

Search: **Assets**

- Backward
- BlueFadeIn
- Forward
- GreenFadeIn
- Left
- RedFadeIn
- Right
- Spin
- WhiteFadeIn

Animations.

- Backward (constant) string Animations.Backward = "Backward"
- BlueFadeIn
- Count
- Equals
- Forward
- GreenFadeIn
- Left
- RedFadeIn
- ReferenceEquals

Search: **Assets**

- Athlete
- BlueSphere
- GreenSphere
- MobileObject
- RedSphere
- WhiteSphere

AnimatorControllers.

- Athlete class HardCodeds.AnimatorControllers.Athlete
- BlueSphere
- Count
- Equals
- GreenSphere
- MobileObject
- RedSphere
- ReferenceEquals
- ToArray

Size: 19

- Horizontal
- Vertical
- Fire1
- Fire2
- Fire3
- Jump
- Mouse X
- Mouse Y
- Mouse ScrollWheel
- Horizontal
- Vertical
- Fire1
- Fire2
- Fire3
- Jump
- Submit
- Submit
- Cancel
- UserAxis

Axes.

- Cancel (constant) string Axes.Cancel = "Cancel"
- Count
- Equals
- Fire1
- Fire11
- Fire2
- Fire21
- Fire3
- Fire31

Layers

Builtin Layer 0	Default
Builtin Layer 1	TransparentFX
Builtin Layer 2	Ignore Raycast
Builtin Layer 3	
Builtin Layer 4	Water
Builtin Layer 5	UI
Builtin Layer 6	
Builtin Layer 7	
User Layer 8	Black Cubes
User Layer 9	Red Cubes
User Layer 10	Green Cubes
User Layer 11	Blue Cubes

Layers.

- AddToLayerMask
- AllCount
- AllToArray
- Black_Cubes (constant) int Layers.Black_Cubes = 8
- Blue_Cubes
- BuiltinsCount
- BuiltinsToArray
- Count
- Default

Resources

- Materials
 - Black
 - Blue
 - Green
 - Hard-Coded Generator_Icon_Dark
 - Red
 - White
- Prefabs
 - Spheres
 - Blue Sphere
 - Green Sphere
 - Red Sphere
 - White Sphere
 - Athlete
- Textures
 - Dialog Bubble
 - Hard-Coded Generator_Icon_Dark
 - Hard-Coded Generator_Icon_Light
 - Cube

Assets.

- Get
- Materials class HardCodeds.Assets.Materials
- NestedCount
- NestedToArray
- Path
- Prefabs
- ReferenceEquals
- Textures
- ToArray

Build Settings

Scenes In Build

- HardCodedGenerator/Demo/Scenes/Main.unity
- HardCodedGenerator/Demo/Scenes/Main (Backup).unity

Add Current

Scenes.

- Count
- Equals
- GetName
- Main (constant) int Scenes.Main = 0
- ReferenceEquals
- ToArray

Sorting Layers

Layer	Default
Layer	Background
Layer	Midground
Layer	Foreground

+ -

SortingLayers.

- AllCount
- AllToArray
- Background (constant) int SortingLayers.Background = -1733561729
- BuiltinsCount
- BuiltinsToArray
- Count
- Default
- Equals
- Foreground

Tags

Tag 0	Plane
Tag 1	BlackCube
Tag 2	RedCube
Tag 3	GreenCube
Tag 4	BlueCube

+ -

Tags.

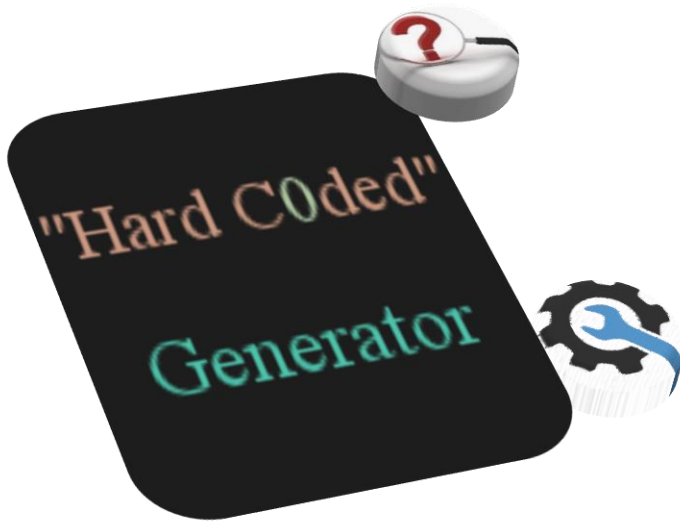
- AllCount
- AllToArray
- BlackCube (constant) string Tags.BlackCube = "BlackCube"
- BlueCube
- BuiltinsCount
- BuiltinsToArray
- Count
- Equals
- Finish

“Hard-Coded” Generator

by Samet Kurumahmut

"Hard Coded"
Generator

"HARD-CODED" GENERATOR



[Unity API](#) works with hard-coded values (strings, integers, etc.) which can be an evil if they are referenced in many different locations in your project because when you change or remove a hard-coded value through Unity Editor, you are not notified about its references. Also, there may be misspellings which are not also informed to you. *Hard-Coded Generator* is a solution for these problems.

To start using *Hard-Coded Generator*, import `HardCodedGenerator.dll` and `HardCodedGenerator.MenuItems.dll` and go to the menu path of "**Window/Hard-Coded Generator/Generate Config Asset**" to generate *configuration asset*.

Then, change values in generated *configuration asset*. After that, to generate/update *HardCodeds* assembly, use the menu path of "**Window/Hard-Coded Generator/Generate Assembly**" or use *(re)generate* button in *configuration asset* or use "**Ctrl + Shift + Alt + H**" hot-key on Windows / "**Cmd + Shift + Option + H**" hot-key on OS X.

Go to [Website](#) and see more on [Details](#) page.

Support [Website](#).



VERSION HISTORY

Version 1.0.0
Mar 21, 2016

Features

- Generating types (classes) and some utilities (members) by working on the hard-coded values existing in the following Unity features.
 - * Animations
 - * Animator Controllers
 - * Axes
 - * Layers
 - * Resources
 - * Scenes
 - * Sorting Layers
 - * Tags
- Auto-generated assembly (.dll).
- Auto generation after each change in the project.
- Customizable assembly through *configuration asset*.
- Editable hard-coded values via inspector.
- Enabling/Disabling auto generation through *configuration asset*.
- Modifiable menu item path and hot-keys.
- Runs in background.