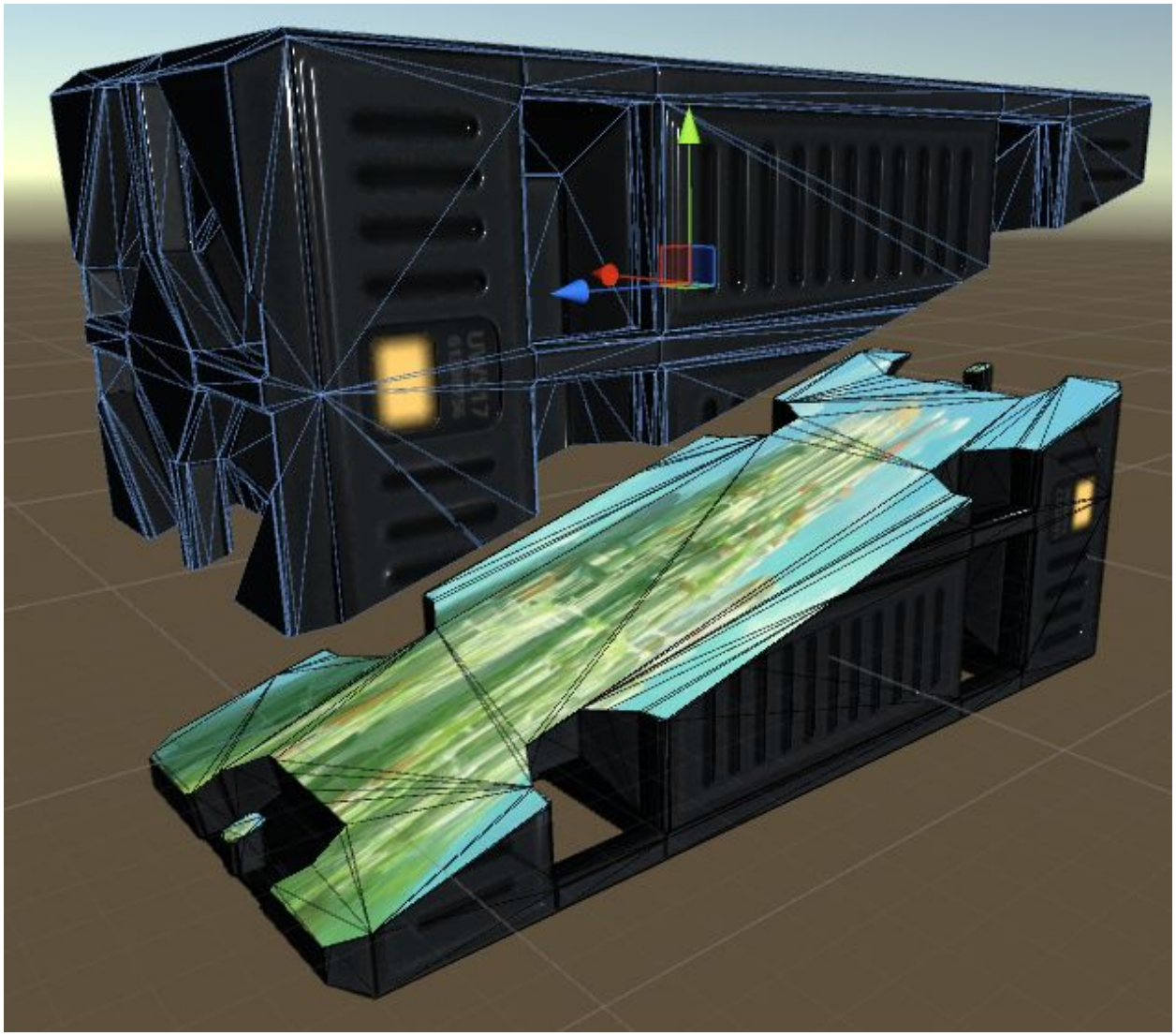
Mesh Slicer



Content

About package

You can slice objects by plane asynchronously. You call Slice method and after a while, a callback method will be called.

Here you can find two folders: ObjectSlicer and ObjectSlicerSamples. In your project you can totally remove ObjectSlicerSamples, because there is only examples of usage.

Package includes a lot of legacy code. So you can read previous version of documentation to work with it: Doc V1.7

You can also try this package on windows link. Here I run a scene with a table and a car.

Performance

Test on quite complex model (vertex count - 17004) takes ~ 55 milliseconds of time on my core i7 2.2GHz in release.

How does it work?

Each sliceable object in game have to implement 'IBzSliceableAsync' interface:

```
/// <summary>
/// Asynchronously sliceable object
/// </summary>
public interface IBzSliceableAsync
{
      /// <summary>
      /// Start slicing the object
      /// </summary>
      /// <param name="plane">Plane by which you are going to slice</param>
      /// <param name="sliceId">To prevent multiple slice requests you should
use sliceId</param>
      /// <param name="callBack">Method that will be called when the slice
will be done</param>
      void Slice(Plane plane, int slicdId, Action<BzSliceTryResult> callBack);
}
```

The IBzSliceableAsync interface implemented in abstract BzSliceableObjectBase class which you have to inherit. Example of inherited class is ObjectSlicerSample.

How to slice:

1) Get sliceable component: var sliceable = GetComponent<IBzSliceableAsync>();
2) Create plane by which you are going to slice: var plane = new Plane(...);
3) Slice: sliceable.Slice(plane, sliceId, null);

To Slice method you need to pass sliceId to avoid slicing same object several times with one swing. So it is better to create slice id manager to get new unique id.

Implementations of BzSliceableObjectBase

In your class you have to make your implementation of abstract BzSliceableObjectBase class. It requires you to implement this methods:

- protected abstract BzSliceTryData PrepareData(Plane plane) - Prepare data before slice

- protected abstract void OnSliceFinished(BzSliceTryResult result) - Will be called when the slice will be finished
- And optionally you can overload protected virtual void StartWorker(Action<object> method, object obj). For example if you want to use your thread pool which is a good practice.

ObjectSlicerSample

For example, you can see sample implementation of ObjectSlicerSample class:

```
public class ObjectSlicerSample : BzSliceableObjectBase
{
	protected override BzSliceTryData PrepareData(Plane plane)
	{
		// remember some data. Later we could use it after the slice is done.
		// here I add Stopwatch object to see how much time it takes
		// and vertex count to display.
		ResultData addData = new ResultData();

		// count vertices
		var filters = GetComponentsInChildren<MeshFilter>();
		for (int i = 0; i < filters.Length; i++)
		{
			addData.vertexCount += filters[i].sharedMesh.vertexCount;
		}

		// remember start time
		addData.stopwatch = Stopwatch.StartNew();

		// colliders that will be participating in slicing
		var colliders = gameObject.GetComponentsInChildren<Collider>();

		// return data
		return new BzSliceTryData()
		{
			// componentManager: this class will manage components on sliced objects
			componentManager =
				new StaticComponentManager(gameObject, plane, colliders),
			plane = plane,
			addData = addData,
		};
	}

	protected override void OnSliceFinished(BzSliceTryResult result)
	{
		// on sliced, get data that we saved in 'PrepareData' method
		var addData = (ResultData)result.addData;
		addData.stopwatch.Stop();
```

```
                drawText += gameObject.name +
                        ". VertCount: " + addData.vertexCount.ToString() + ". ms: " +
                        addData.stopwatch.ElapsedMilliseconds.ToString() + Environment.NewLine;
        }

        static string drawText = string.Empty;

        void OnGUI()
        {
                GUI.Label(new Rect(10, 10, 2000, 2000), drawText);
        }

        // DTO that we pass to slicer and then receive back
        class ResultData
        {
                public int vertexCount;
                public Stopwatch stopwatch;
        }
}
```

Slice objects with a knife

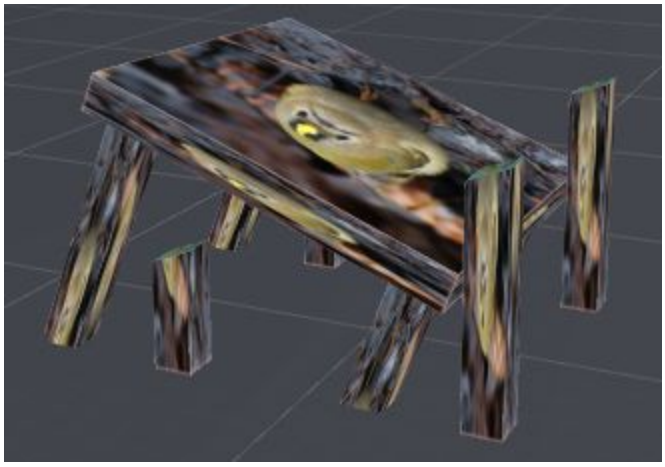
In sample folder you and find two files “KnifeSliceableAsync” and “BzKnife”:

- KnifeSliceableAsync.cs - for objects that you want to slice via knife (e.g. an apple);
- BzKnife.cs - for object that you want to slice with (e.g. knife).

KnifeSliceableAsync must be attached to objects with rigidbody on it.

BzKnifeSliceable

Simple implementation that slices mesh.
Test scene: “BzKovSoft/ObjectSlicerSamples/testSceneAsync”. If you click on Table in Game window, it will be sliced to two objects. And every time you click on it, it will be sliced again:



Important to note, table sliced only into two objects, so legs at the bottom will be connected.

BzSliceConfiguration

If your sliceable object contains of multiple renderers, you can configure each differently by applying BzSliceConfiguration script to it.
Here you can configure slice material and slice type.
Slice types:

- Slice - object will be sliced
- Duplicate - object will be duplicated after slice
- KeepOne - object will not be sliced, and it stays only on one side of the plane

Extensions

If you attach an object that implements an IBzObjectSlicedEvent interface, is will be invoked after slicing:

```
public interface IBzObjectSlicedEvent
{
        void ObjectSliced(GameObject original, GameObject duplicate);
}
```

The method ObjectSliced will be invoked after successful slicing where it passes old object and new instantiated.

BzFixMass

BzFixMass - is example implementation of IBzObjectSlicedEvent where it fixes weight and center of the mass of sliced objects.

BzFixMassSmart

Correctly fixes weight and center of the mass of sliced objects. It works correct only for closed objects.

BzReaplyForce

If you slice an object, it creates two new objects that do not inherit velocity and angularVelocity. This class fixes this problem.

BzDeleteSecondJoint

Suppose you slice an object that attached to another object via Joint. But the two duplicated copy will both have their own Joint. The script delete Joint from the farthest object from the anchor.

Problem zones (issues)

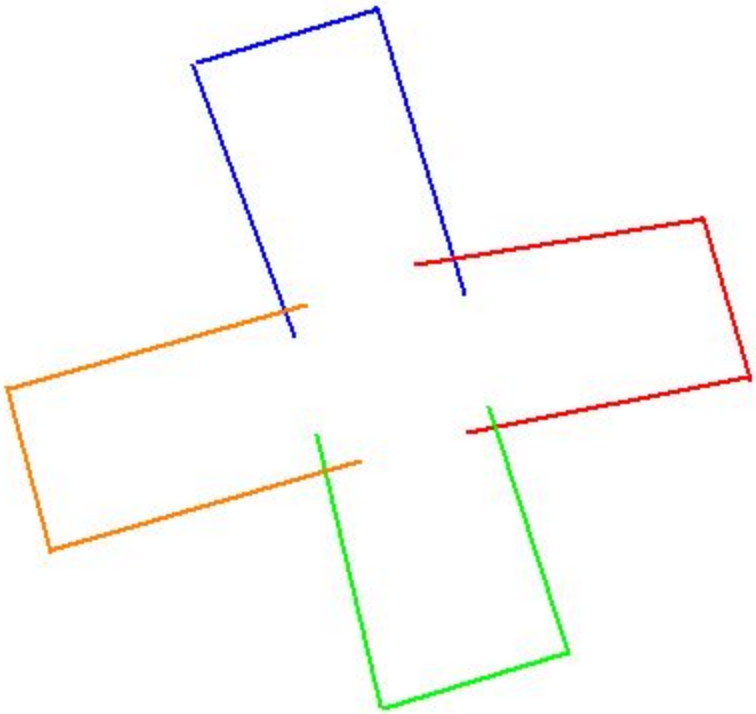
Of course, I'm working to get rid of problem zones. But they are still exists. And there are some that I noticed:

Skinned Mesh Renderer

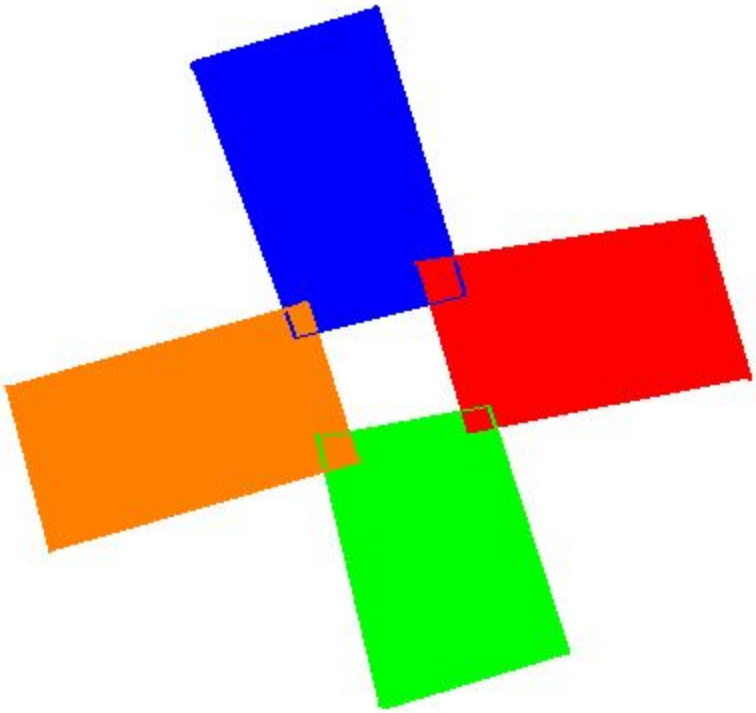
It does not support **SkinnedMeshRenderer**. It only supports **MeshRenderer**.

Different sequences

If section view looks like this:

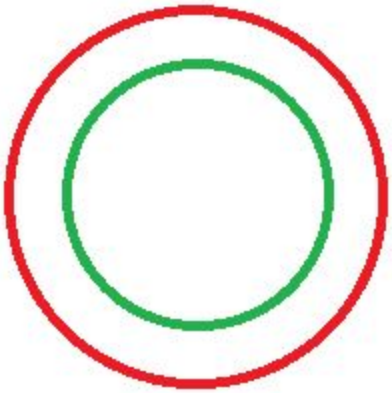


Each connected sequences of vertex will be joined to polygon. The result is 4 polygons, but the center stays empty:

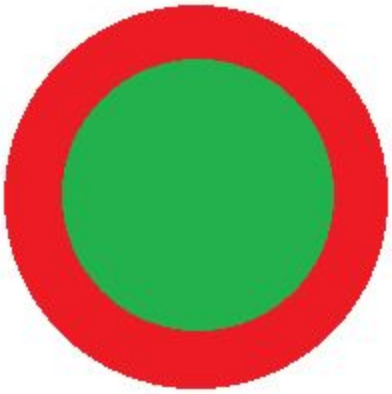


Tube

Another problem can occur with tube. For example, if you slice this tube:



You are going to have two polygon applied to it. Result:



So the whole in the tube will be closed.