http://store.dcsoft.uk/

Rapid Mesh Combiner

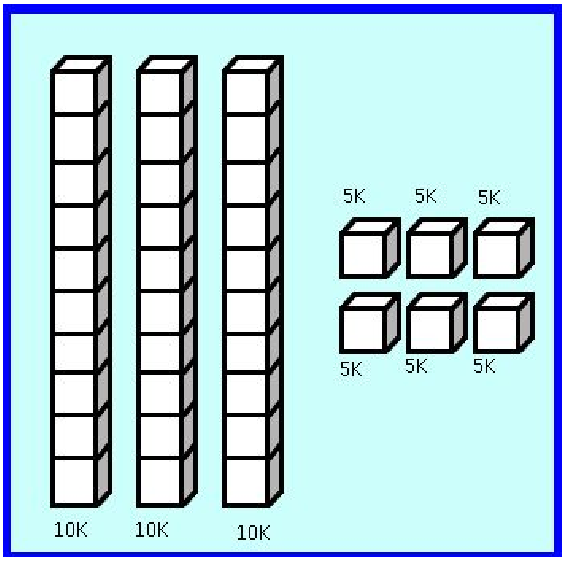
Mesh Combiner and Bundler for Unity.

created by *Cagatay Yapıcı*

Contents

1. Introduction

This package provides the functionality of Mesh Combine and Mesh Bundling .This product is an alternative for Unity's Mesh.CombineMeshes*. Furthermore with Bundling property using Rapid Mesh Combine you can group objects by vertices count.The image which represents the transformation is shown below.



*https://docs.unity3d.com/ScriptReference/Mesh.CombineMeshes.html

2. FAQs

Q: What is difference between Unity's Combine Mesh and Rapid Mesh Combine?

A: The Rapid Mesh Combine does not uses Unity's Mesh.CombineMeshes. Instead of creating submeshes as Unity's CombineMeshes extension method , Rapid Mesh Combine creates a single Mesh.

Q: What is Mesh Bundling?

A:With mesh bundling you can group combined meshes as separate objects , by defining maximum vertices count per object. For example you have lots of meshes in game and you want to combine them, in addition to combine you want them to have maximum 10K vertices. This is what you are looking for.

Q: So this plug-in contains Mesh-Slicing feature to group?

A:Actually no . I don't mean mesh slicing it is a different future project. This is bundling which is defining the maximum vertex count per object.

Q: Can I save the prefabs that I manipulate during run time?

A : You can not save with Rapid Mesh Combine System.It is not designed to do that. But there are tools in the asset store about serialization of runtime instantiated gameobjects.

Q: How is the performance is really fast ?

A: Because of the threads .Carrying out the algorithm to thread is completely frees Unity's main thread so there will be no performance loss.

B: Carrying unsafe code to c# have a little bit help too.

Q: Why the source code of core functionality has been provided as a DLL.

A: The CagLib DLL contains unsafe context codes to gain extra speed. For this reason modification of core codes will cause problems.

Q: Is this multi-threaded?

A: Yes. You can even set the thread count .

Q: Can i integrate to my game ?

A:Yes. You can integrate it . The Rapid Mesh combine is designed for that. You just need to include Combine Controller and Cagmeshlib.dll .

Q: Does it have mesh editor script ?

A:Yes, but you can not combine the gameobject before the game starts.

3. Demos & Videos

Demos:

Included with the package are three demo scenes displaying some of the capabilities of this plug-in. Some details to get you started are presented here:

Bundling:

This scene shows how to bundle meshes at run time.

LayerTag:

This scene shows how to gather resources using gameobject's layer and tag.

Parent:

This scene shows how to how gather resources from parent object.

Videos:

You can access all videos from :

http://store.dcsoft.uk/RMC

Online docs:

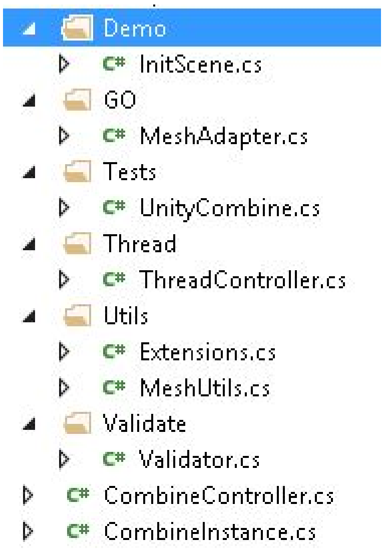
You can access all docs from :

http://store.dcsoft.uk/Documentation

4. Package Contents & Setup

a)Scripts:

The scripts are listed as follows

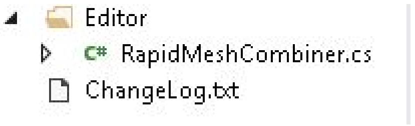


```
◢ Demo
    ▷ C# InitScene.cs
◢ GO
    ▷ C# MeshAdapter.cs
◢ Tests
    ▷ C# UnityCombine.cs
◢ Thread
    ▷ C# ThreadController.cs
◢ Utils
    ▷ C# Extensions.cs
    ▷ C# MeshUtils.cs
◢ Validate
    ▷ C# Validator.cs
▷ C# CombineController.cs
▷ C# CombineInstance.cs
```

InitScene.cs is the prefab creator script for this demo. You can delete it when you integrate Rapid Mesh Combiner with your project.

Extensions.cs is the extension methods scripts especially written for data wrappers widely used in CagLib.dll

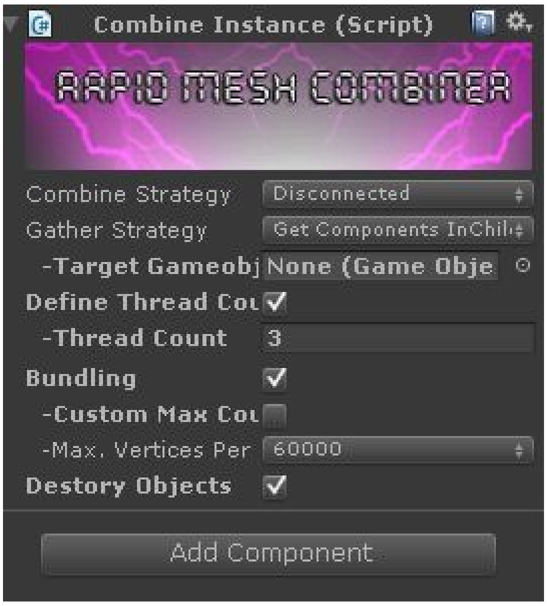
The MeshAdapter.cs script has been used in Disconnected Strategy. The ThreadController is backbone of threading in this project.UnityCombine.cs in Tests folder is about comparing Unity’s combine method with Rapid Mesh Combine.Validators.cs the validation library in this project.The CombineController.cs initializes threading and resources in scene. The Combine instance is the mapping between editor script RunTimeMeshCombiner.cs which has been shown below .



```
◢ Editor
    ▷ C# RapidMeshCombiner.cs
    ChangeLog.txt
```

These wrappers carries the datas from Unity's main thread to parallel threads which are used by ThreadController.cs.

b)Properties of Editor Script:



Combine Instance (Script)

Combine Strategy: Disconnected

Gather Strategy: Get Components InChil

-Target Gameobj: None (Game Obje

Define Thread Cou: ☑

-Thread Count: 3

Bundling: ☑

-Custom Max Cou: □

-Max. Vertices Per: 60000

Destory Objects: ☑

Let's have a look at properties of editor script.

1)Combine Strategy :

There are two types of combine strategies Uniform and Disconnected.

a) Uniform Strategy digests the meshes in a single thread which is parallel to Unity's main thread.
b) The Disconnected Strategy eagerly calculates how much object is needed for completing the mesh combine operation. First creates the gameobjects then digests them with predefined thread count.

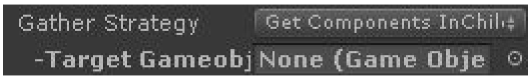
They are both fast and easy to implement feel free to use both of them.

2)Gather Strategy:

There three different types gathering resources using Unity Framework.

a)Get Components in Children :

Uses Unity's GetComponentsInChildren extension method. This is faster but you need to assign the target object .



Gather Strategy: Get Components InChil

-Target Gameobj: None (Game Obje

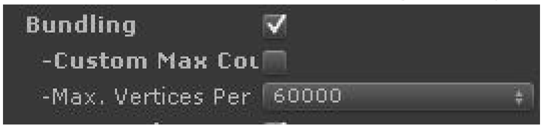
You need to assign the Target GameObject before the game starts because the objects which will be created in run time will be the child of this object.

b)Find GameObjects With Tag:
Uses Unity's FindGameObjectsWithTag extension method .

c)Find GameObjects With Layer:
Uses FindObjectsOfType<GameObject> method. And iterates through the results which match with the desired layer.

3)Define Thread Count :
You can just define thread count when you are using Disconnected Combine strategy othewise thread count will be forced to one .

4) The Bundling Option:
This feature is Rapid Mesh Combiner's one of important features. With this you can limit allowed maximum vertices to combined gameobjects.



Bundling ☑
-Custom Max Cou □
-Max. Vertices Per 60000

The Bundling is to define maximum vertices count for merged objects. For example if you have 100 gameobjects each of which has 3000 vertices. As a result you have 300.000 vertices . Unity's vertice limitation for an object is 65534 . So this 300,000 vertices will be combined into 5 different gameobjects with 65534 vertex count limit. If you define the maximum count using Bundling property to 10,000 vertices then you will have 30 game objects each of which has been bundled with 10,000 vertices.

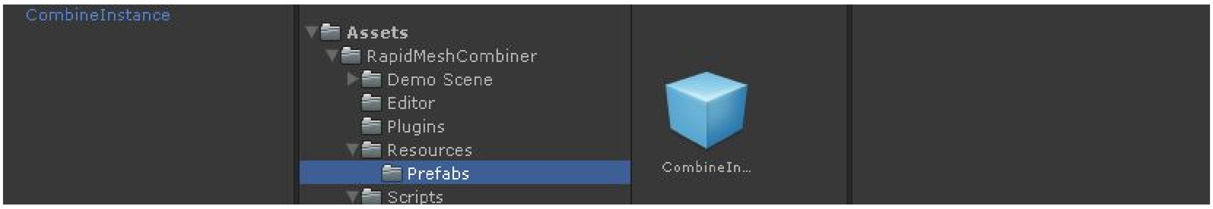
*Caution: The maximum vertices count defined in Bundling must be equal and greater than target gameobjects overall vertex count.Because there is no implementation of mesh split exists in this project , you just need to use exact count or multiples of it.
For instance if you have 100 objects with 100 vertices in scene and you want to bundle them with 90 vertices. You can not do this as the gameobjects meshes won't be splitted. So you need to use 100 and multiples of 100.

target>=x^n or target>=x*n where x is the vertex count,n is positive integer.

5)Destroy Objects:
With this option you can delete the gamobjects has been used in combine process. If not checked Rapid Mesh Combiner will hide used gameobjects by default.

c)Setup:

All you need to do is drag and drop CombineInstance prefab which is located in Resources/Prefabs folder.



CombineInstance
Assets
RapidMeshCombiner
Demo Scene
Editor
Plugins
Resources
Prefabs
Scripts
CombineIn...