atlasToCube documentation

- Overview
- Abbreviation
- Description
- Examples
    - Work In editor
    - Change faces from code
- Source code
- Feedback
- Release notes

Overview

This is an unity3d utility, which allow to assign any image from texture atlas to any frame to your primitive.cube

Abbreviation

- **atlas** - is a large image containing a collection of sub-images, or "atlas" which contains many smaller sub-images, each of which is a texture for some part of a 3D object.
- **inspector window** - window which appeared in unity editor, below information about transform.position and rotation, where usually you modify public parameters in your custom scripts

Description

This script won't help you to create atlas, so use external tools to do it

Using atlases, reduce draw calls. And it's convenient to take any sub-image in atlas and assign as a texture for any face on your cube.

Atlas must have equidistant rows and equidistant columns.

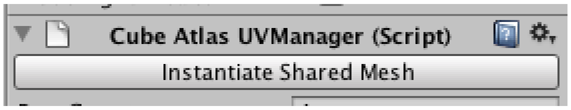
To assign images to the faces, script modify mesh uv cooridnates.

You need to create a new material and put your atlas as texture on it.

In editor mode unity does not allow to modify mesh, because of that we can work only with sharedMesh.

When you add cube to the scene, all cubes shared their meshes, and only when you run your scene, unity create separated instances.

If you want to work with 2 or more cubes in your scene and assign textures independently you have to instantiate the mesh, to do that, click "instantiate mesh" button in inspector.



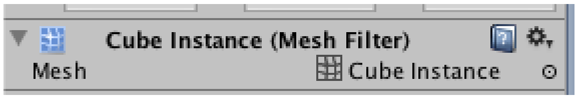
When you will create prefab with modified mesh, script will generate mesh instance for you and will store it as an asset in the same folder as your prefab.

Examples

You will find 2 scenes with examples in the **/Assets/Demo/Scene** folder

Work In editor

1. You already should have exported texture and created material
2. In unity menu click **GameObject/CreateOther/Cub**
3. Drag **CubeAtlasUVManager** script from **/Assets/Resources/Scripts/Util/Mesh** folder to created cube
4. Add your material with atlas as texture to your cube (you can use material from demo folder **Assets/Demo/Material/CubeMaterial)**
5. That's it, now in inspector you will have **Cube Atlas UVManager** panel, and then you will change any of your face details in inspector, you will se that it reflected in editor
    a. **Instantiate Shared Mesh** button, allow to instantiate mesh (you need to use it, if you want to modify several cubes in your scene in editor)
    b. **RowCount** - number of rows in your atlas
    c. **Column Count** - number of columns
    d. **face**
        i. **position** position in the atlas, e.g. if your atlas has 4rows and 4 columns 0 - is top left position, 3 top right, 4 first on 2d row from top.
        ii. **rotation** this number multiplied to 90 degree (you can rotate image on your face)
6. Now add 2d cube and repeat step 3 and and 4 or just copy your existing cube in inspector, change position, to be able see both objects
7. If you will change any faces on one cube, you will se that face for another one is changed as well, the reason of this behavior - cubes use the same shared mesh, to make changes separately please follow to next step
8. Take any cube and click "Instantiate Shared Mesh" in inspector, after that you will se that **CubeMesh inspector** now has link to **CubeInstance**, onother cube still linked to **Cub**

    Cube Instance (Mesh Filter)

    Mesh Cube Instance

9. Try to change cube face on any objet and you will see that now face change independently.
10. try to save your scene, if you will face a message *Cleaning up leaked objects in scene since no game object, component or manager is referencing them Mesh has been leaked 43 times.* just ignore it, it won't harm your project
11. Try to create a prefab with one of your **cube** which has **InstantiatedMesh,** you will find that mesh asset with a prefab name have been created.

Change faces from code

In **/Assets/Demo/Scene** you will find **Spawn.random.faces** scene,

It use **SpawnPrefabController** script, if you will run this scene you will see how many cubes with rigged body are falling to the ground, all of them has faces with random images. This script use prefab from **/Assets/Demo/Prefab** folder.

Just open this script and you will find how to instantiate object with a different faces

```
using UnityEngine;
using System.Collections;

public class SpawnPrefabController : MonoBehaviour {
 public GameObject prefab;
 public int numberOfObjects=10;
 public float rndFactor=1.2f;
 public float timeout=0.5f;
 public int atalasRandomIndexMax=10;

 private GameObject parentGO;
 // Use this for initialization
 void Start () {
  StartCoroutine("startSpawning");
 }
 IEnumerator startSpawning ()
 {
  if (parentGO==null){
   parentGO=new GameObject("GeneratedObjects");
  }

  for (int i=0;i<numberOfObjects;i++){
   Vector3 spawnPosition=transform.position;
   float rnd=Random.Range(-rndFactor,rndFactor);
   spawnPosition.z+=rnd;
   spawnPosition.x+=rnd;
   GameObject go=Instantiate(prefab,spawnPosition,Quaternion.identity) as
GameObject;
   go.transform.position=spawnPosition;
   CubeAtlasUVManager cm=go.GetComponent<CubeAtlasUVManager>();
   if (cm==null){
    throw new System.Exception("Prefab must have CubeAtlasUVManager on it");
   }
   cm.randomizeFaces(10);
   go.transform.parent=parentGO.transform;
   yield return new WaitForSeconds(timeout);
  }
 }
}
```

key comands in this listing is the following

```
GameObject go=Instantiate(prefab,spawnPosition,Quaternion.identity) as GameObject;
//instantiate prefab here (which must already have CubeAtlasUVManager on it)
CubeAtlasUVManager cm=go.GetComponent<CubeAtlasUVManager>();
cm.randomizeFaces(10); //this command will assign random faces in diapason 0-10, and
update mesh uv.
```

Source code

Latest source code is available here https://github.com/nicloay/atlasToCube

Feedback

I'll be glad to receive your feedback, feature request, or bug report, by email nicloay [a] gmail.com or on unity forum thread forum.unity3d.com

Release notes

Change history

- **v1.0** (initial release)
- **v1.001** (10/10/2012)
  - New Event system
  - Mesh instantiation in editor mode
  - Cleaning the code
- **v1.002** (10/10/2012)
  - fixed tangents for instantiated meshes
  - fixed prefab creation (when you create a prefab, mesh asset create and link to object automaticaly)
- **v1.003** (11/10/2012)
  - CRITICAL bug fix, "back face share image with front face" (thanks mbolt from unity forum who posted it)
- **v1.004** (11/10/2012)
  - fixed "rotation/position"
  - provided documentation in README.txt