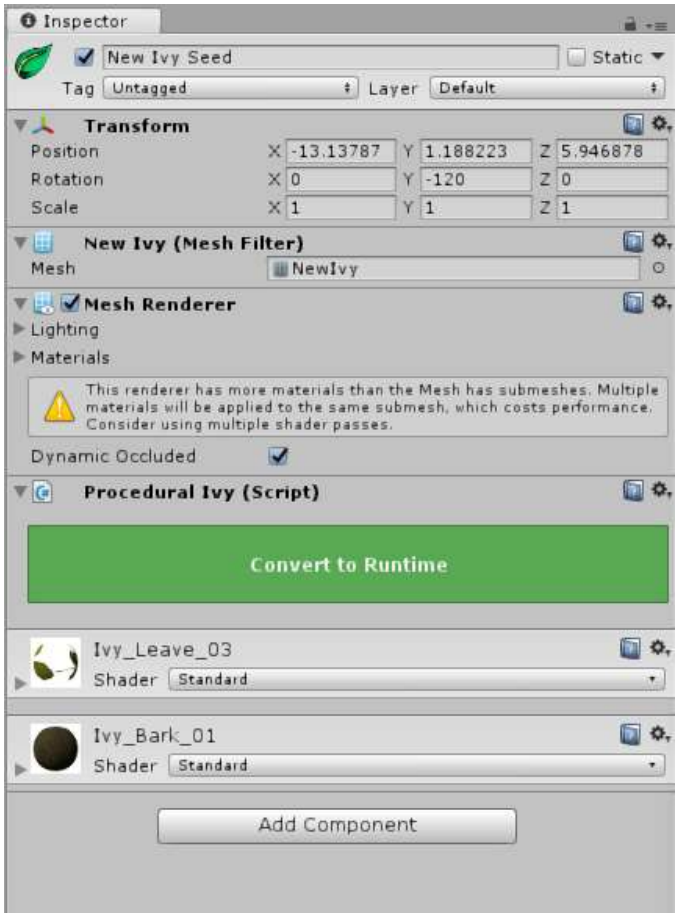
Real Ivy Manual Update (v1.2)

New feature: Runtime Ivies

With the new feature you will be able to spawn ivies and climbing plants during the runtime of your games. You can let them places in your scenes and initialize them during the runtime, or you can create them at any position during your game.
This is a very powerful feature for create stunning effects.



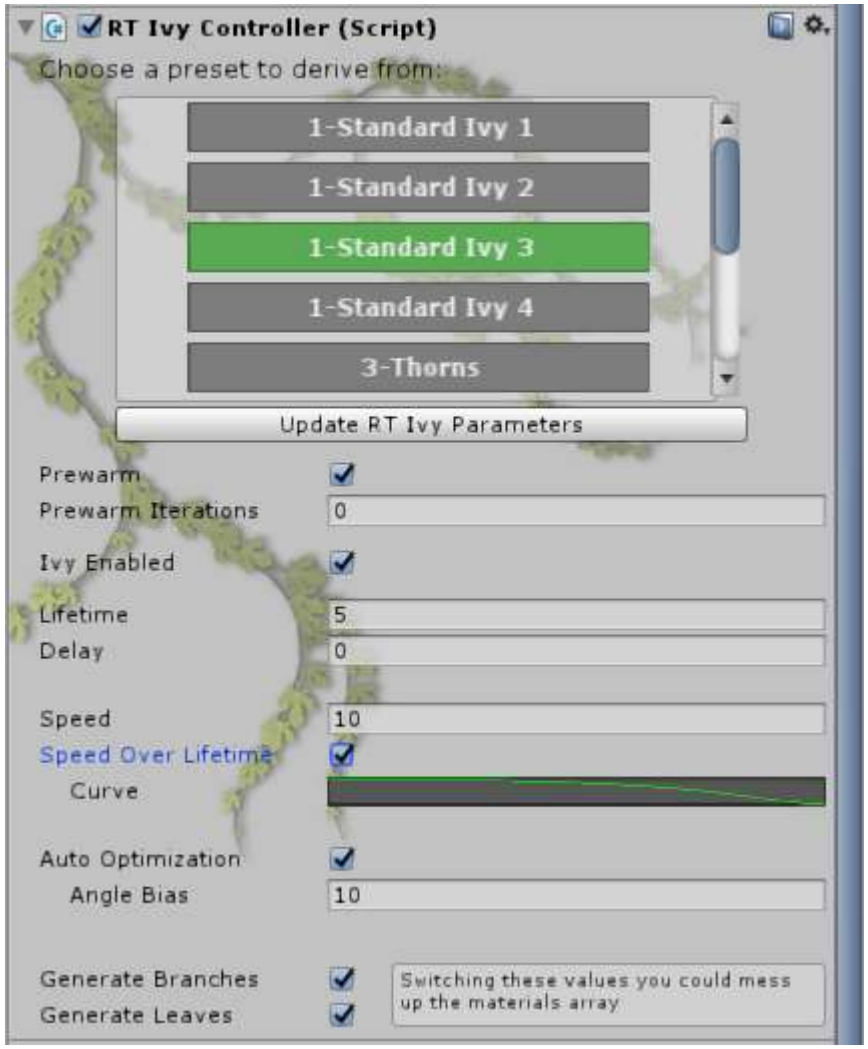
After creating a **Procedural Ivy** with the preset and settings you desire, you'll see a big green button in a component of the created GameObject. The button says "**Convert to Runtime**"

When you push this button, this GameObject will become a RuntimeIvy. You will have a few new components:

- Runtime Ivy. Here are the settings of the path of the growth.
- Runtime Ivy Geom. Here are the settings of the shape of the branches.
- Runtime Ivy Leaves. Here are the settings relatives to the leaves

(These settings are the same that the settings in the Real Ivy Window. For more information about them, go to the Real Ivy Manual 1.0)

- Runtime Ivy Controller. Here there are new options. We take a look in depth in the next page.



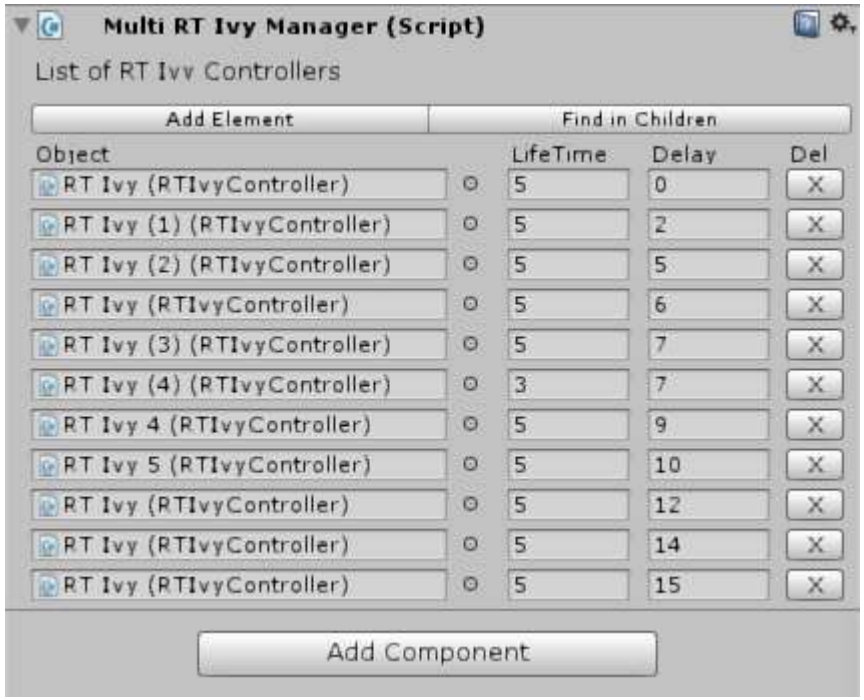
RT Ivy Controller

- In the top you have a **Preset List**. You can change the original preset. You have to push the button "**Update RT Ivy Parameters**"
- If **Prewarm** is enabled, the ivy will spawn grown up.
- The **Prewarm Iterations** are the number of iterations that the ivy will grow at the spawn time.
- **Ivy Enabled** will activate or deactivate the ivy. When prewarm is active, you usually want this set off.
- **Lifetime** is the time that the ivy will be growing up.
- **Delay** is the time the ivy will wait before starting the growth.
- **Speed** is the growth speed for this ivy.
- **Speed Over Lifetime Curve** is the curve that will lead the speed over the life time of the ivy, where speed will be the maximum speed.
- **Auto Optimization** whether the ivy should optimize its geometry or not.
- The higher the **Angle Bias**, more optimized will be the result
- **Generate branches** and **Generate leaves** don't need explanation, but you need to know that if you change their values, you have to manually set correctly the materials array on the Mesh Renderer.

Coordinating multiple ivies

If you need to coordinate the growth of multiple ivies, you have this tool at your service.

- Add the component **MultiRTIvyManager** to a GameObject and from here you can have access to the LifeTime and the Delay of multiples ivies.
- If you push the button "**Find in Children**" the component will automatically search in the hierarchy of the GameObject with the component attached and add all the ivies to the list.
- You can also add elements manually, pushing the "**Add Elemen**t" button and dragging the ivies to the slots.
- The "**Del**" buttons delete the item from the list, not the ivies from the scene.



Generating ivies at runtime

By last, in order to make easy for you create ivies in runtime you have the "**Ivy Caster**" component.
Is a script you can attach to any gameobject in your scenes. **It will manage all the generation stuff for the ivies.**
The **interface is the same as the RT Ivy Controller**, and the data set in this interface will be transferred to the controllers of the ivies created via this instance of Ivy Caster.
Each Ivy Caster can create ivies of a **single preset**. If you want multiple kind of ivies, you have to create multiple Ivy Caster, each one with the preset you desire.

For make the Ivy Caster to create an ivy at runtime, you only have to call a method in a instance of Ivy Castar via script. The method is called “CastIvy” and you have to give it 3 arguments:

- **Position** (vector3). Is the position where you want to put an ivy.
- **ForwardVector** (vector3). Is the forward vector of the transform of the ivy. It should be pointing to the surface the ivy is attached to.
- **Offset** (float, 0.05 by default). This optional argument will be the distance the transform will be displaced from the position in the opposite of the ForwardVector direction. Is useful if you are placing ivies with raycast.

There’s a scene included in the plugin with this script, for “shoot” ivies.

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class IvyShooter : MonoBehaviour {
6
7      public IvyCaster ivyCaster;
8      public Transform cameraTransform;
9
10     // Update is called once per frame
11     void Update () {
12         if (Input.GetKeyDown(KeyCode.Mouse0)) {
13             Ray ray = new Ray (cameraTransform.position + cameraTransform.forward * 0.5f, cameraTransform.forward);
14             RaycastHit RC;
15             if (Physics.Raycast (ray, out RC)) {
16                 ivyCaster.CastIvy (RC.point, RC.normal);
17             }
18         }
19     }
20 }
```

Basically stores one Ivy Caster instance in a variable, and calls the method CastIvy (line 16) and as arguments adds the point and the normal of a raycast casted from one camera.