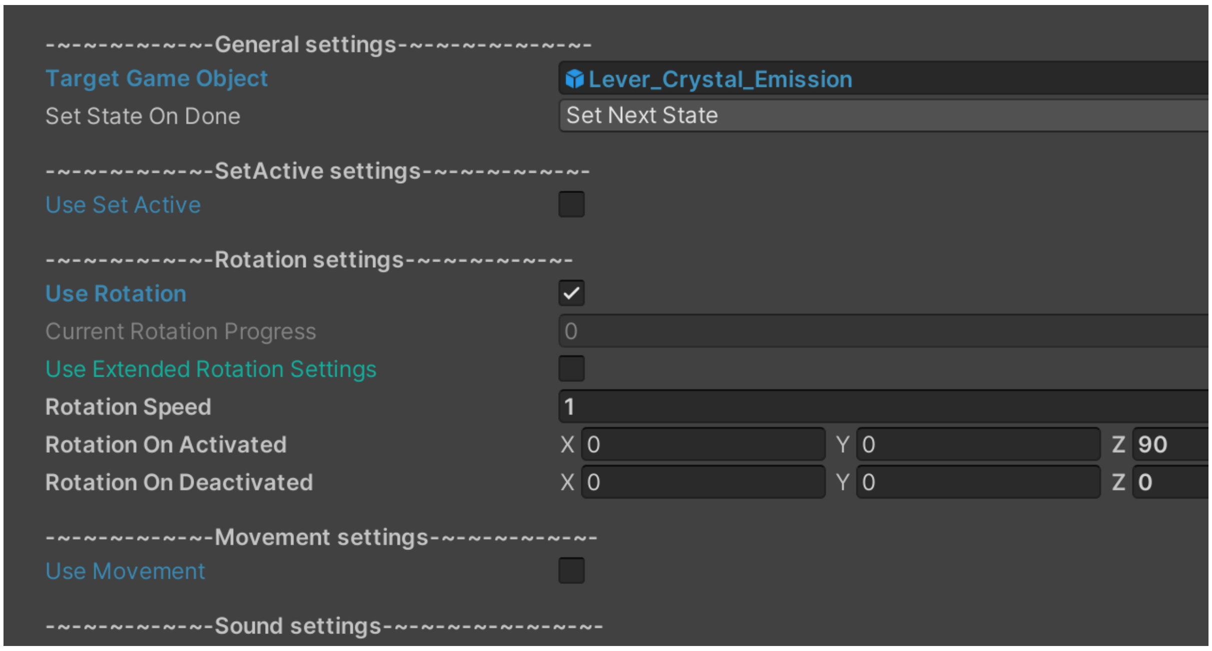
Show scene gizmo option ✓

Show hierarchy icons ✓

Show colors in inspector ✓

6.1 The coloring in the inspector

The SwitchControlles provides you with a lot of options to control objects. The coloring helps to keep easier track of main fields(blue tones) and optional or advanced settings (green tones). Other colors helps to keep apart some other settings.



-~-~-~-~-~-~-General settings-~-~-~-~-~-~-~-

Target Game Object: Lever_Crystal_Emission

Set State On Done: Set Next State

-~-~-~-~-~-~-SetActive settings-~-~-~-~-~-~-

Use Set Active: ☐

-~-~-~-~-~-~-Rotation settings-~-~-~-~-~-~-

Use Rotation: ✓

Current Rotation Progress: 0

Use Extended Rotation Settings: ☐

Rotation Speed: 1

Rotation On Activated: X 0 Y 0 Z 90

Rotation On Deactivated: X 0 Y 0 Z 0

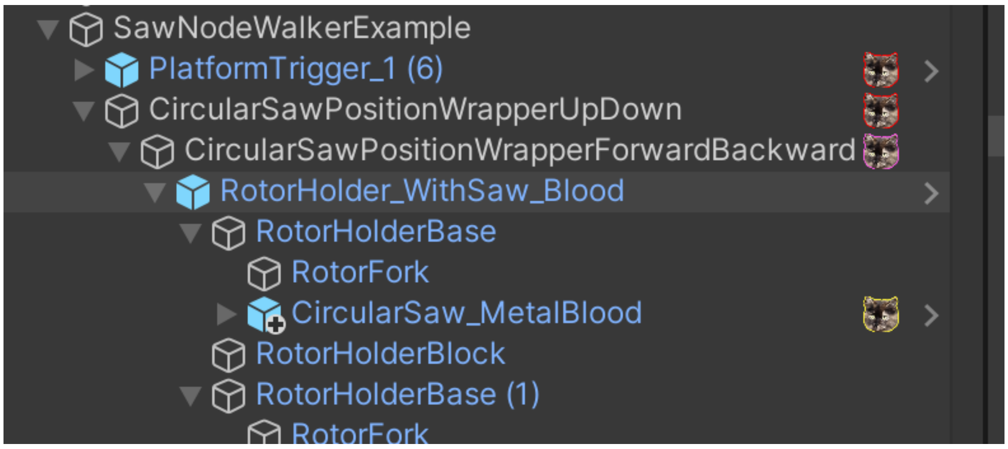
-~-~-~-~-~-~-Movement settings-~-~-~-~-~-~-

Use Movement: ☐

-~-~-~-~-~-~-Sound settings-~-~-~-~-~-~-~-

6.2 Hierarchy icons

To keep better control on attached scripts to objects there is an option to show the kitten icons on the hierarchy window. The border indicates which kind of script is attached. You can implement existing light interfaces in your custom scripts to mark them with icons.



```
SawNodeWalkerExample
  PlatformTrigger_1 (6)
  CircularSawPositionWrapperUpDown
    CircularSawPositionWrapperForwardBackward
      RotorHolder_WithSaw_Blood
        RotorHolderBase
          RotorFork
          CircularSaw_MetalBlood
        RotorHolderBlock
        RotorHolderBase (1)
          RotorFork
```

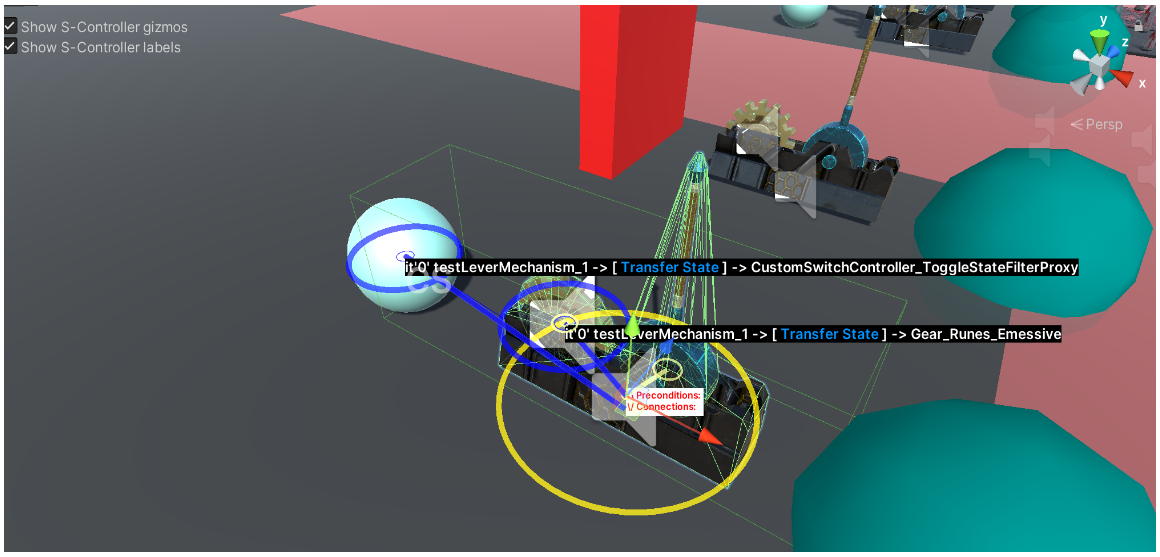
	MainSwitchController is attached.
	NodeWalker is attached.
	ConstantRotator is attached.

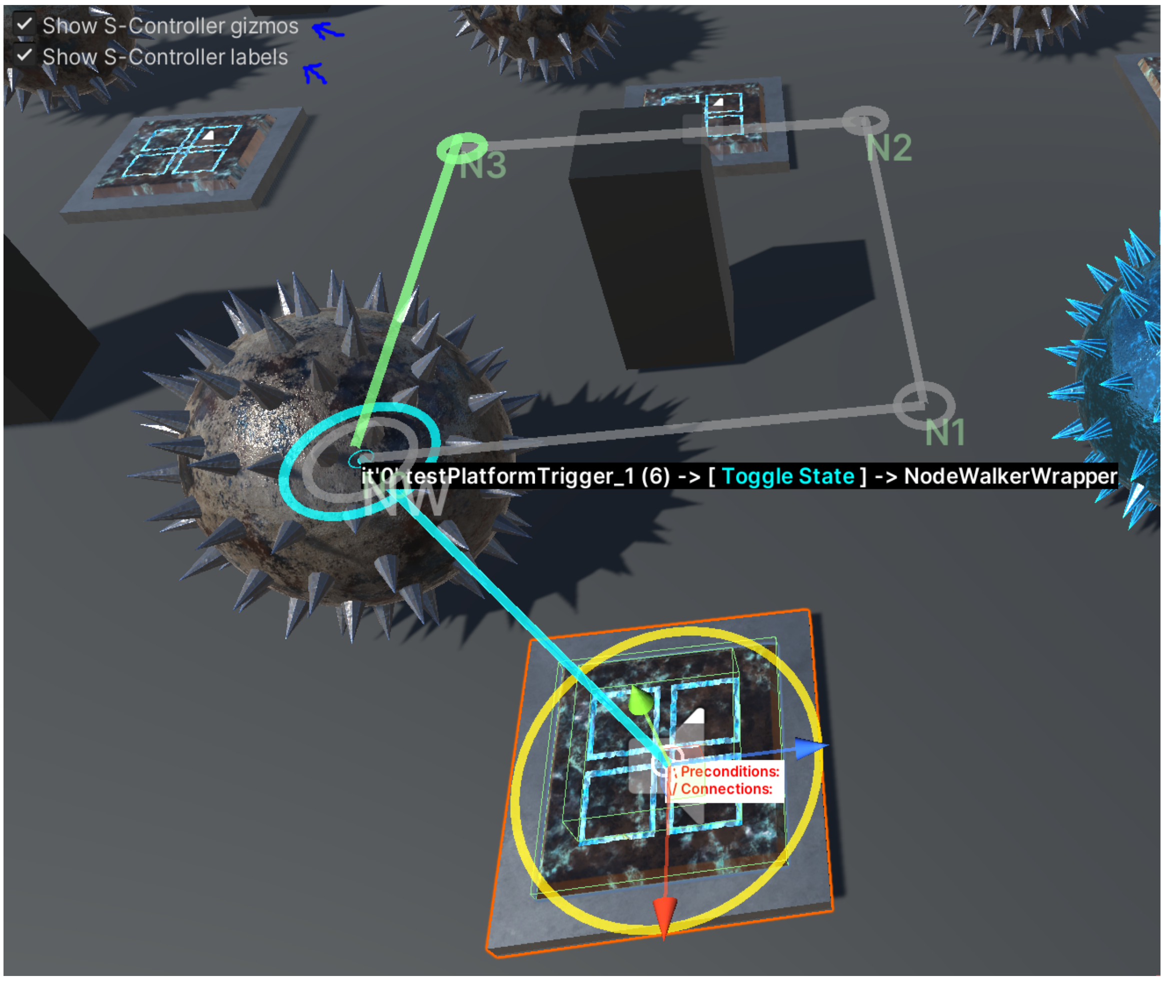
	Any other script which implements the “ISwitchController” interface is attached. Could be yours. just implement the ISwitchController interface. This will add icon and allow your script to be chain-connected with SwitchControllers.
	
	`public class YourScriptName : MonoBehaviour, ISwitchController`
	`{`
	Implement those methods:
	
	OnSetStateByCaller(..)
	and
	OnGetStateByCaller();

	Scripts attached that can set the state of SwitchControllers for example scripts on collider which activates buttons, see “OnCollision_SwitchControllerStateSetter”. You can create custom scripts and implement “IStateSetter” to add to them the icon.

6.3 Gizmos in scene view

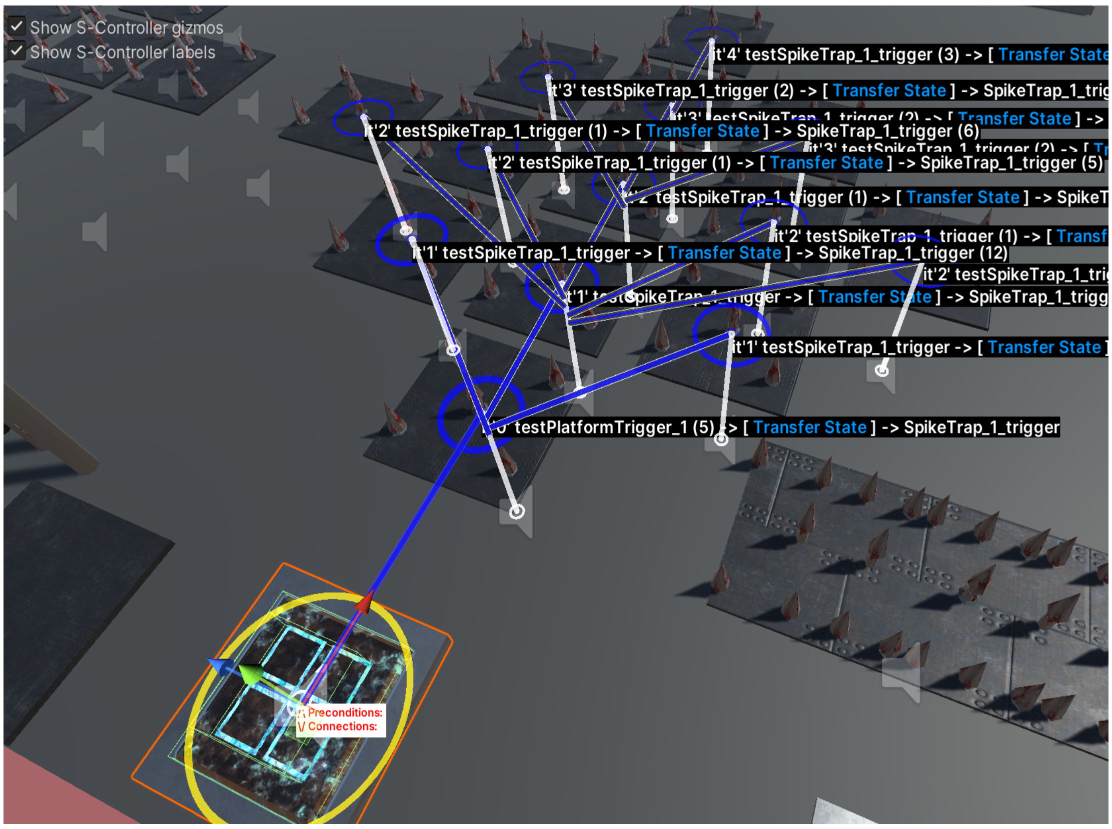
To help visualize the connection between SwitchControllers you can use the SwitchNTrap-gizmos. The color indicators and labels show the configuration of the SwitchControllers.


Show S-Controller gizmos
Show S-Controller labels
Persp
it'0' testLeverMechanism_1 -> [Transfer State] -> CustomSwitchController_ToggleStateFilterProxy
it'0' testLeverMechanism_1 -> [Transfer State] -> Gear_Runes_Emessive
Preconditions:
Connections:



The lines for MianSwitchController can show the state transmission to other SwitchControllers or preconditions and positions. Compare the gizmos on demo scenes with the scripts to fully understand the meaning of the indicators and colors.

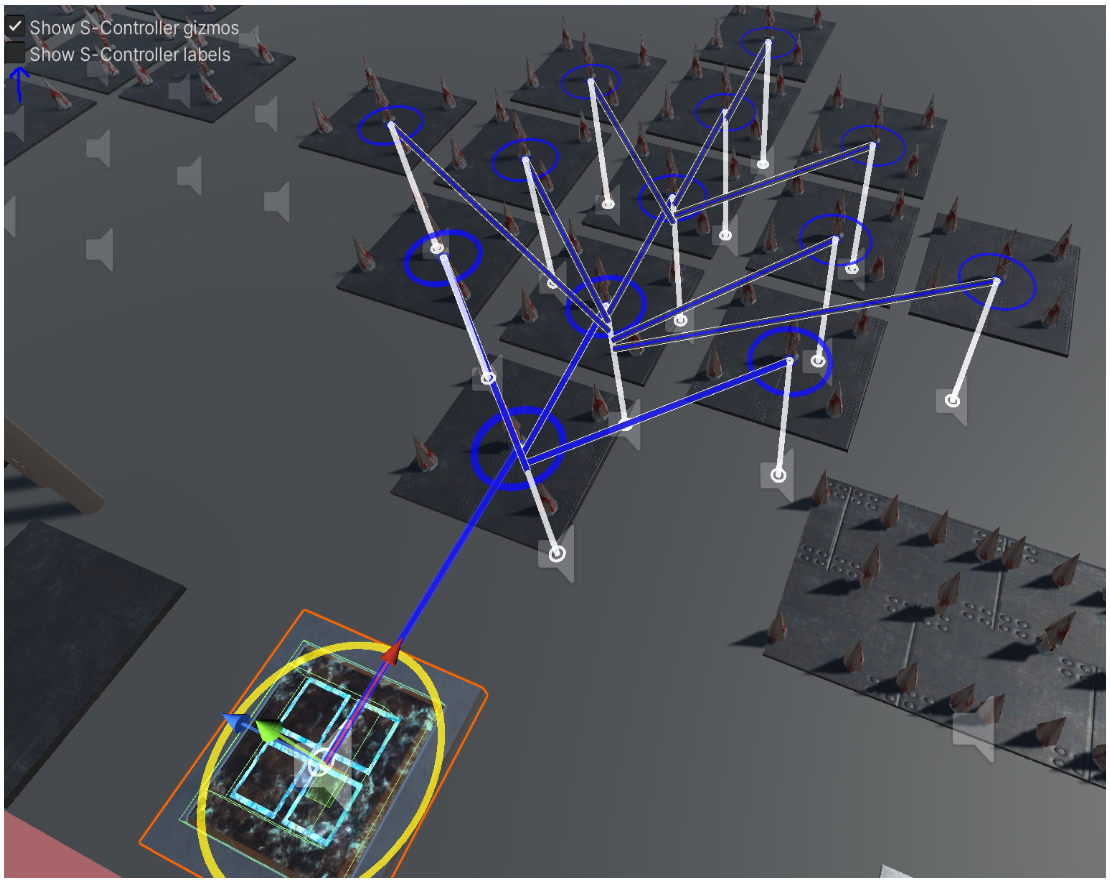
To easily read the labels, zoom-in the object by focusing it first with the button “F” and rotate the perspective.



- [✓] Show S-Controller gizmos
- [✓] Show S-Controller labels

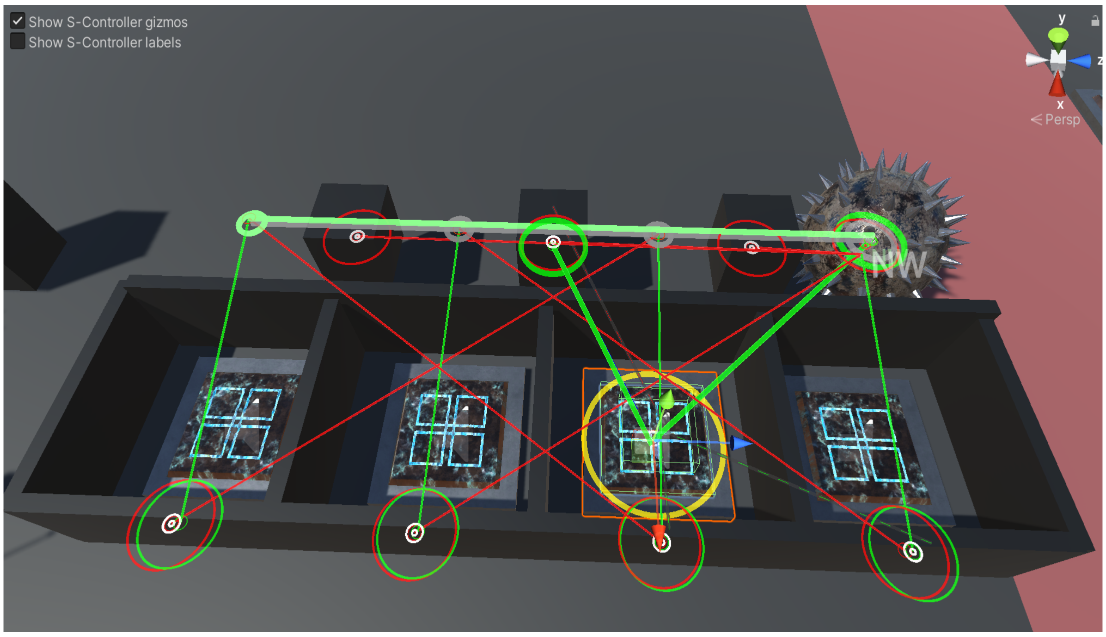
it'4' testSpikeTrap_1_trigger (3) -> [Transfer State
it'3' testSpikeTrap_1_trigger (2) -> [Transfer State] -> SpikeTrap_1_trig
it'2' testSpikeTrap_1_trigger (1) -> [Transfer State] -> SpikeTrap_1_trigger (6)
it'2' testSpikeTrap_1_trigger (1) -> [Transfer State] -> SpikeTrap_1_trigger (5)
it'1' testSpikeTrap_1_trigger -> [Transfer State] -> SpikeTrap_1_trigger (12)
it'1' testSpikeTrap_1_trigger -> [Transfer State]
'0' testPlatformTrigger_1 (5) -> [Transfer State] -> SpikeTrap_1_trigger
Preconditions:
Connections:

Sometimes you don't want to see labels because they can overload the view. just turn them off locally by clicking the checkbox.



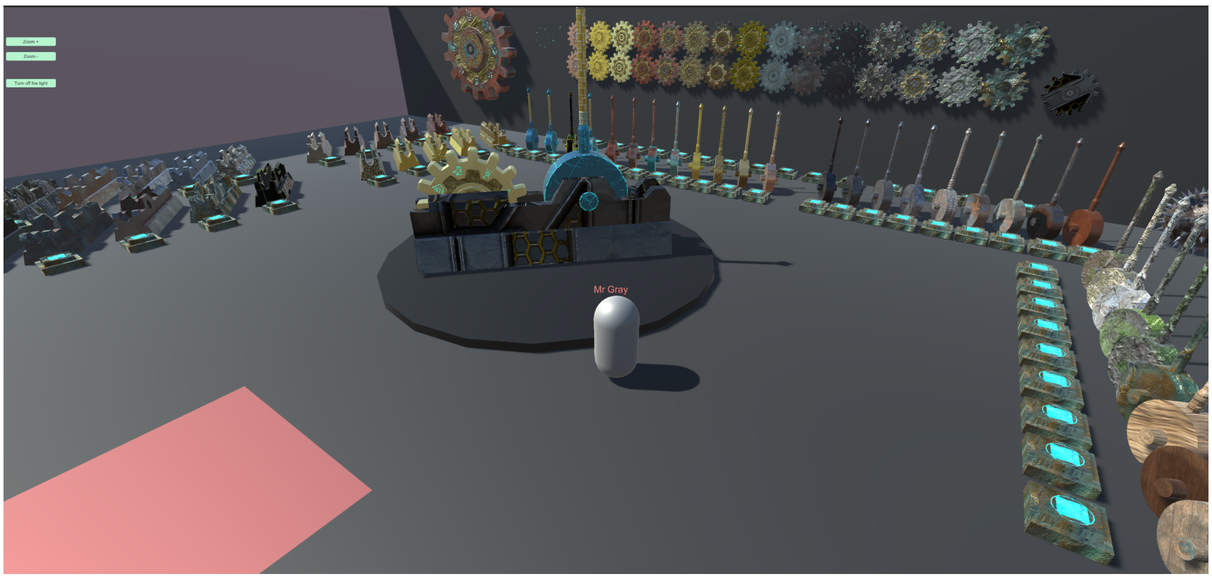
- [✓] Show S-Controller gizmos
- [ ] Show S-Controller labels

This way you get better control on full configuration of complex connections.

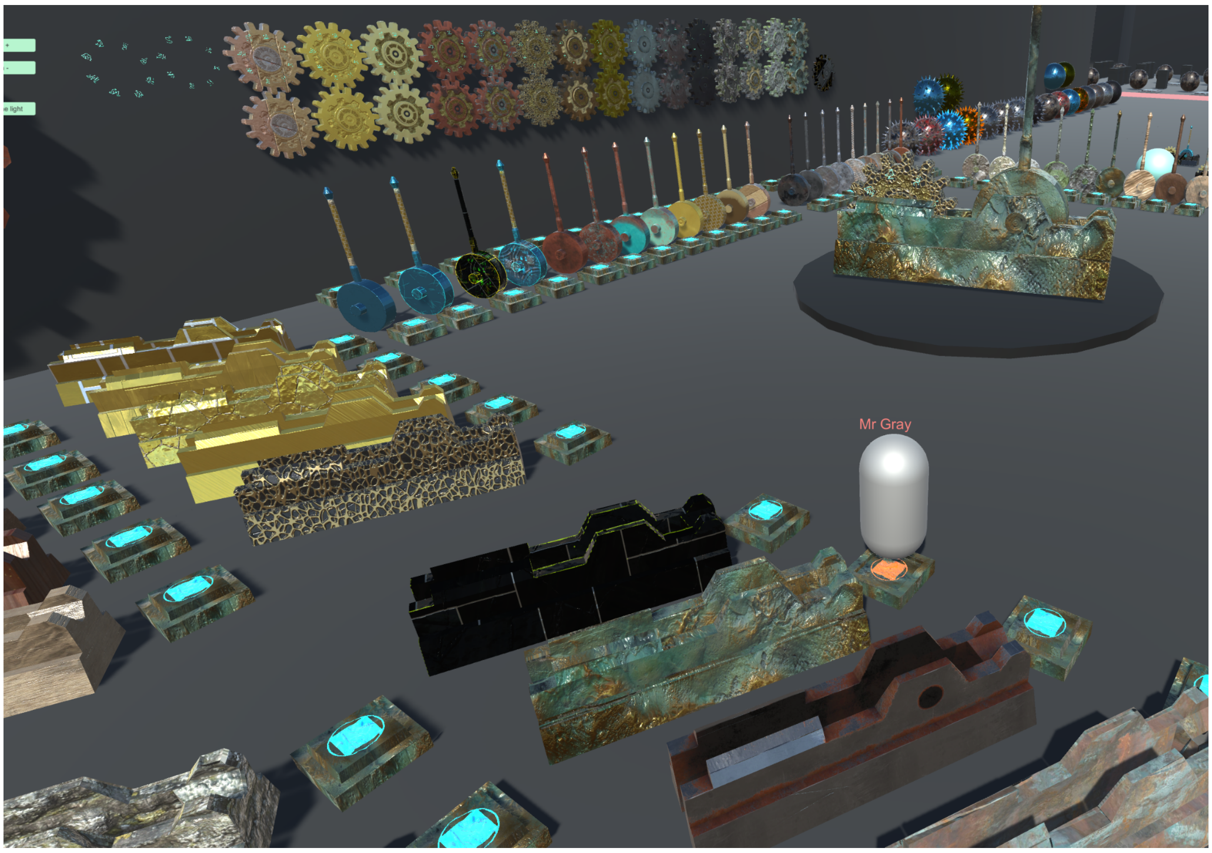


7. Lever generator (in demo scene)

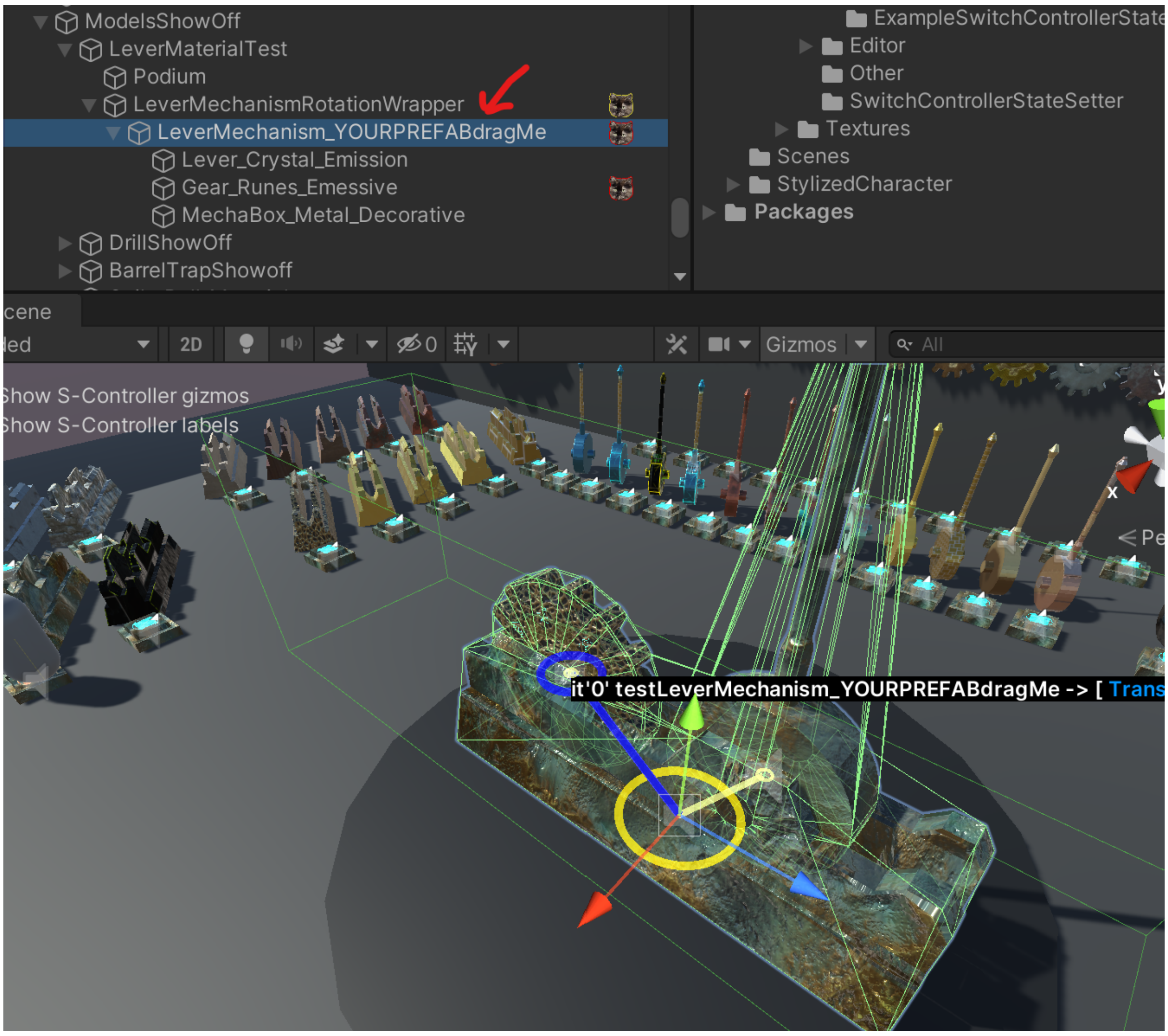
In the demo scene you can combine your custom levers.



Just step with the character on the ground switches to choose the materials.



Select in the scene the leaver, drag and drop it in your asset folder, give it a name.


ModelsShowOff
LeverMaterialTest
Podium
LeverMechanismRotationWrapper
LeverMechanism_YOURPREFABdragMe
Lever_Crystal_Emission
Gear_Runes_Emessive
MechaBox_Metal_Decorative
DrillShowOff
BarrelTrapShowoff
ExampleSwitchControllerState
Editor
Other
SwitchControllerStateSetter
Textures
Scenes
StylizedCharacter
Packages
cene
led
2D
0
Gizmos
All
Show S-Controller gizmos
Show S-Controller labels
it'0' testLeverMechanism_YOURPREFABdragMe -> [Trans
y
x
< Pe