

# Texture Sequence

Version 1.3.0



**Infinity Code, 2013**

<http://www.infinity-code.com/>

## Description

Component is designed to automatically create an animation from a set of images, with the ability to manually add items, automatic loading of folders «Resources», or nested folders and it, as well as external folders. The same component can work with texture atlases created for Starling.

## Using

Select the menu item «**Component / Infinity Code / Texture Sequence**», to add a component to the selected GameObject. As the target will be automatically selected current GameObject, determine whether he components GUITexture or Renderer to display the images.

If not found any GUITexture, or Renderer, then the component will display a warning.

Parameter «**Textures**» indicates where the image will be taken:

- **None** - then load the texture, you can use the code.
- **Manual** - the texture, you must manually add the appropriate fields. To delete a texture, set it to none.
- **Atlas** - will use textures atlas for Starling. You must specify a texture atlas, XML-configuration file and the original image size.
- **from Resources folder** - textures will load in your path, relative to «**Assets / Resources**».
- **from External folder** - textures will load the specified path. The path can be absolute or relative.

If the parameter «**Textures**» select «**from Resources folder**» or «**from External folder**», in parameter «**Load**» you can tell when will load texture.

In some modes, the component may appear button «**Find textures**». This means that the available texture to find and index by clicking on the button. Indexed texture, you can see in the «**Number of Textures**».

The parameter «**Duration**» set the time in seconds the entire animation.

Parameter «**Speed**» time scales. For example, where a value 0.5 animation will last 2 times longer than it should go. A value of -1, the animation will go to the original speed, but in reverse order.

Parameter «**Autostart**» is responsible for starting the animation when switching to play mode. If the value is not set, then the animation will need to run the program code.

Parameter «**Loop**» indicates the type of repeating animation. Can be:

- **None** - after all the frames of the animation will be shown, playback stops.
- **Loop** - after all the frames of the animation will be shown, the playback will start from the beginning.
- **PingPong** - after the animation is finished, will be played in reverse.

If you select «**None**», then the set value «**AutoDestroy**», after the animation complete component is automatically destroyed.

If the target is specified **«Material»**, you can specify MaterialID and parameter, which will display the image.

Use the **«Set first image to target»** to put the first texture to the selected location to display.

### Redefining the component

There are situations when you need to bring animation to a different GameObject. To do this, select the property to GameObject **«Target Object»**. If you change the target will be detected automatically, display images on GUITexture or Material.

### Working with texture atlases

Texture atlas - a collection of images, fused into one large image. Each part of the image has a name, location and size.

In order to use the atlas texture as the source image, choose the parameter **«Textures»** meaning **«Atlas»**.

Set values:

- In the **«Atlas»** add a texture atlas.
- In the **«Settings (XML)»** configuration file atlas.
- In the **«Original image size»** the original size of the image.

In the Import Settings set the value of the texture atlas **«Texture type - Advanced»** and enable the checkbox **«Read / Write Enabled»**. If this is not done, the component displays a warning.

The settings file is an XML atlas file types:

```
<TextureAtlas>
  <SubTexture name="Name_01" x="0" y="0" width="477" height="486"/>
  <!--   MORE VALUES   -->
  <SubTexture name="Name_XXX" x="477" y="0" width="477" height="486"/>
</TextureAtlas>
```

If you create an animation using Adobe Flash, the file will be created automatically when you export texture atlas.

### Working with sound

Set in parameter **«Sounds sync»**, as you want to sync audio: by frame or by time. Click **«Add sound»**, to add a new sound. In a field with an index of sound, place the sound, and set the value when it should start.

## Links

**Product page:** [http://infinity-code.com/products/texture\\_sequence](http://infinity-code.com/products/texture_sequence)

**API Reference:** <http://infinity-code.com/docs/api/texturesequenc>

## Example of use

### C#

```
public Texture[] pictures;

void OnGUI()
{
    if (GUI.Button(new Rect(5, 5, 300, 30), "Create dynamic texture sequence C#"))
    {
        GameObject go = new GameObject("Grandfather");
        go.transform.localScale = new Vector3(0, 0, 1);
        go.AddComponent<GUITexture>();

        Rect layout = new Rect(Random.Range(0, Screen.width - 120),
                                Random.Range(0, Screen.height - 120), 120, 120);
        go.guiTexture.pixelInset = layout;
        TextureSequence.CreateAndPlay(go, pictures, 1.5f, 1, TextureSequenceLoop.loop);
    }
}
```

### JavaScript

**Important. Before using the component together with the language of JavaScript, you must put the component files in a folder «Standard Assets».**

```
var pictures:Texture[];

function OnGUI()
{
    if (GUI.Button(new Rect(5, 45, 300, 30), "Create dynamic texture sequence JS"))
    {
        var go:GameObject = new GameObject("Grandfather");
        go.transform.localScale = new Vector3(0, 0, 1);
        go.AddComponent("GUITexture");

        var layout:Rect = new Rect(Random.Range(0, Screen.width - 120),
                                    Random.Range(0, Screen.height - 120), 120, 120);
        go.guiTexture.pixelInset = layout;
        TextureSequence.CreateAndPlay(go, pictures, 1.5f, 1, TextureSequenceLoop.loop);
    }
}
```