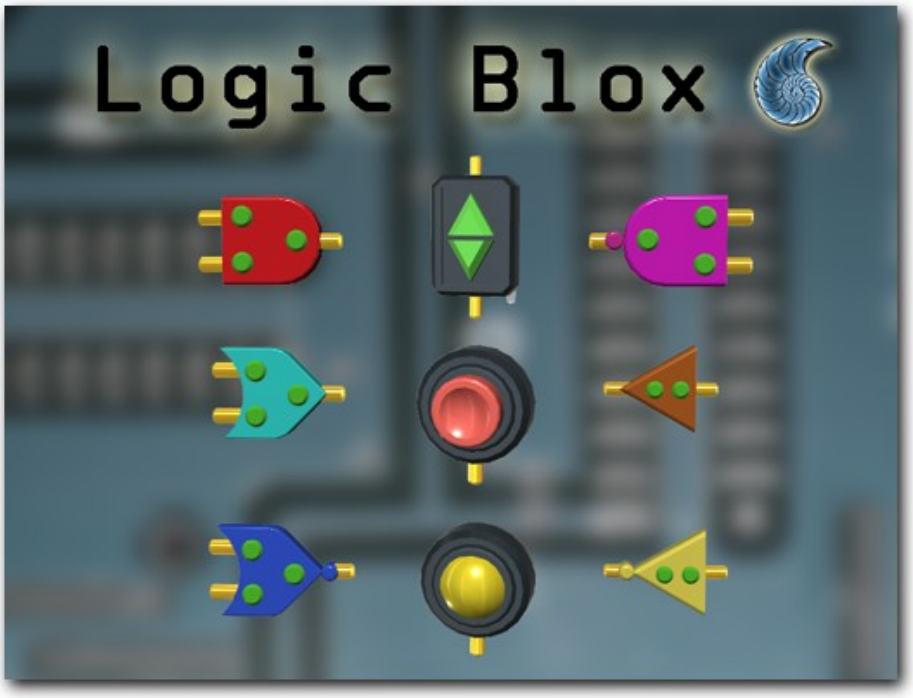


Primitive logic gate series

User Guide

Oct/2018

Granby Games

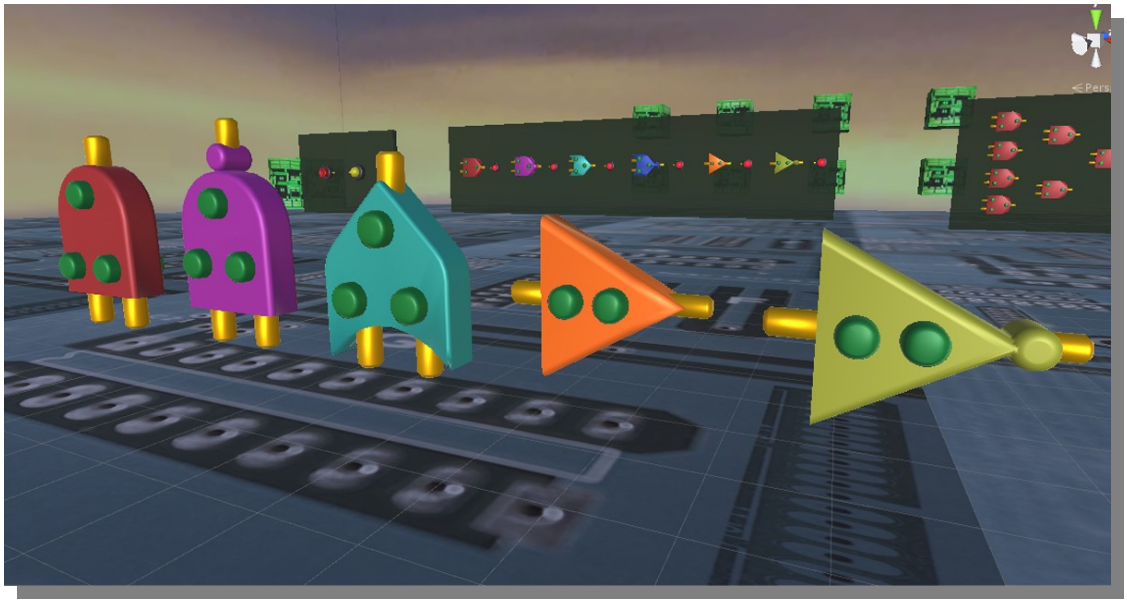
Table of Contents

Logic Blox – Primitive gates series, User guide

Introduction

Granby Games logic prop prefabs (Logic Blox) can be used to create, debug and design simple to complex virtual electronic circuits and or game play controls. Logic Blox are placed in game and can be connected together in order to run other logic props, sounds, animations and game objects. Pre-attached script file(s) manage behavioral logic, animations and connections to other props.

Gate primitives are the first in a series of virtual logic props providing foundational logic to all Logic Blox components. Future releases include vintage computer components, integrated circuits and advanced virtual machinery.



Contained in this user guide are detailed Logic Blox prop descriptions, examples and operational settings. Logic Blox packages can be downloaded from the Unity asset store.

Logic Blox prefabs

Logic Blox prefabs contain meshes, materials and Logic Blox Script(s) which have been presetup for each logic gate type. Simply place props in game scene and use inspector to establish pin connections to other props. Clicking on pins and or buttons changes prop logic state which in turn drives logic flow to other connected props.

In addition, prefabs have optional user changeable settings for button colors, led emissions, clocking, sound, animation and game object control.

Prop logic circuit types (And, Nand, Or, Nor, Not, Buffer, Switch, Lamp) are setup with different classes that represent common pin configurations:

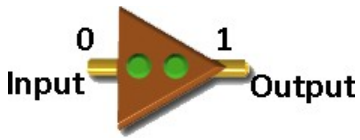
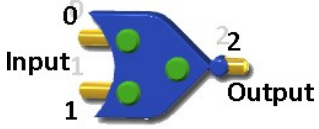
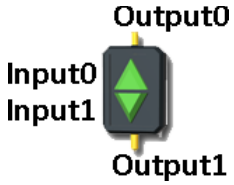
Gate Classes

1. **Gate** - Has one input and one output. Includes: Buffer, Not gate, Switch, Lamps

2. **Primitive** - Has two inputs and one output. Includes: And, Nand, Or, Nor

3. **Axis** - Has two inputs and two outputs. Includes up/down dials

Standard pin configurations

		
Gate	*Primitive*	*Axis*

Logic Blox – Primitive gates series, User guide

Script and component details

Several Unity built-in components (Rigid Body, Colliders and AudioSource) plus two Logic Blox script files (UniLogicChip and UniLogicEffects) are attached to each prop and are required to run core logic engine, input detection, game object control and external effects. Purchased prefabs have been presetup with necessary scripts and components. Special functions and or prop behavior modifications can be altered in the scripts inspector fields. Details for modifying advanced logic prop settings are provided in later sections.

Input output

Input pins and button tops are click sensitive. Clicking an input pin or button will change internal gate values which in turn produce outputs based on chip type (ie: and, or, not). Pin states are tied to indicator Led's and will illuminate when pin state changes to logical on. Inputs can also be buttons and or objects attached to props and defined by the inspector(pinObj)

External links

When a gates logical true condition is met the props gates output will trigger and all connected gates (listed in Output Link Obj field) will receive its output signal at defined pin(s).

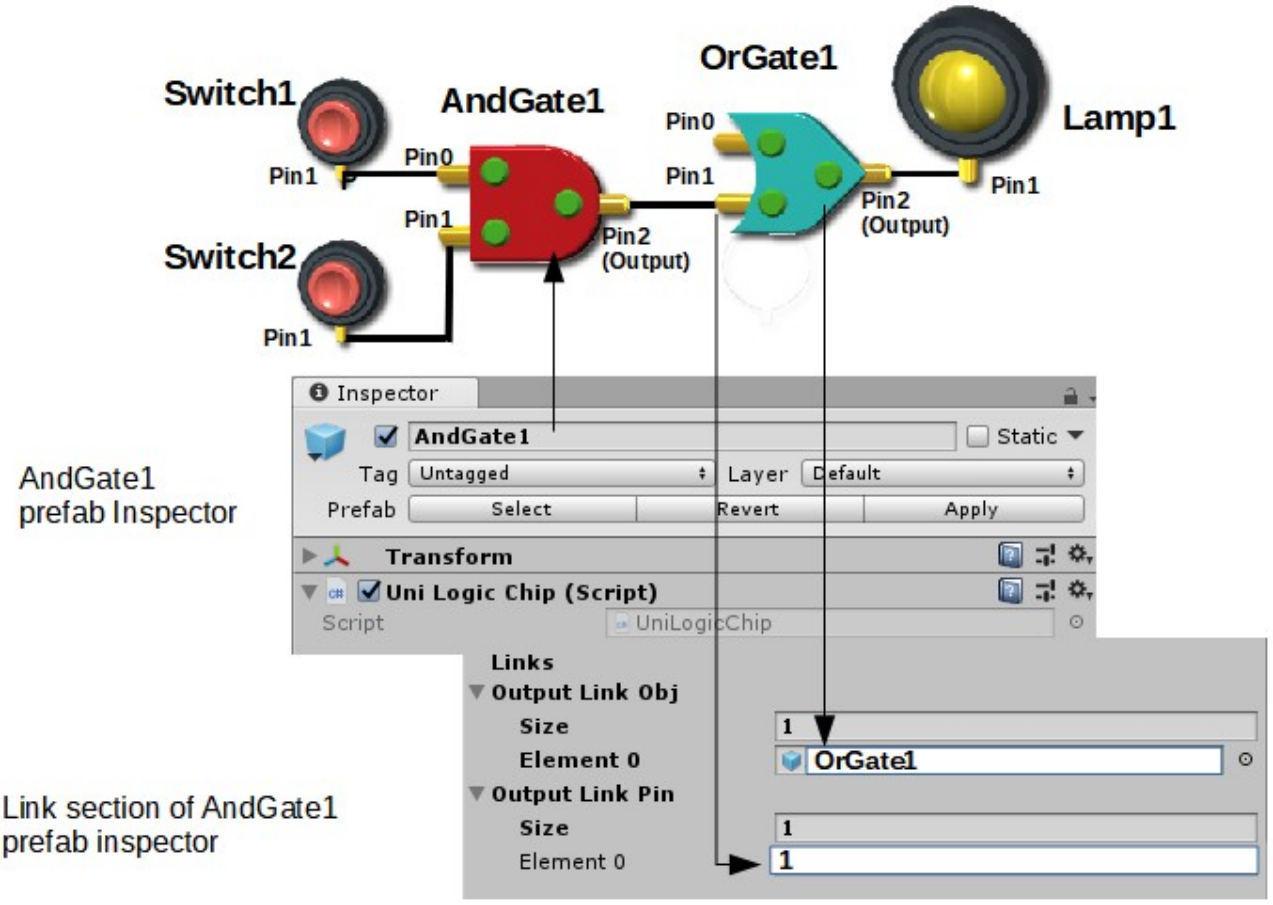
Effects

Pre-attached UniLogicEffects script will provide activation of sounds, animations and other game objects. (details in later section)

Setup steps

1. Install Logic Blox primitives package from Unity Asset store:
   https://assetstore.unity.com/packages/3d/props/logic-blox-129409
   Package can be re-installed at any time to apply updates and newly added components
2. Add the following tags to the Unity editor tag list : 0,1,2,3,4,5,6,7,8,9
3. Activate demo scene to become familiar with operation and features
4. Set up your own gates and circuits by dragging Logic Blox prefabs onto desired scene.

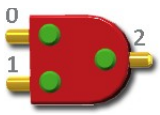
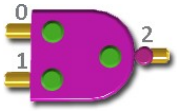
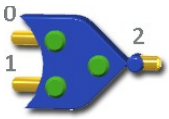
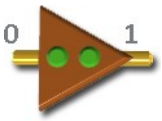
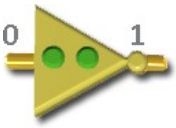
5. In order to connect gates, drag and drop other Logic Blox objects onto inspector (Output Link Obj) and enter pin number in (Output Link Pin) field.



Note: Multiple gates can be linked by dropping additional gates onto the (Output Link Obj) array. Element size of (Output Link Pin) array must match Output Link Obj element size.

Primitive gates series reference table:

Type	Class	Pin Config	Logic behavior	Special features

Name				
And	primitive		Both inputs set to On changes output to ON	Illuminated pin status
Nand	primitive		Both inputs set to OFF changes output to ON	Illuminated pin status
Or	primitive		Ether input set to ON changes output to OFF	Illuminated pin status
Nor	primitive		Either Intput set to OFF changes output to ON	Illuminated pin status
Buffer	gate		Output follows input	Illuminated pin status Can be a clock or countdown timer Can be momentary on
Not	gate		Output is inverse of input	Illuminated pin status Can be a clock or countdown timer Can be momentary on
UpDn	axis		Direction pin changes based on button pressed (2 outputs)	Illuminated Buttons Can be momentary On Can be preset to run special functions

Lamp	gate		Illuminates when input set to ON	Color choices and illumination brightness
Switch	gate		Out changes to ON when button presses	Changeable button color choices Can be preset to run special functions

Audio Effects and gameobject control

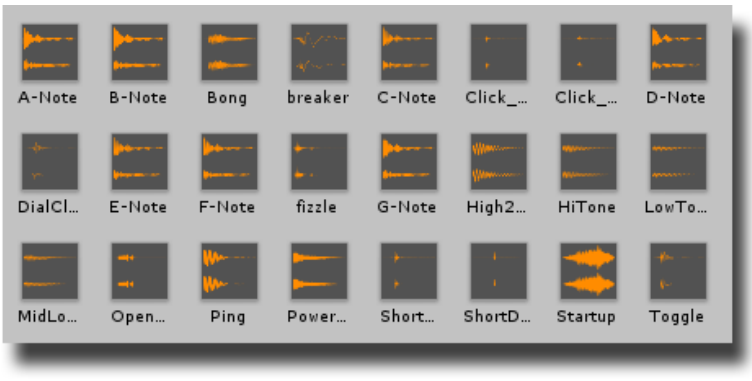
UniLogicEffects.cs - attached script

Provides sound effects and remote game object control. Specific events are triggered when output states switches to On (and when output switches to Off). For example. When user clicks an Input pin the Click_On audio clip will play.

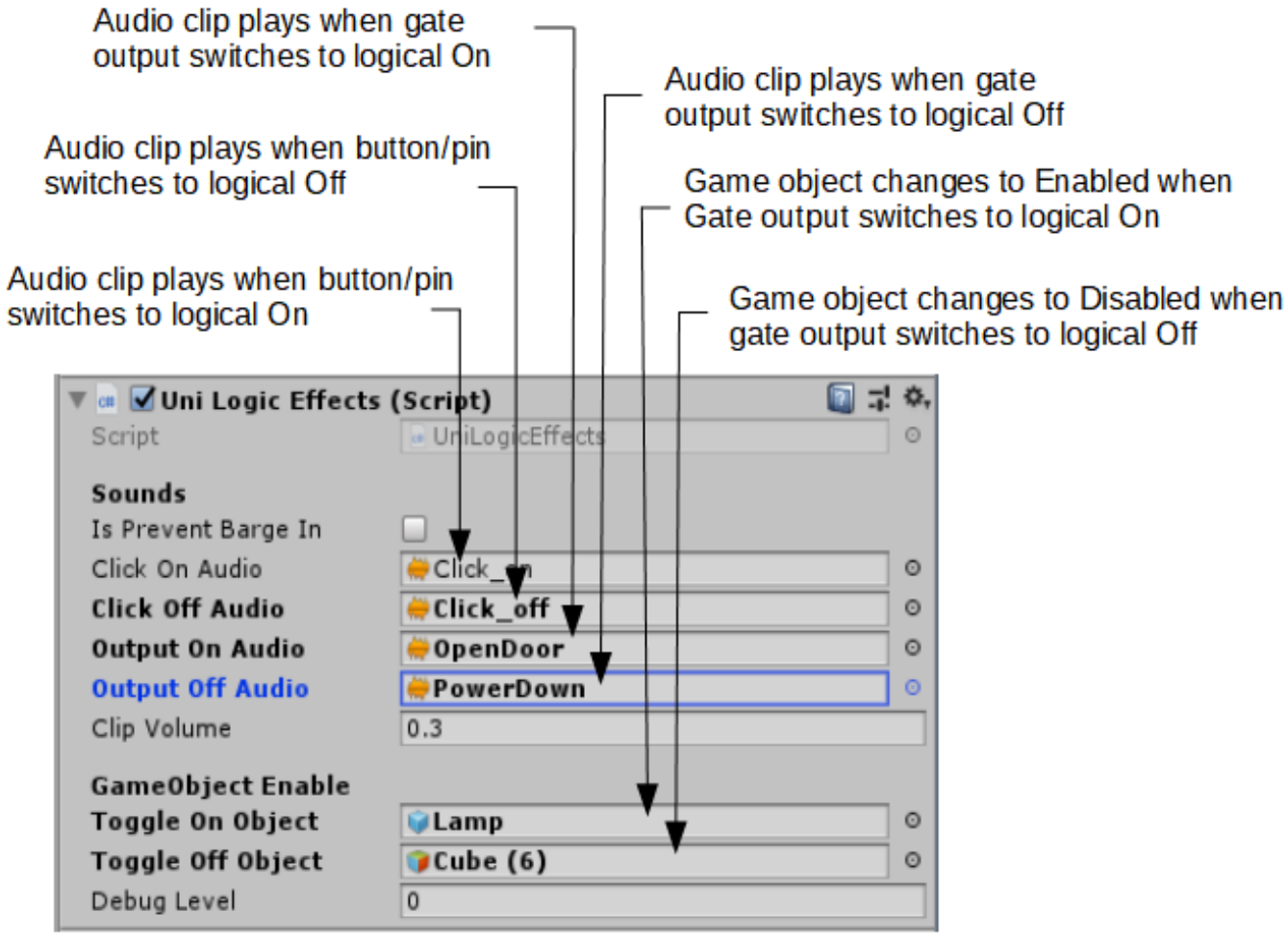
Logic Blox prefabs have been setup with Audio Source components and default Click_On Audio clip enabled. You may want to remove audio file from the inspector field to avoid click sounds every time a gate fires.

> Note: If your build has gates that don't require effects, UniLogicEffectsscript and or individual sounds can be safely removed from prefab copies..

Audio clip files are located in the LogicBlox package Audio folder. Drag and drop desired audio file onto appropriate trigger field. (See UniLogicEffects inspector options diagram)



UniLogicEffects inspector options



Additional fields

Is Prevent Barge In – Prevents new audio clip from playing on specific gate while its audio source is currently playing a clip.

Clip Volume – Adjusts gates audio source volume by set amount. (float value between -1 and +1). Can be set globally using gate triggerable sub function calls (see details in later section)

Debug Level – Set to 1 for verbose gate activity logging. Includes object names, pin numbers, audio files and link connection signals. Extremely handy for debugging. Can be set globally

using gate triggerable sub function calls (see details in later section)

Colors and illumination

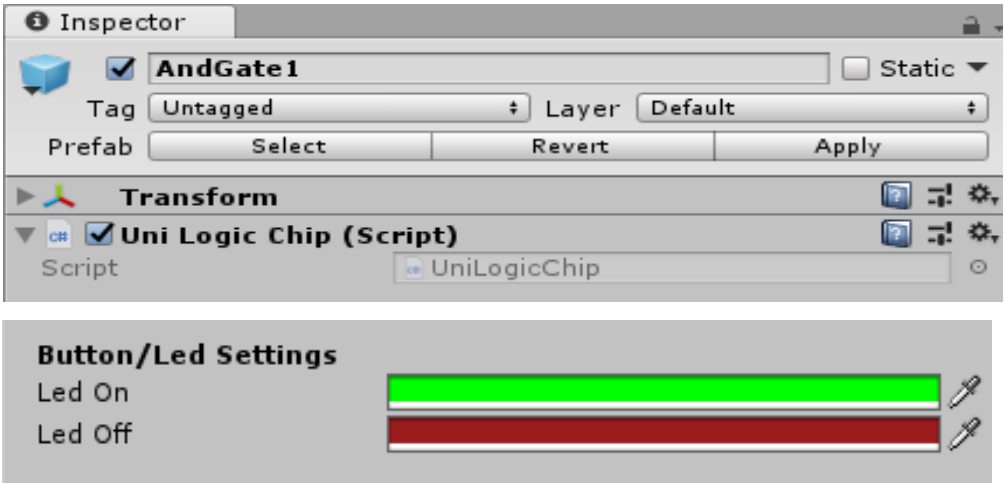
Lamp and led default colors can be set via the UniLogicChip's Led/Buttons inspector settings. Defaults are (yellow or green) for On and (dim yellow or red) for Off. Emissions are toggled based on pin state and can be altered from pin objects material settings (Unity materials editor).

Push Buttons

Props that contain user push buttons such as switches and dials are setup with colliders in order to detect user mouse clicks and or collisions with other objects. Button top colors can be set in the UniLogicChip inspector Leds/Buttons section.

Note: Button animations(transform positioning) for button pushes are pre-setup in provided switch prop prefabs.

UniLogicChip Inspector settings for changing Lamp, Led and Button top colors.



Logic Blox – Primitive gates series, User guide

Advanced Functions

At the basic level. Logic Blox communicate with each other via pin states. A logical On output state from one prop connected to another prop's input will drive the receivers input to a logical On. In the case of inverted outputs, logical On=Off

A receivers output event is controlled by input states and internal logic defined by chip type. For example:: If an And gate receives a logical On state on both input pins its output state will change to logical On. If one or more of the inputs changes to logical Off, the output state changes to logical Off.

Logic Blox have user definable inspector settings for enabling functions such as clocks, triggers and timers. Various parameters including default state, momentary on, clock speed, pulse width, countdown counts, sub functions and circuit groups can be defined.

Gate Parameters

Chip Type – Predefined value that determines props logical behavior

Default State – lamps and gates output state defaults to logical on.

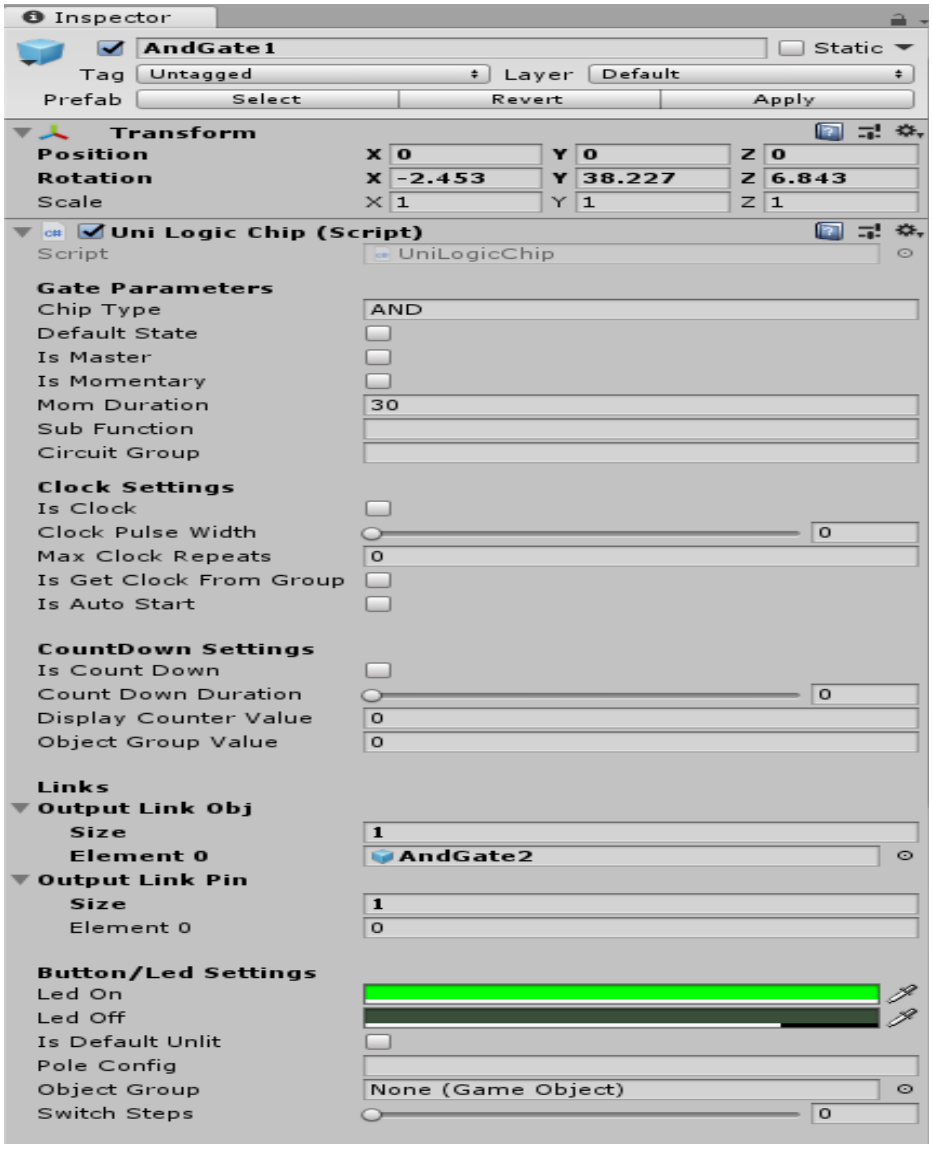
Is Master - Set to true if is a clocking gate and is first in a chain of clock driven gates.

Is Momentary - For switches and UpDn props. After being turned on, gate will turn off after specified delay (Mom Duration)

Sub Function – Runs predefined sub functions (listed in table)

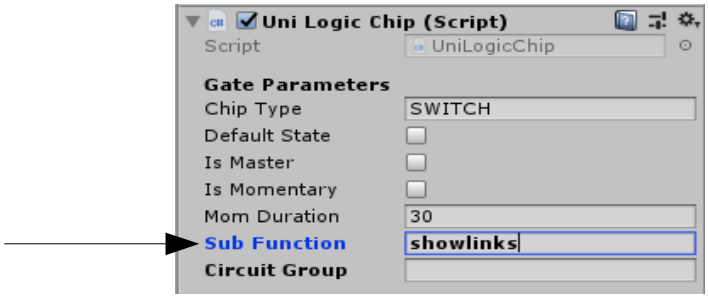
Circuit Group – Used to group sub function actions to performed of group of logic props

UniLogicChip.cs script inspector fields



Sub Function Values

String values can be entered into the Sub Function field in order to send out advanced commands to one, some or all props.



1. "clockRate" - sets speed of repeating buffer.
2. "volume" - sets volume of connected prop that uses audio clips
3. "showLinks" - Toggle link lines on and off
4. "resetGlobal" - Reset all props to default state (global reset)
5. "power" – Prop will not be reset by global reset
6. “disableLeds” - Disables props Led output while internal logic continues to run.
7. “disableCircuitLeds” - Disables leds with in same circuit group name

Clock Settings

Gates and switches can perform clocking functions (repeating pulse) by enabling the following inspector settings

1. IsCLock - Set to true makes buffer or switch into clock.
2. MaxClockRepeats - Stop clock after number of repeated cycles
3. ClockPulseWidth – pause time between clock cycles

Countdown Settings

1. IsCountDown - Gate will trigger output when countdown reaches 0.
2. CountDownDuration - Number of clock cycles to wait until output triggers

Note: Gate cannot have isClock and isCountDown set at the same time.

Gate type features and abilities

	LOD	Clock	Countdown	Momentary	Audio	Animations	Lighting	Remote
And					x			x
Nand					x			x
Or					x			x
Nor					x			x
Not		x	x	x	x			x
Buffer		x	x	x	x			x
Switch		x	x	x	x	x		x
UpDn					x	x		x
Lamp					x			x
								x
								x