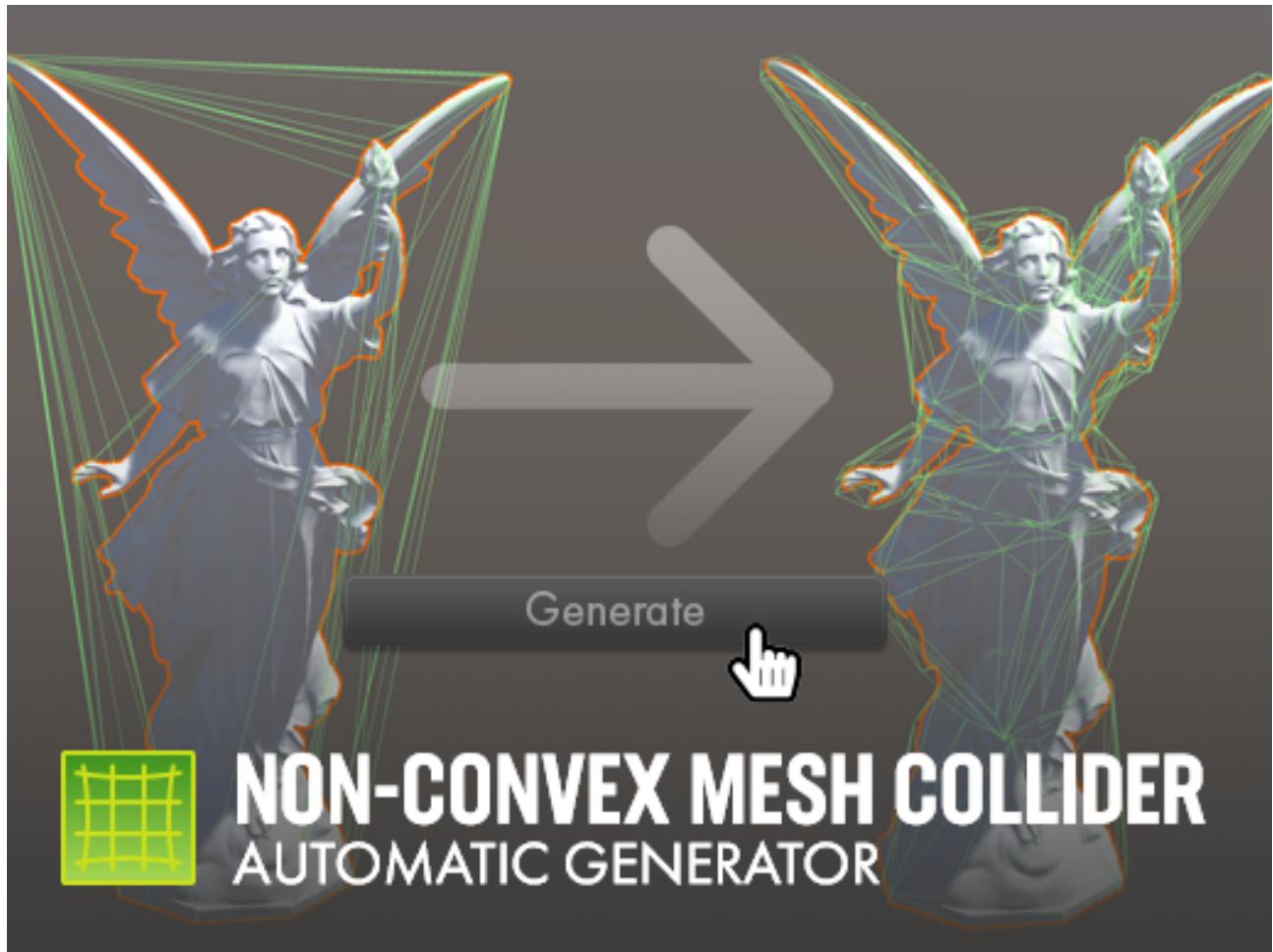


Non-Convex Mesh Collider. Automatic Generator

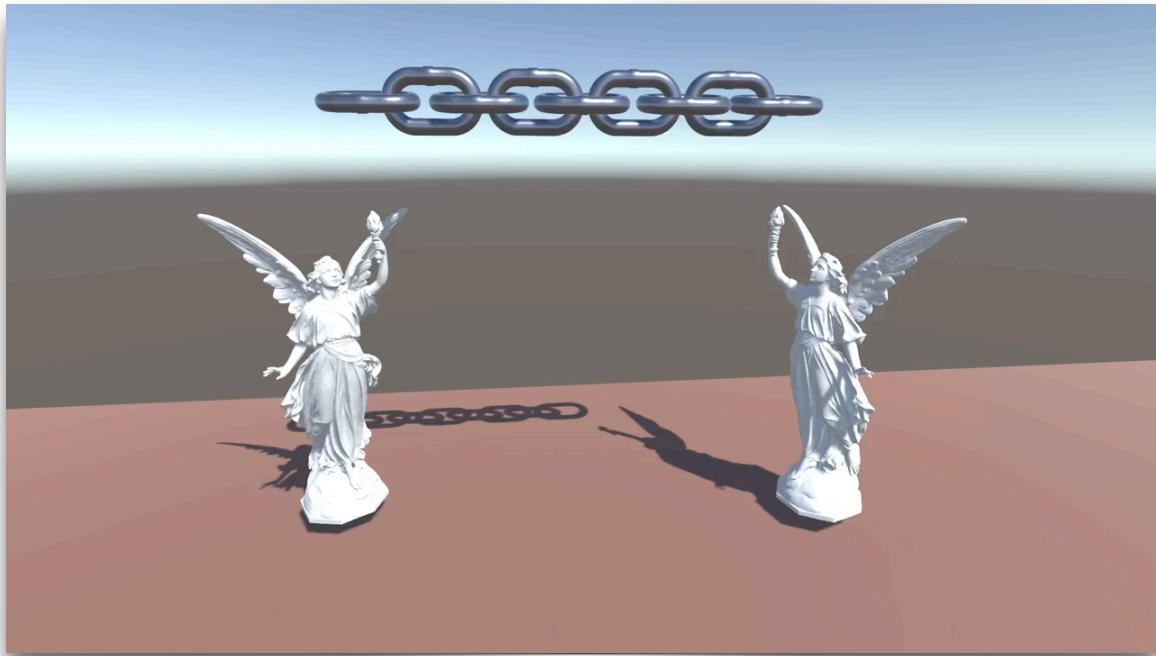
v1.1



For support please email contact@plawius.com

What is the problem? How to solve it?

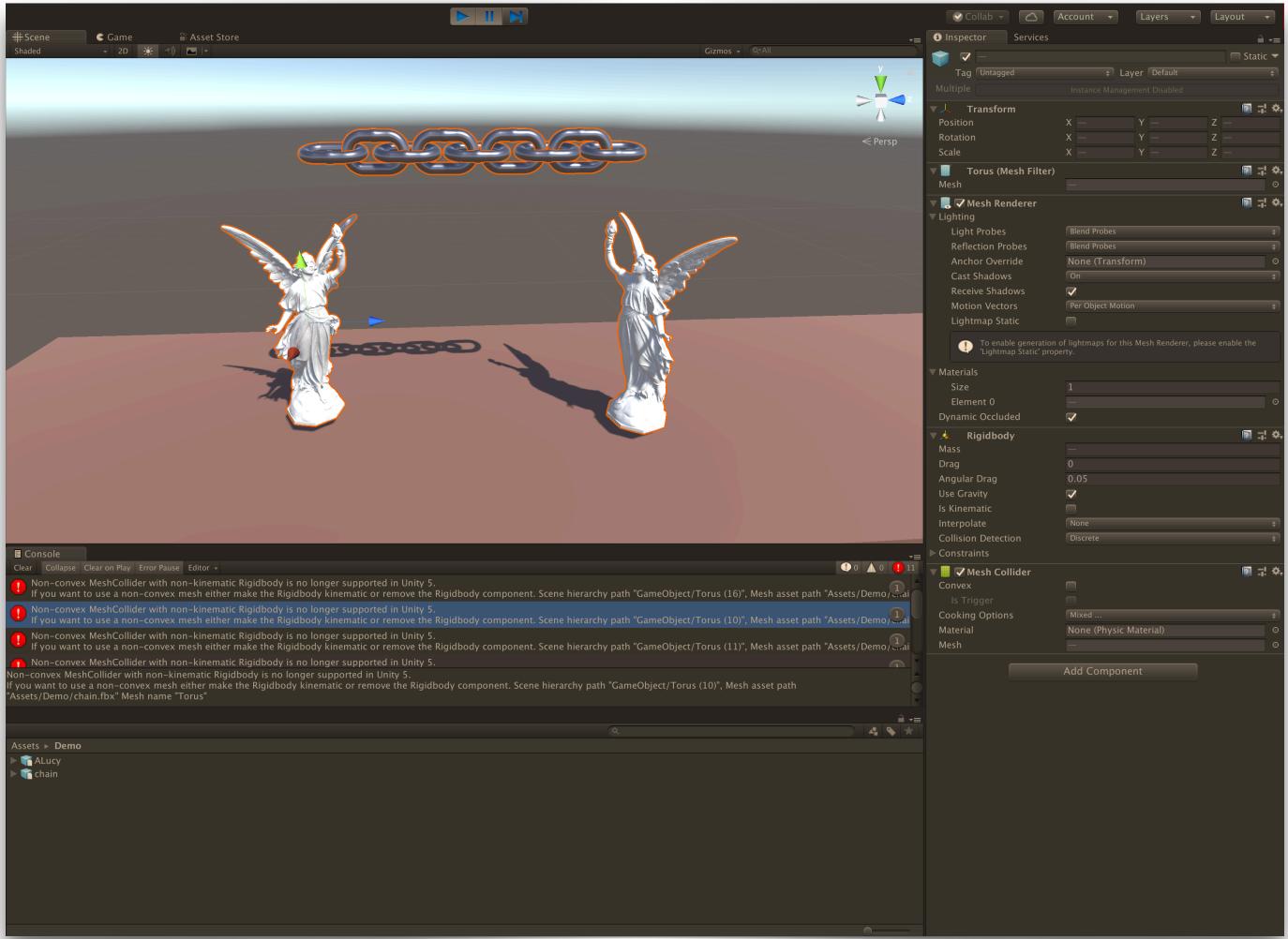
You want to create something like [this video \(click here\)](#):



You added a new object to your non-static geometry. And you want to detect collisions and/or do a physics simulation. But what you have is:



This is (obviously) a really bad approximation of your geometry. Ah, that's because it is marked as Convex, so let's disable that...



No, it doesn't work at all: **since Unity 5** it is no longer supported!

And it's impossible to simulate chain's physics like that at all.

Solution

Non-Convex Mesh Collider. Automatic Generator fixes this for you!

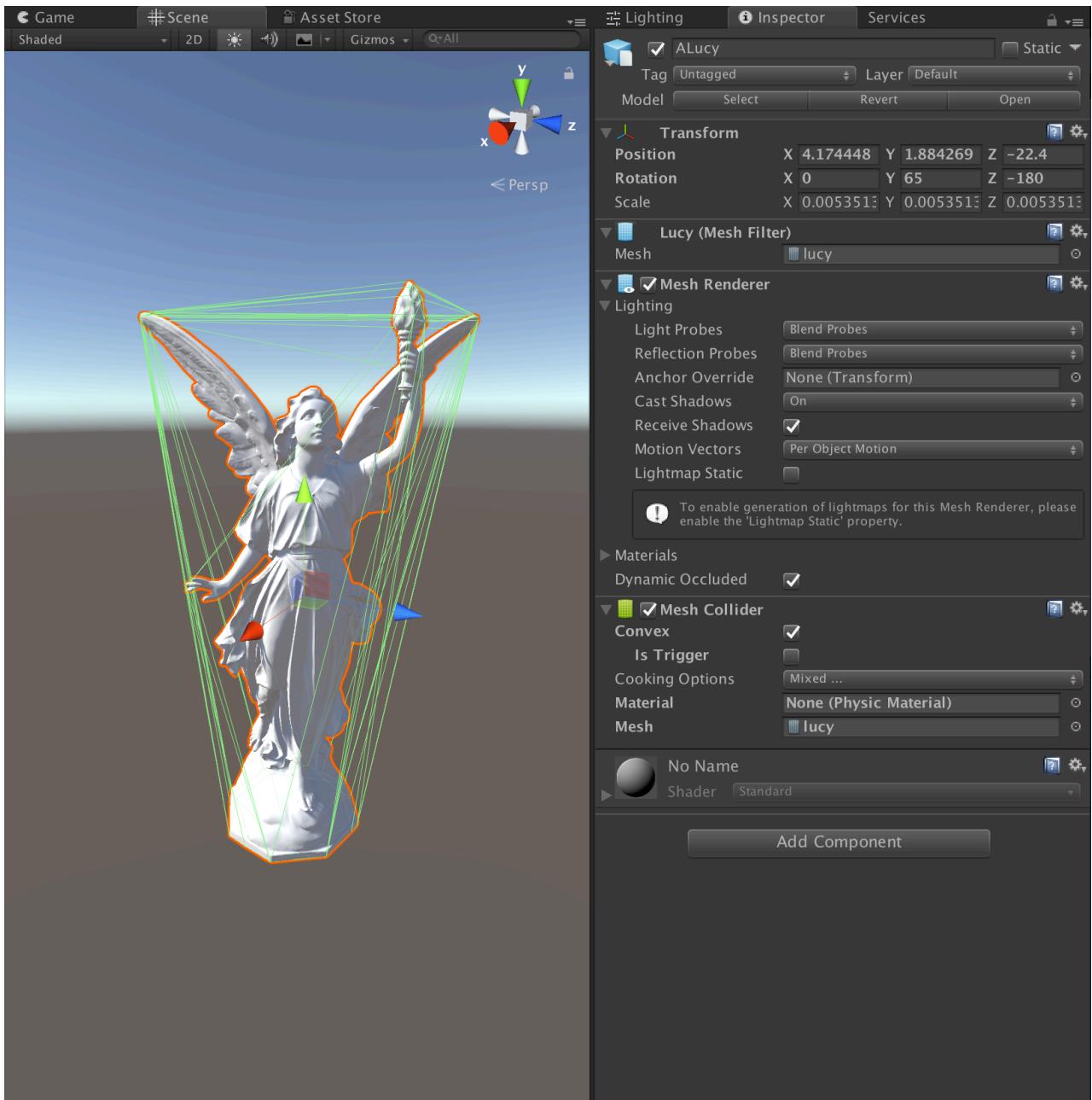


It creates a set of convex colliders that together forms the non-convex collider that you expect. This is industry-standard way of solving this problem and it is used already by other well known engines. It is based on the latest V-HACD 2.0 library.

So, how to use it exactly?

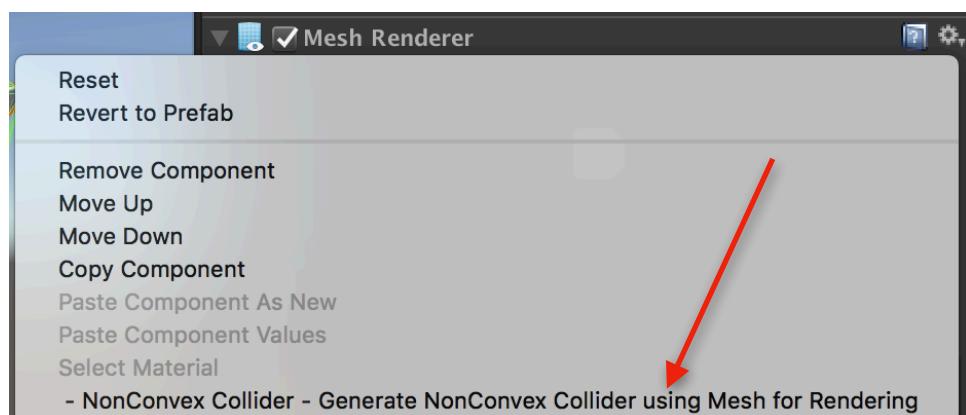
You can use it only in the editor to create a Mesh. It is possible to use via Unity Editor Inspector or via UnityEditor compatible code.

Step #1. Select your object

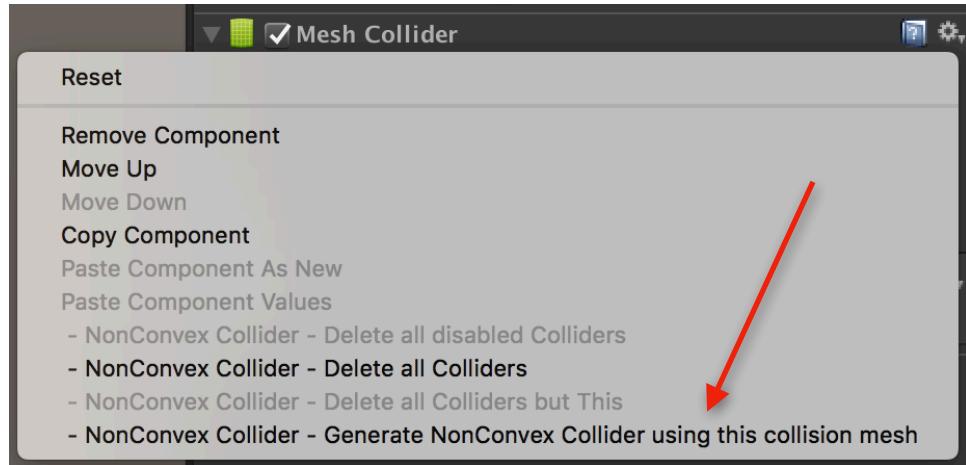


And there are 3 options to create non-convex mesh from this point:

1. Press the gear icon on MeshRenderer or MeshFilter:



2. Press the gear icon  on MeshCollider:



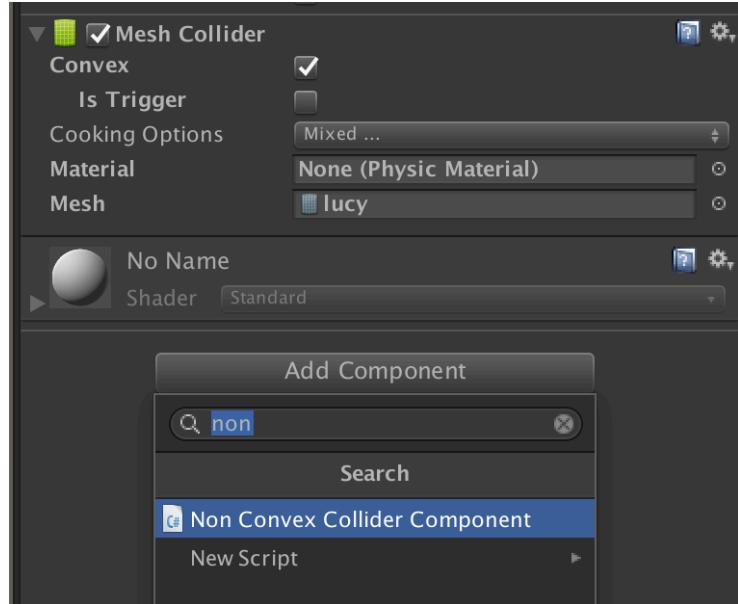
as you can see there are few more options in here.

Note:

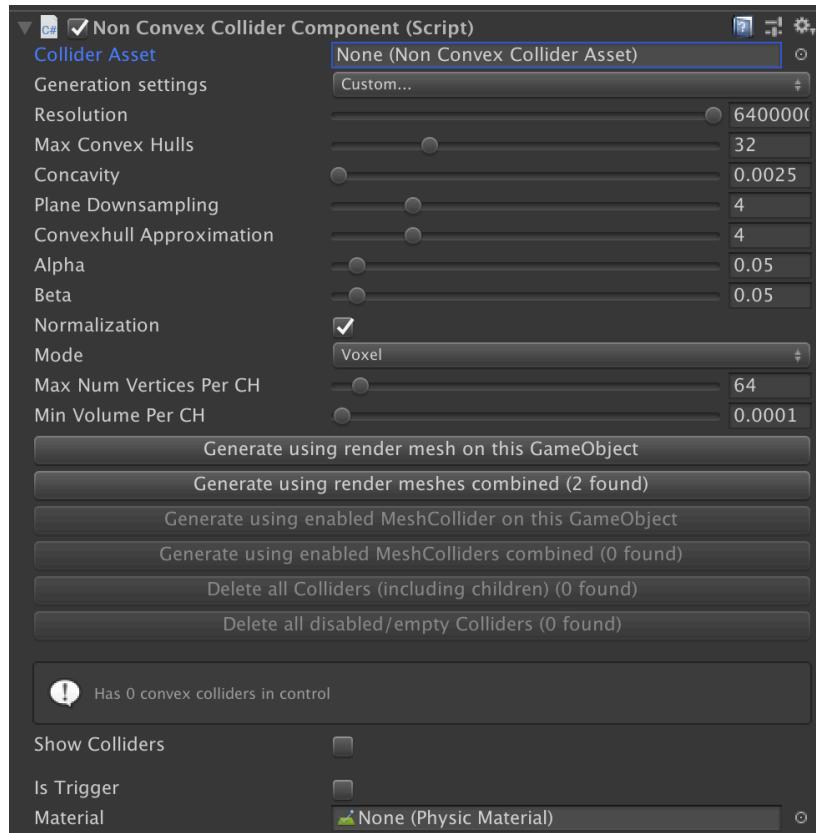
Delete all disabled Colliders,
Delete all Colliders,
Delete all Colliders but This

will delete the colliders, no matter were they created by this asset or not.

3. Add Non Convex Collider component



this provides you more options to tweak:

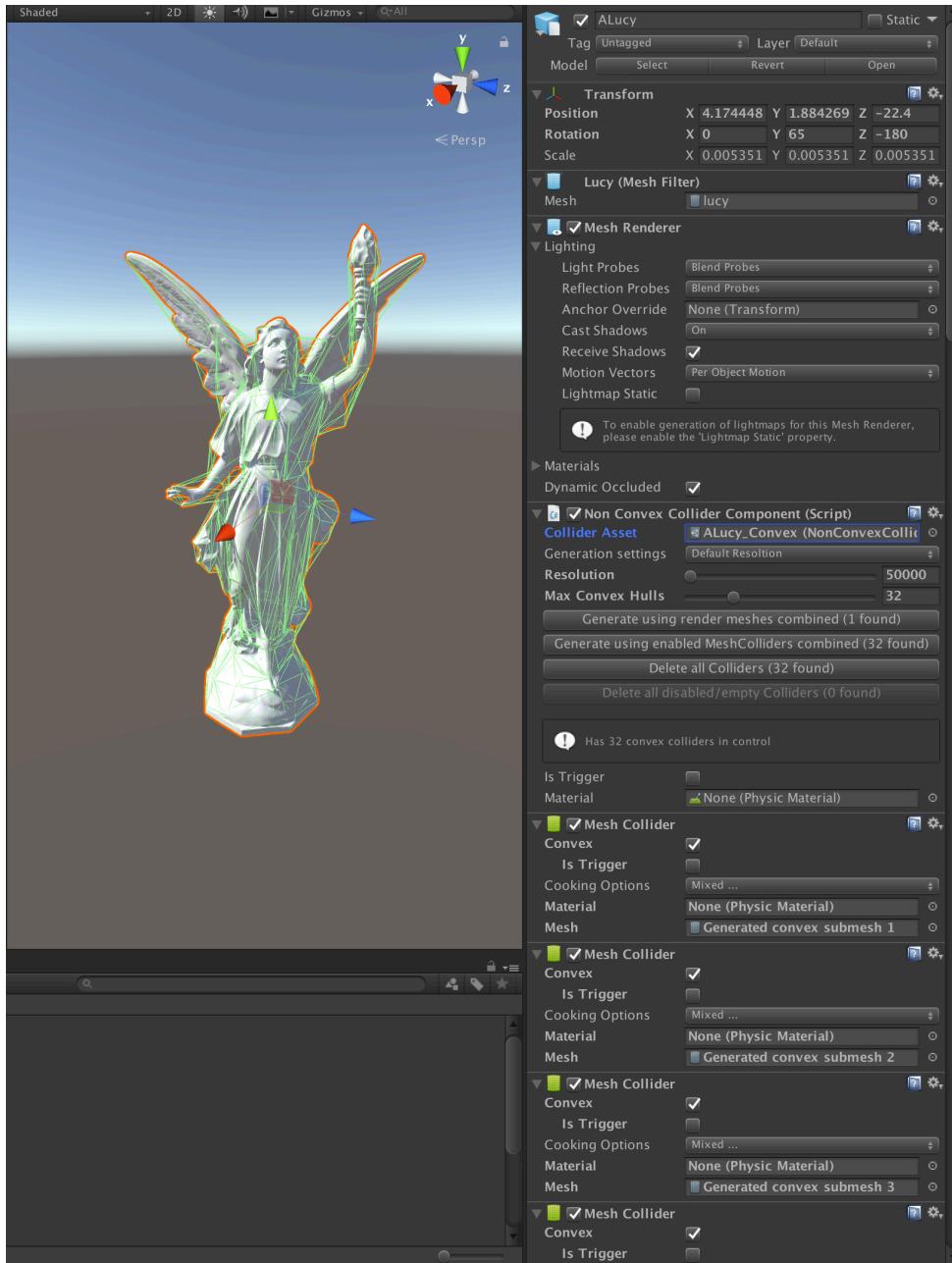


Detailed description of these parameters are [here](#).

If you click **Generate using rendering meshes combined** - it will generate the same way as 1) (if you have only one rendering mesh).

If you click **Generate using enabled MeshCollider combined** - it will generate the same way as 2) (if you have only one enabled rendering mesh).

As a result you will have this:



So as you can see - there are lots of convex mesh colliders. You can easily control them using buttons on Non Convex Collider component. Also if you change IsTrigger or Physics Material on it - all colliders will sync up.

Also as you can see - new Collider Asset was generated that contains all convex colliders information inside (you can drag'n'drop already generated ones, for example). Also next time you want to generate

- it won't re-generate, but just pick already existing asset with the same settings and meshes.

Runtime Generation

It is possible to generate non-convex colliders in runtime, on your procedurally generated meshes. Runtime Demo is included in the package.

```
void Start()
{
    gameObject.AddComponent<Rigidbody>();

    var filter = gameObject.AddComponent< MeshFilter >();
    var mesh = filter.mesh;

    using (var _ = new StopwatchScoped("Generate mesh"))
    {
        GenerateTorusMesh(mesh);
    }

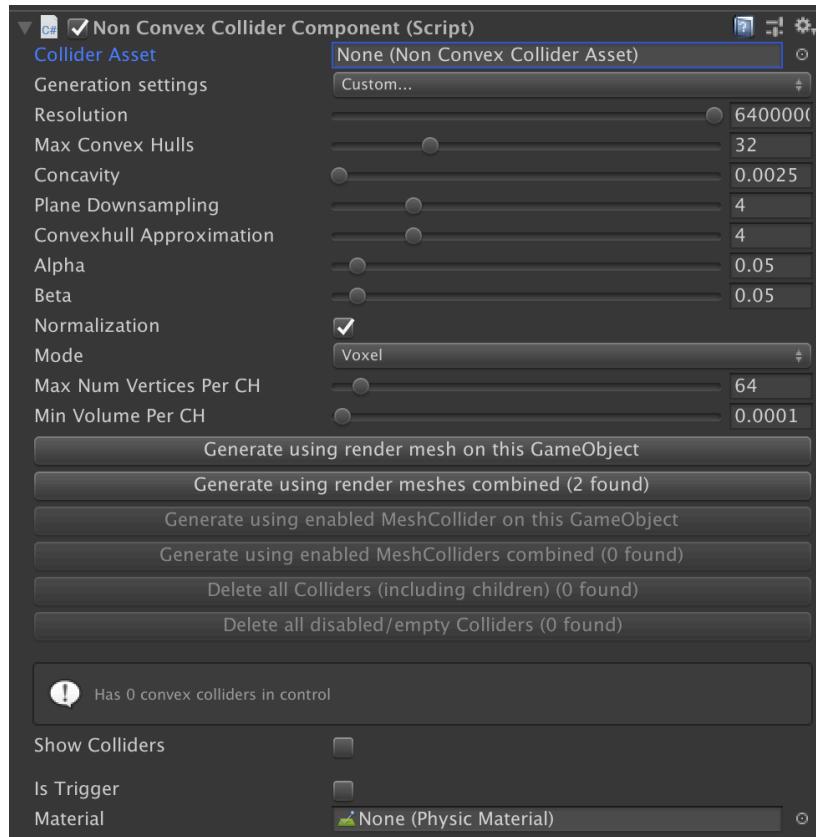
    gameObject.AddComponent<MeshRenderer>();

    Mesh[] meshes;
    using (var _ = new StopwatchScoped("NonConvexCollider generate meshes"))
    {
        meshes = API.GenerateConvexMeshes(mesh, Parameters.Default());
    }

    using (var _ = new StopwatchScoped("NonConvexCollider generate add to gameobject"))
    {
        var colliderAsset = NonConvexColliderAsset.CreateAsset(meshes);

        var nonConvex = gameObject.AddComponent<NonConvexColliderComponent>();
        nonConvex.SetPhysicsCollider(colliderAsset);
    }
}
```

Non-Convex Collider component parameters



Collider Asset - points to the asset that was generated by the generator.

You can remove it (all colliders connected to it will be removed as well)
You can replace it with other one (new colliders will be replaced as well)

Generation settings - you can select a preset for the settings.
Custom... will give most options:

Resolution - the more you select, the slower generation will be, but better result will be generated

Max Convex Hulls - max amount of convex colliders to generate. Try to keep it as small as you can.

Other parameters are tied to V-HACD 2.0 library (<http://kmamou.blogspot.com/2014/12/v-hacd-20-parameters-description.html>)

Generate using render meshes combined - sometimes you have a complex object but you still want to generate a MeshCollider - with this button you can generate it.

Generate using enabled MeshColliders combined - if you already have physics meshes you can combine them and generate a new one.

Delete all Colliders

Delete all disabled/empty Colliders - cleanup operations, sometimes useful. Note: empty collider - collider without mesh or with missing mesh.

When you click **Generate** all existing unknown colliders will be disabled.

Scriptable API

– NonConvexColliderAsset : ScriptableObject

Accessible from the runtime (build)

Asset that you can see in **Collider Asset** on Non Convex Collider component.

Fields

Mesh[] ConvexMeshes

Array of Mesh'es that were generated.

long[] HashOfSourceMeshes

Hashes for the ConvexMeshes + generator parameters

Methods

public static NonConvexColliderAsset CreateAsset(Mesh[] meshes)

Create asset from meshes

public bool SameHash(long[] meshHashes)

Returns true if **meshHashes** is the same as this.**HashOfSourceMeshes**

– NonConvexColliderComponent : MonoBehaviour

Accessible from the runtime (build)

Non-Convex Collider component that is a replacement for classic MeshCollider. Controls all convex MeshColliders.

Fields

List<MeshCollider> ConvexColliders

List of all the children Convex colliders added to this GameObject

bool IsTrigger

The same as IsTrigger for MeshCollider. Will sync it with all ConvexColliders

PhysicMaterial Material

The same as Material for MeshCollider. Will sync it with all ConvexColliders

public NonConvexColliderAsset ColliderAsset

Read only field to get NonConvexColliderAsset used. If you want to set it - use SetPhysicsCollider method

Methods

public void SetPhysicsCollider(NonConvexColliderAsset newColliderAsset)

Deletes all MeshColliders (ConvexColliders), add new from
newColliderAsset (nothing if it's null)

Plawius.NonConvexCollider.Editor.API

*Accessible from the UnityEditor, macOS and Windows builds
Yes, you can generate colliders in runtime since v1.1.0*

public struct Parameters

Structure that contains generator's parameters. See
Parameters.LowResolution(), **Parameters.Default()**,
Parameters.HighResolution().

public static Mesh[] GenerateConvexMeshes(Mesh nonconvexMesh)

Simple version of API. You send a Mesh, get array of Convex meshes that could be used in MeshColliders

public static Mesh[] GenerateConvexMeshes(Mesh nonconvexMesh, Parameters parameters, Action<float, string> progressCallback = null)

Full version of the API. You send a Mesh, get array of Convex meshes that could be used in MeshColliders.

Parameters are V-HACD 2.0 parameters.

Progress Callback could be used to call `DisplayProgressBar`, for example.

Changelog

v1.0.0

Initial release

v1.1.0

Runtime API added

Bug connected to mesh hashing mechanism fixed

Licenses

V-HACD 2.0

<https://github.com/kmammou/v-hacd>

BSD 3-Clause License

Copyright (c) 2011, Khaled Mamou (kmamou at gmail dot com)
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.