

Particles for UI Toolkit

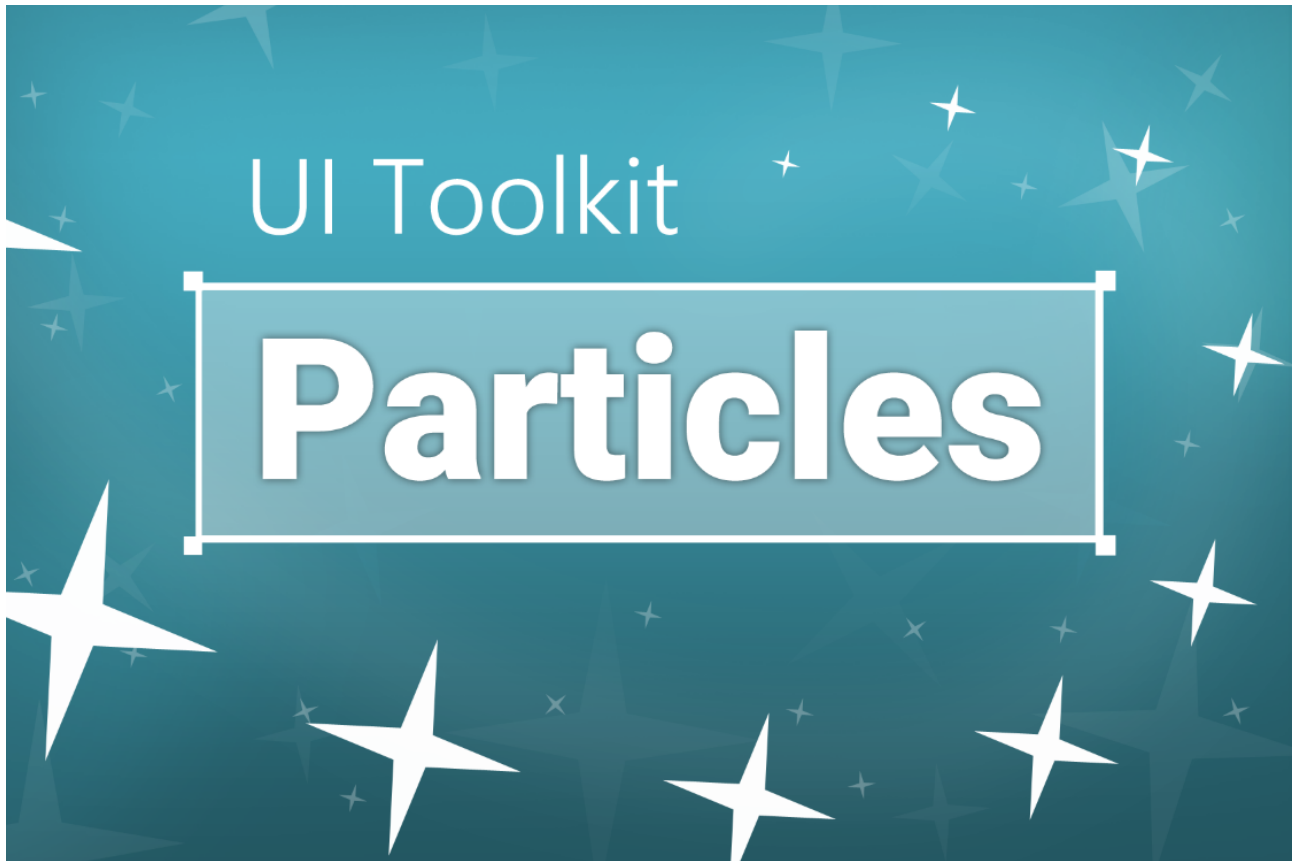


Table of contents

Requirements & Setup	3
Requirements.....	3
How does it work (Please read this FIRST)	4
Tutorial: How to create a particle image	6
Particle Image	9
Play On Show.....	9
Origin.....	10
Tint.....	12
Pixels per Unit.....	12
Attractor.....	12
Preview In Game View (Editor Only).....	13
Create Particle System.....	14
Particle System For Image	15
Guid.....	15
Texture.....	15
Origin Transform.....	15
Control Buttons.....	15
Particle System	16
Unsupported modules.....	16

Tips & Tricks	18
Origins and Attractors in simulation space „World“	18
Debugging the particle system.....	19
Frequently Asked Questions	20
What's the „There is no UI Document containing this ParticleImage in the Scene.“ warning?...	20
Why are some features not supported?.....	20
There is some memory garbage created in the first couple frames.....	20
The element origin seems off in the GameView window if simulation space „World“ is used...	21
The element origin does not move in the UI Builder if Origin „Transform“ is used.....	21
The particles seem off in the UI Builder if simulation space is „World“ and viewport scale is not 100%.....	21
The attractors seem off in the GameView window if simulation space „World“ is used.....	21
The particles move with the element. How do I make them stay?.....	22
The particles are visible outside my element. I want them to be cropped.....	22

Requirements & Setup

Requirements

Unity 2021.2 or higher is required since that is when Unity added the UI Toolkit Module for runtime use. If you can, please upgrade to the highest LTS version of Unity. The newer the version the less „glitches“ the UI Toolkit has.

Please keep in mind that UI Toolkit as a whole is still a work in progress and not quite ready for prime time. Unity itself still recommends using UGUI instead of UI Toolkit for runtime applications ([source](#)).

How does it work (Please read this FIRST)

This section will help you understand what is going on and why some things are a bit quirky.

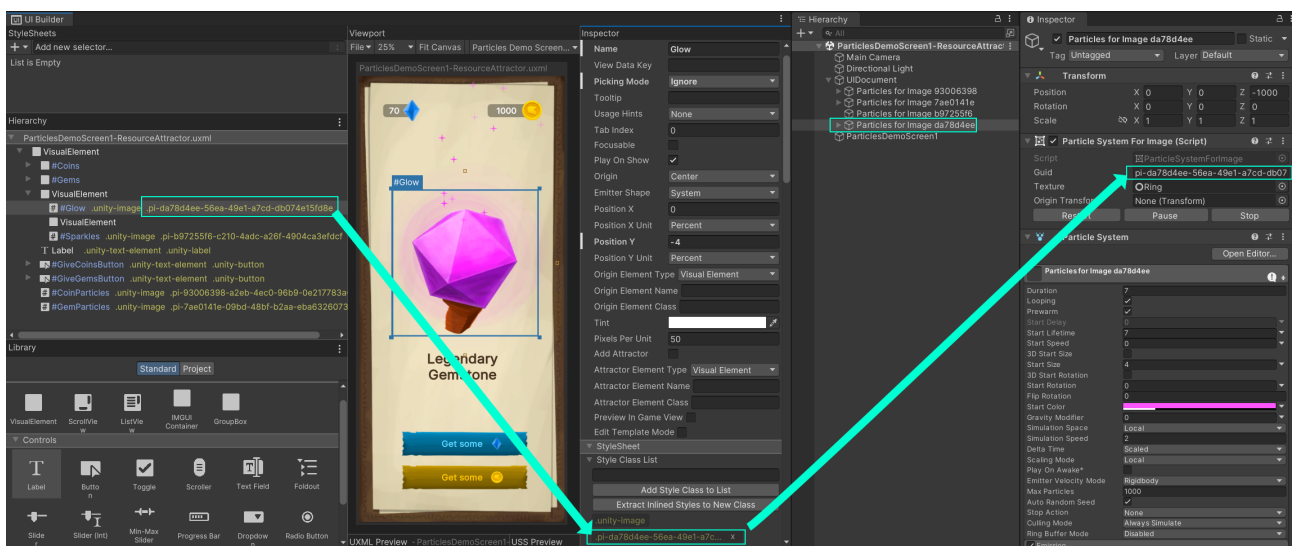
The particles are controlled by three objects:

- The ParticleImage custom control in the UI
- The ParticleSystemForImage component in the scene
- The ParticleSystem in the scene (tightly coupled with the ParticleSystemForImage)

The tool uses a normal Unity particle system and copies most (not all) of its properties into the UI Toolkit VisualTree. To keep up with the current state of the particle system it has to do this every single frame. Don't worry, it's highly optimized.

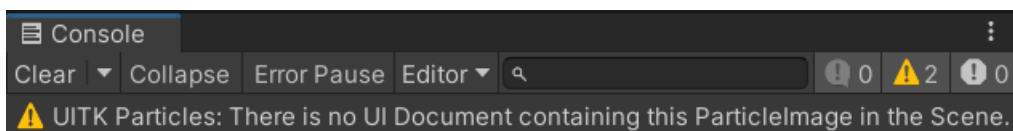
The key part is this: **The particle system is NOT part of the UI. It lives in the SCENE (see image below).** Without a scene no particle simulation will happen.

The reason is that the UI Builder is pretty limited by design. It's not possible to attach components to ui elements. This means we also can not add a particle system directly to a ui element. Therefore we have to create a game object in the current scene that will keep hold of the particle system instead of the ui element. It is created as a child of the UI Document.



This also means **if you load your UI into the UI Builder without a scene (containing a UIDocument with that UI attached) then you will not see any particles.**

If you try to edit a UI with a ParticleImage and you do not have the right scene opened then the tool will log a warning to let you know.



The tool checks for the particle system in the scene every time a ParticleImage is selected in the UI Builder.

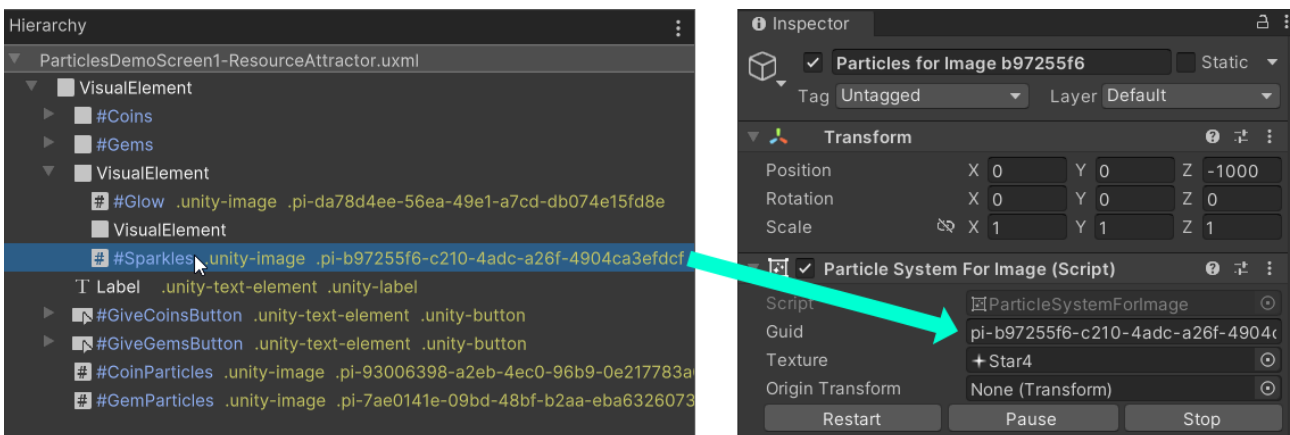
Hint: You can disable the warning in the settings.

ONLY if a UI Document containing the current ui is found will the tool create a particle system. Otherwise it will log the message and do nothing. This is done to avoid creating unneeded particle systems all over the place.

If you click a ParticleImage in the UI Builder the tool will automatically select the matching particle system in the scene.

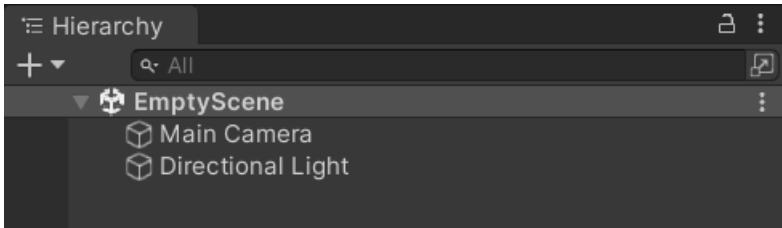
It may happen that you see the „No UI Document“ warning despite having the right UI Document in your scene. This may happen if you have not yet **SAVED** your ui. To fix that please save the ui asset. Then it should create the particle system in the scene.

The connection between the ui element and the particle system in the scene is maintained via a **GUID** stored as a USS class.

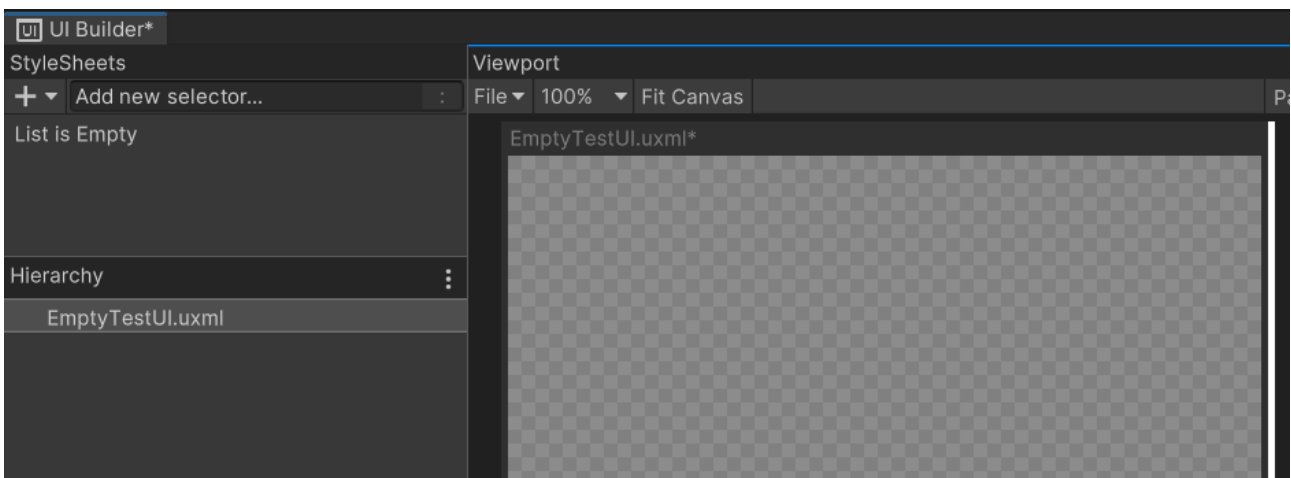


Tutorial: How to create a particle image

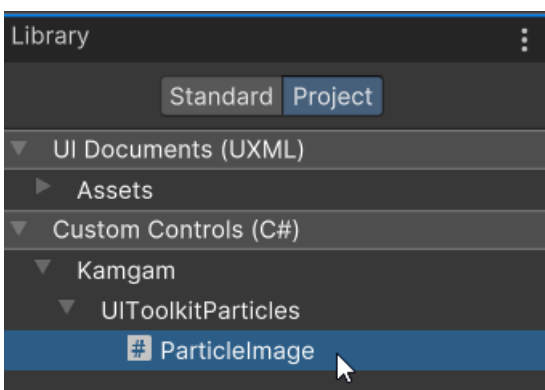
Let's assume you have an empty scene. Like this:



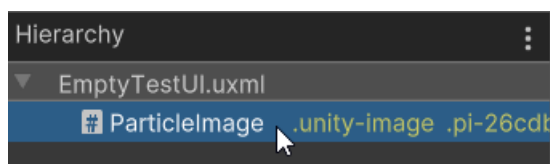
And you also have an empty UI opened in the UI Builder. Like this:



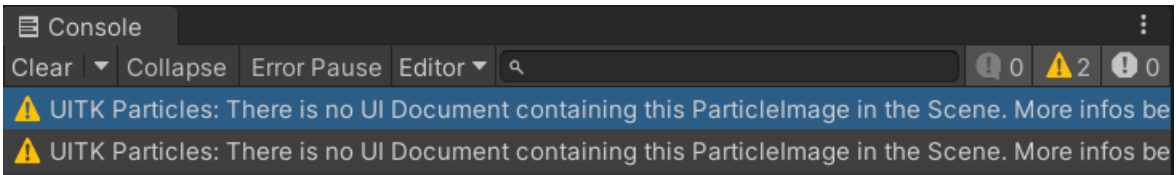
In the Library within the UI Builder switch to the „Project“ tab and unfold the „Custom Controls“ tree until you find the „ParticleImage“.



Drag it into the Hierarchy. It should look like this:



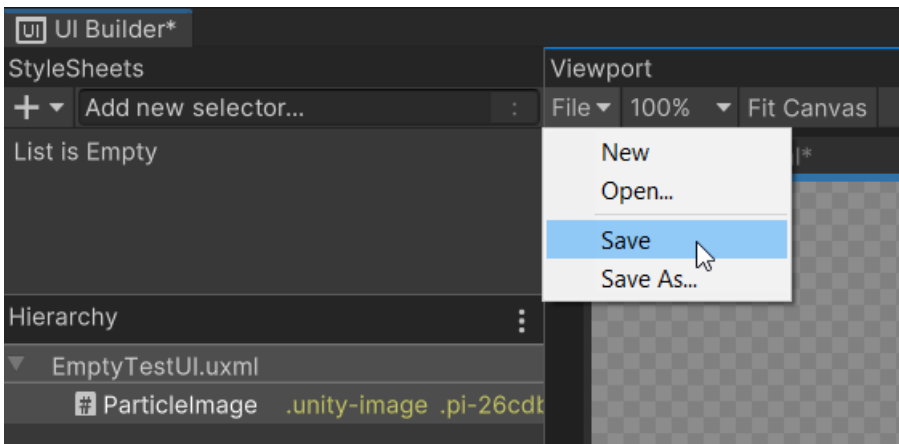
You may have noticed some warnings in the console.



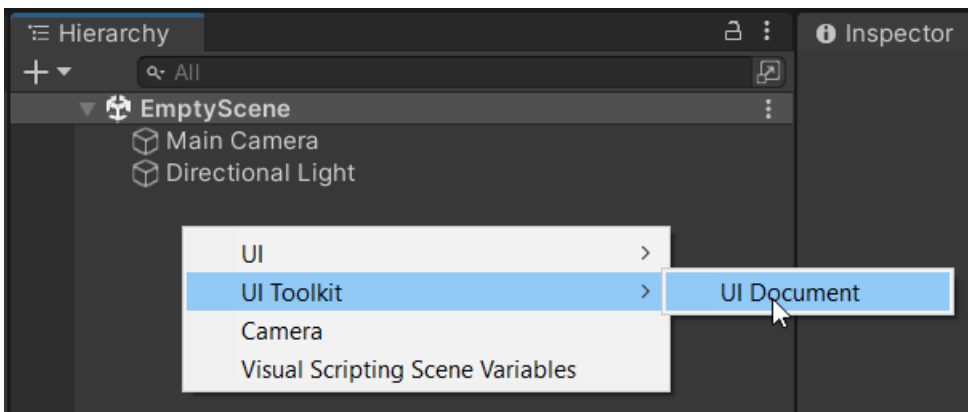
The warning is shown because we do not yet have a UI Document that uses our ui (more on that in the „How does it work ..“ section above). The ParticleImage does not know where in the scene to create the particle system.

To fix that please add a UI Document to your scene and make it use your current ui asset. Don't forget to save your ui first.

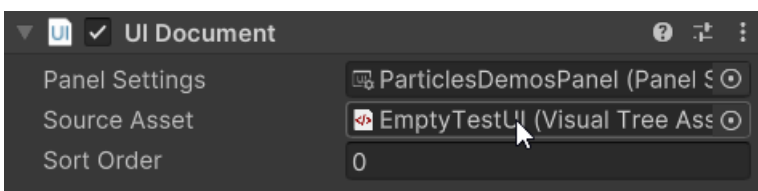
Save your ui:



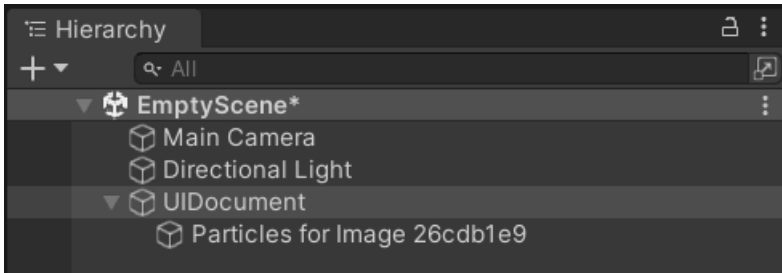
Create a UI Document:



Make sure the UI Document references the correct UI (the one you have saved just now):

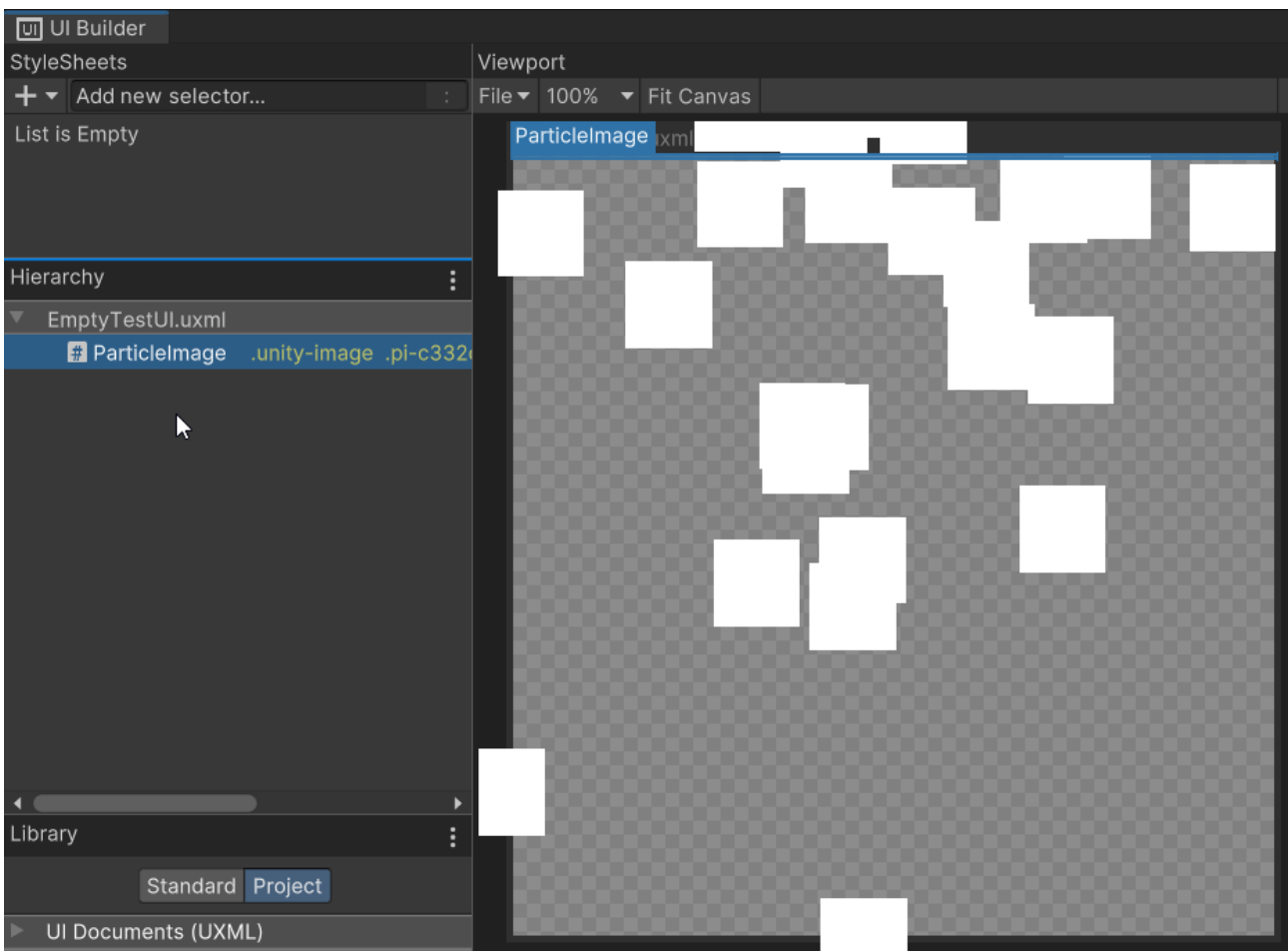


Once the UI Asset is referenced the ParticleImage should pick up on the change and create the particle system in the scene. Like this:



Don't forget to save your scene file too!

If you select the ParticleImage in the UIBuilder you should now see some white square particles streaming out. Congratulations, you have added your very first particle image.

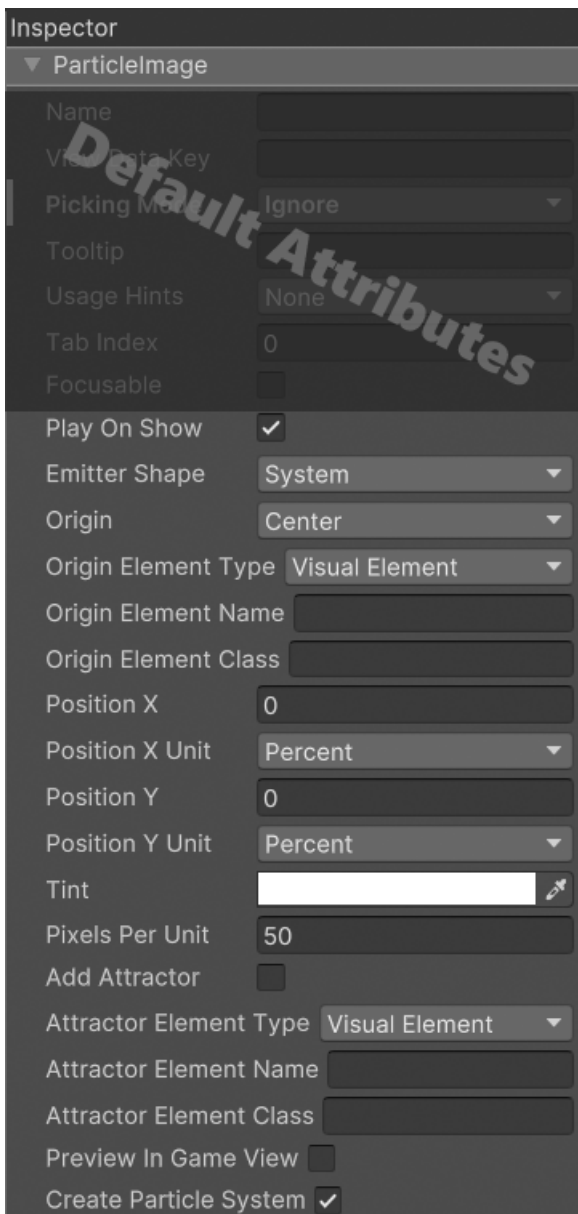


Of course this does not look good yet. To make those particles shine you will have to configure them. Go check out the sections below or the examples under **Assets/Kamgam/UIToolkitParticles/Examples**.

Particle Image

The particles are controlled by three objects:

- The ParticleImage custom control in the UI
- The ParticleSystemForImage component in the scene
- The ParticleSystem in the scene (tightly coupled with the ParticleSystemForImage)



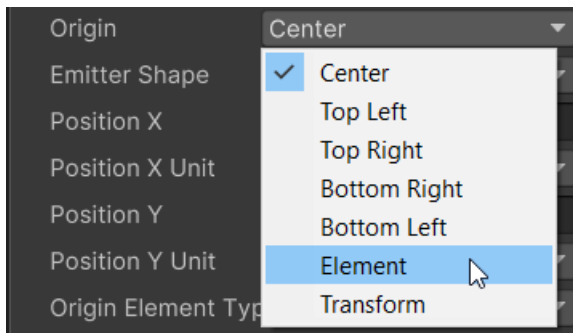
Play On Show

This one is similar to the „Play On Awake“ you know from the normal particle system. Except it reacts to show events instead of awake. The logic determining when a show even occurs takes multiple things into account: visibility, display, enabled state in the hierarchy and opacity.

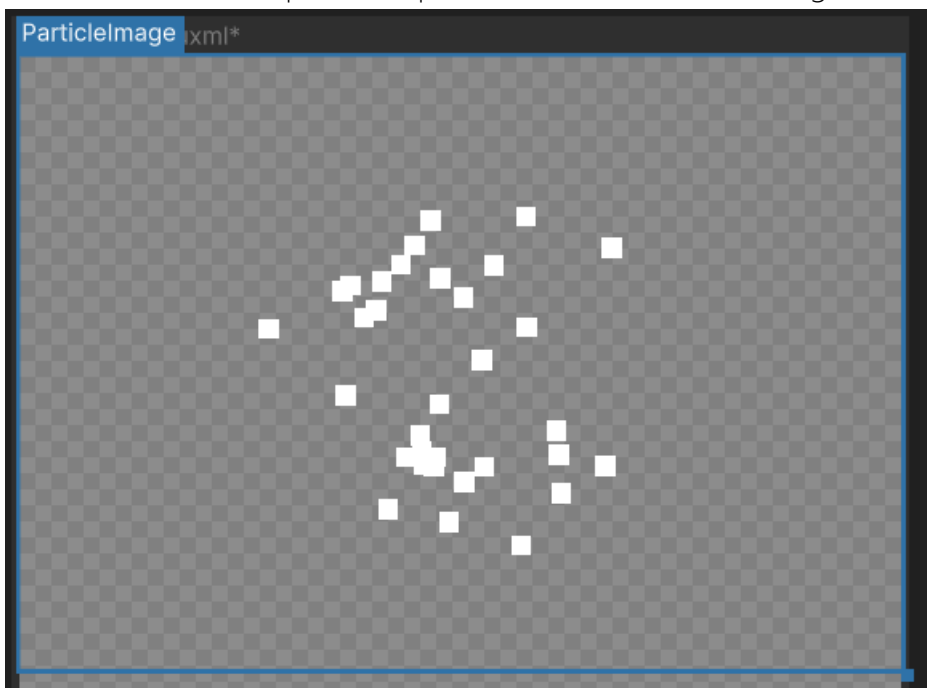
If enabled then the particles will start to play every time the ui element is shown.

Origin

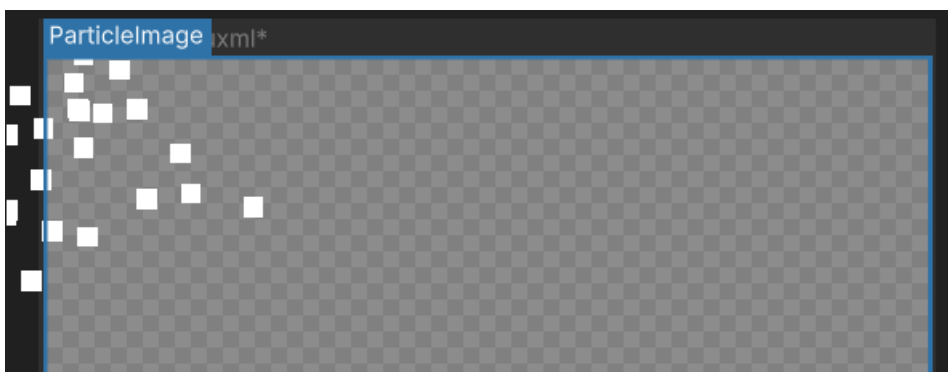
The origin refers to where the particles will spawn in relation to the ParticleImage local space.



Center will make the particles spawn in the center of the image.

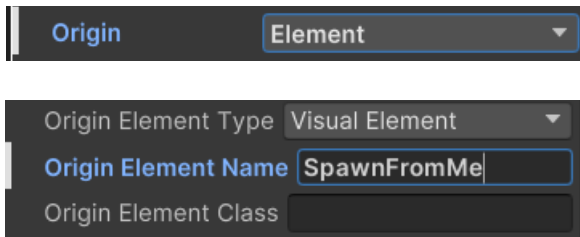


TopLeft will make them spawn from the top left corner



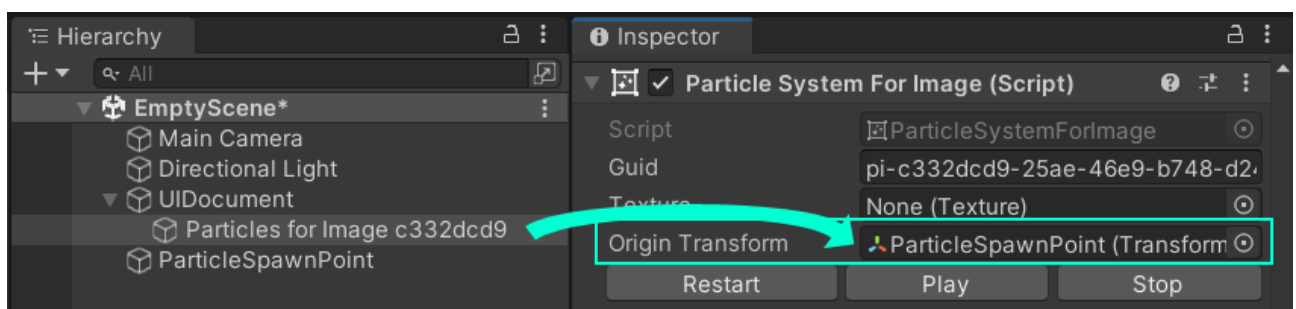
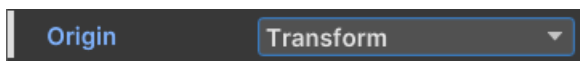
There are two special settings: **Element** and **Transform**.

If you choose **Element** then the origin of the particles will be based on the position (center) of another ui element. To find that element a query is made based on the „Origin Element“ attributes.



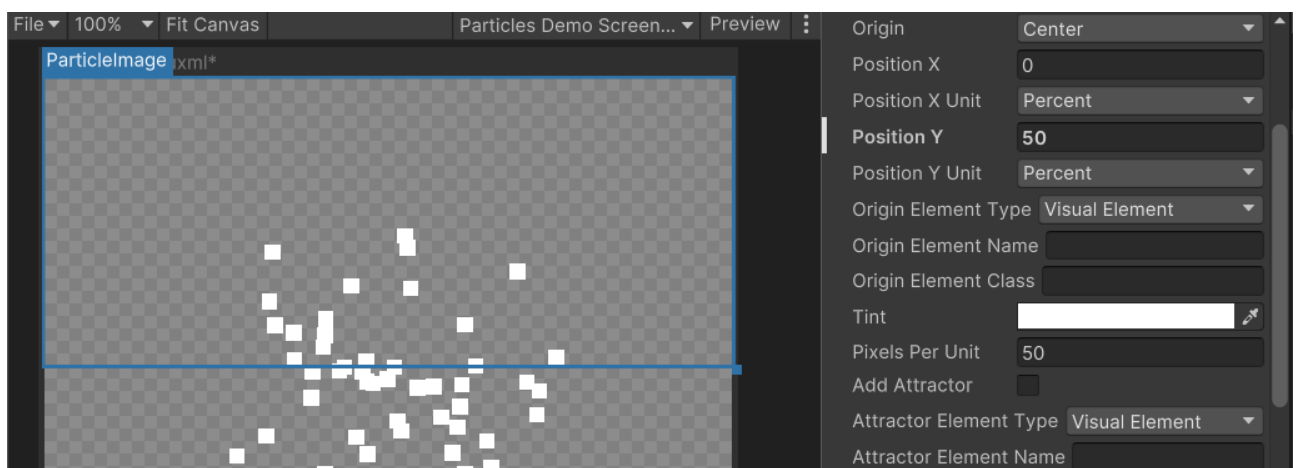
If you leave the name and/or the class empty then it will be ignored. The particle origin will be the first ui element that matches the criteria.

If you choose **Transform** then the origin of the particles will be based on the „Origin Transform“ referenced in the scene (the Particle System For Image).



The „Position“ values are relative to the origin. This means that these values are added to the origin.

For example: If you want your particles to spawn at the bottom center then choose „Center“ as the origin and add 50% to the position.



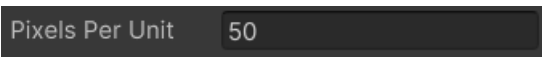
Tint

You can use this to change the color tint of your particles, even after you have set a color in the particle system. It supports opacity too.



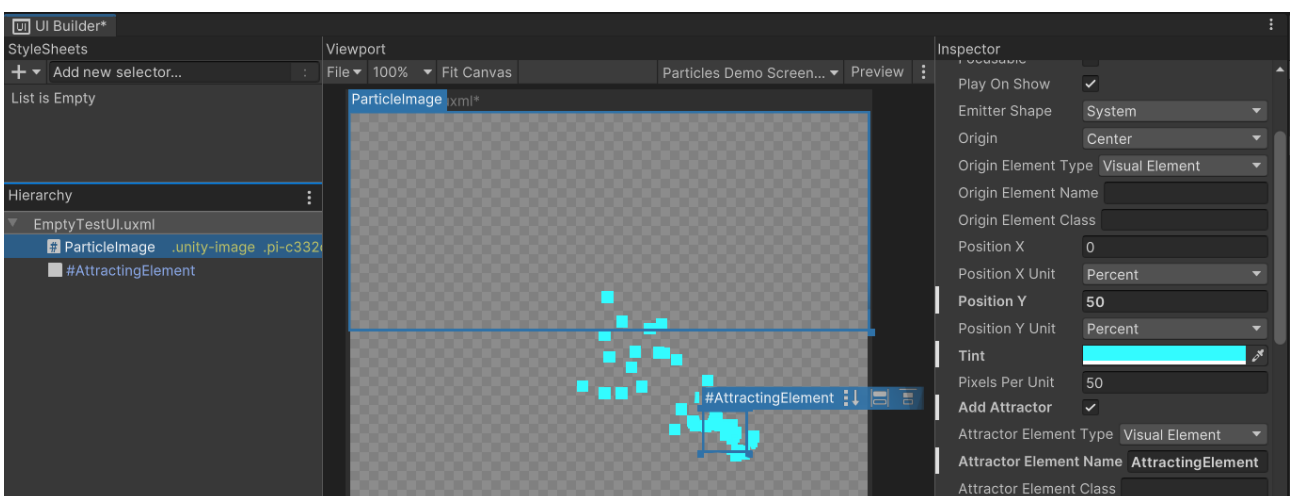
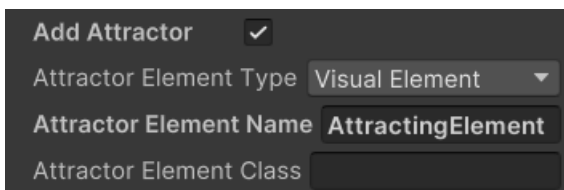
Pixels per Unit

Since the actual particle system runs in the scene and not the UI we need to convert the size from the 3D world space to the ui space. Basically it scales them up. By default one 3D world unit will become 50 pixels.

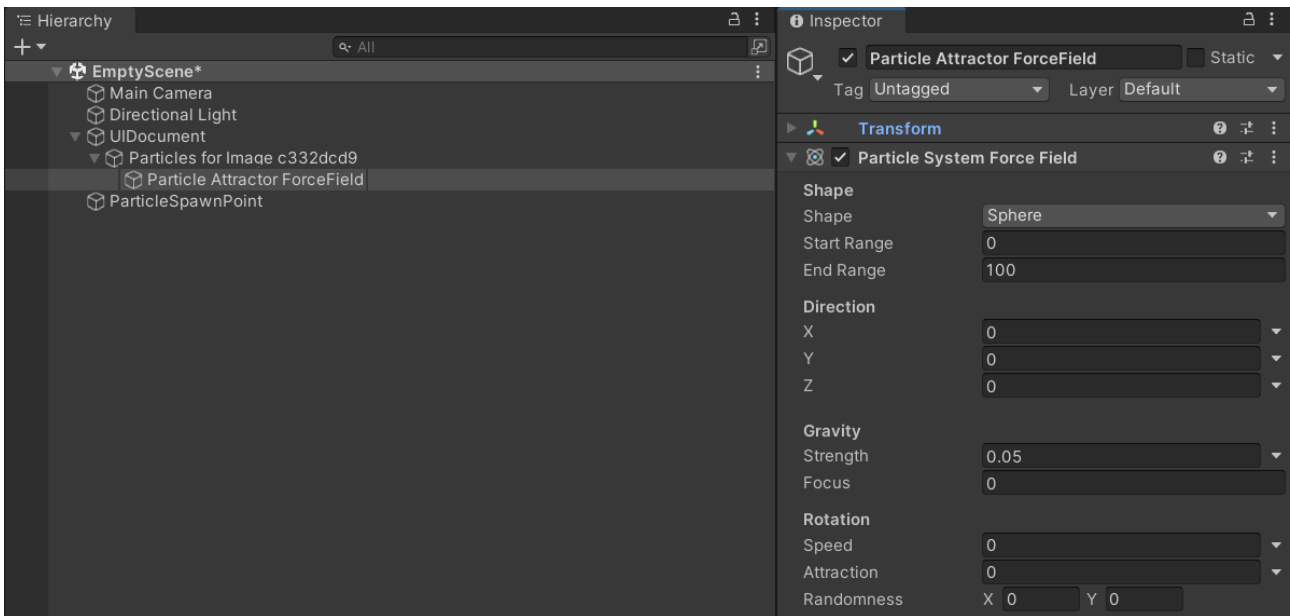


Attractor

You can add a particle force field (called attractor) by enabling this checkbox. If you enable it you need to specify the „Attractor“ query attributes. The particles will be attracted to the first element matching the criteria.



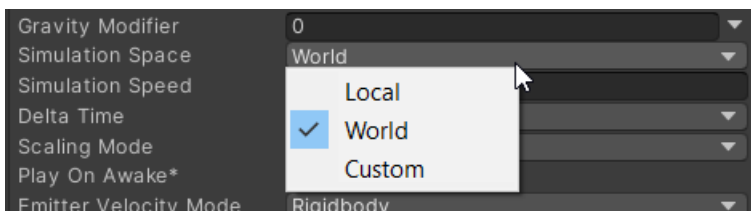
To configure the details of the attractor you will have to select the Particle Force Field in the scene. It is added automatically to the particle system.



Check out the Attractor examples under **Assets/Kamgam/UIToolkitParticles/Examples**.

Preview In Game View (Editor Only)

When using the Origin setting "Element" or "Transform" in combination with simulation space "World" you may have noticed that the particles in the Game View do not align with the UI.



This is because by default the particles do align with the UI Builder window to make layouting easier.

However, for development purposes it is sometimes handy to have the particles show up at the right place in the Game View. To do that use the **Preview In Game View** attribute.



This ensures the particles are shown at the correct location in the Game View. If you enable this they will NOT be at the correct location in the UI Builder.

In play mode and in builds this option is ignored and particles are always shown in alignment with the game view.

Create Particle System

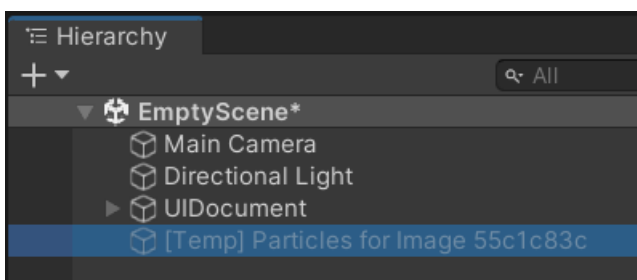
You uncheck this to disable the automatic particle system creation in the scene.

This is useful if your ParticleImage is part of a template that is to be added dynamically. In that case you may not want to create one particle system for each image.

Create Particle System

For templates you should still add a guid class (.pi-...). Once the image is added it will always try to find a matching particle system in the scene.

NOTICE: The system actually will add a particle system, even if this is unchecked. Yet that system will not be editable and it will not be saved. It's just there to help you visualize the other attributes on the ParticleImage ui element (things like the origin).

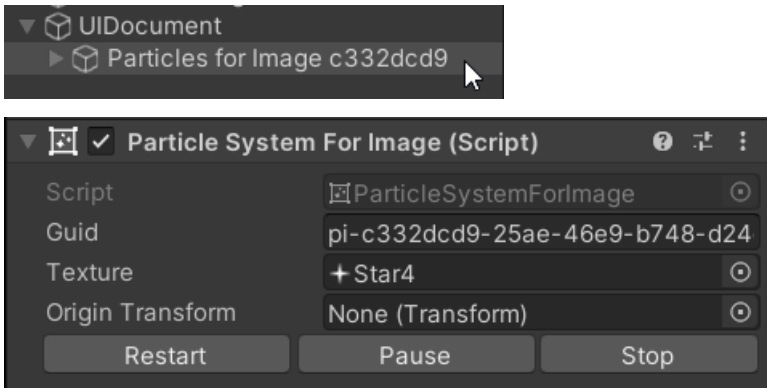


Check out the „MultipleInstances“ demo and the „Template“ under **Assets/Kamgam/UIToolkitParticles/Examples** for more details.

Particle System For Image

The particles are controlled by three objects:

- The ParticleImage custom control in the UI
- The [ParticleSystemForImage](#) component in the scene
- The ParticleSystem in the scene (tightly coupled with the ParticleSystemForImage)



Guid

The unique class name that links this particle system to the particle image in the ui. Particle image guides have to start with „pi-“ and need to be unique.

Texture

The texture used for each particle. You can only use one texture per particle system. That's a limitation by the UI Toolkit. Hopefully they will add more textures (and materials) some day.

Origin Transform

This is the origin source if the „Origin“ attribute of the particle image is set to „Transform“. For more details please refer to the section on the „Origin“ attribute above.

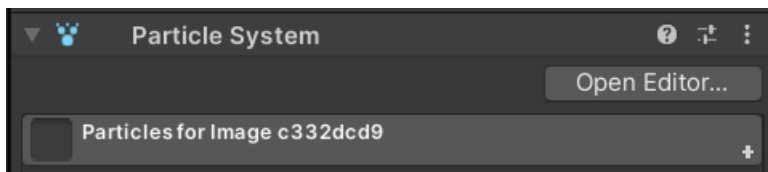
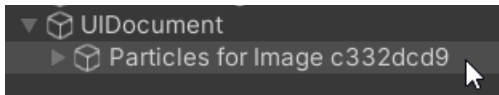
Control Buttons

Use these buttons to control the particle play state in the editor.

Particle System

The particles are controlled by three objects:

- The ParticleImage custom control in the UI
- The ParticleSystemForImage component in the scene
- The ParticleSystem in the scene (tightly coupled with the ParticleSystemForImage)



Unsupported modules

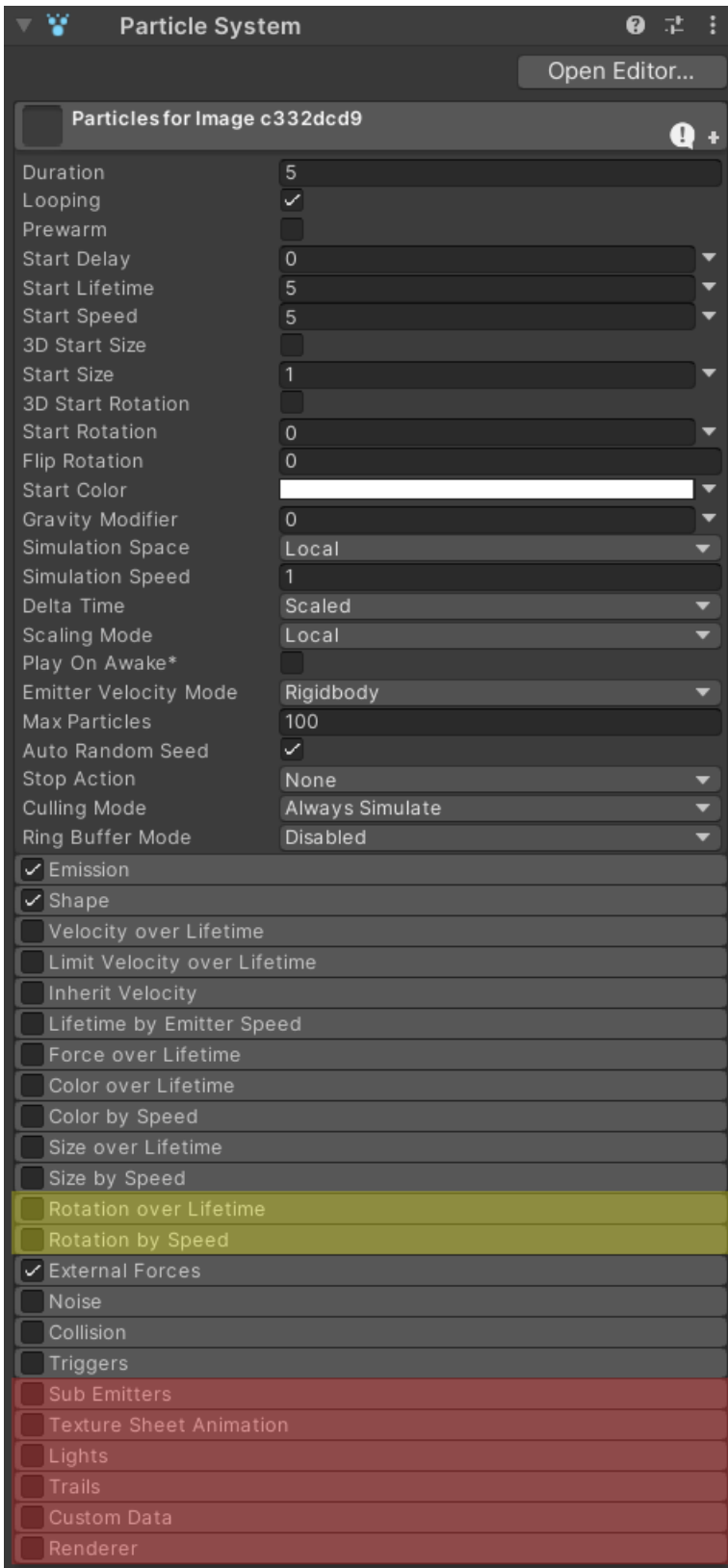
The particle system is the Unity default component. However, not all its features are supported.

The most notable limitations are:

- **Z-axis rotation only:** Only rotation around the z-axis is supported. The reason is that UI Toolkit will automatically clip anything that is rotated on the other axis.
- **Texture Sheet Animation:** Texture sheet animations would require multiple texture support in UI Toolkit. Sadly that is not possible at the moment. Hopefully Unity will add this feature one day. If you need multiple textures simply use multiple particle images.
- **Custom Materials:** Since we can not set the material in UI Toolkit rendering we can not assign any shaders. The UI Toolkit default shader is unlit. Hopefully Unity will add this feature one day.
- **Sub Emitters:** These are not supported at the moment.
- **Trails:** These are not supported at the moment as they would require a second material.
- **Custom Data:** The ParticleSystemForImage does not have an API to access custom particle data. Yet, if needed you can code this yourself. The particle system is fully accessible via ParticleImage.ParticleSystem.
- **Renderer:** The particle renderer is not used at all in UI Toolkit and thus none of it is supported

To give an overview the modules that are definitely not working are marked red in the image below. Those working partially are marked yellow.

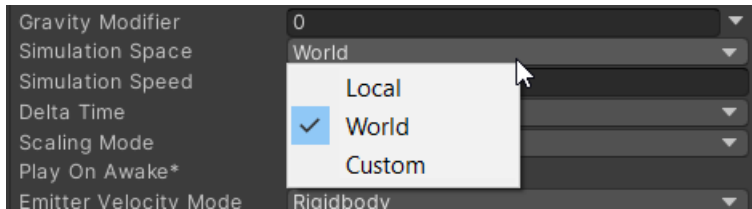
That does not mean all the other modules and all their options do work. For example the „Simulation Space“ setting supports „Local“ and „World“ but not „Custom“. The best way to find out what's working is to just try it out.



Tips & Tricks

Origins and Attractors in simulation space “World”

When using the Origin setting “Element” or “Transform” in combination with simulation space “World” you may have noticed that the particles in the Game View do not align with the UI.



This is because by default the particles do align with the UI Builder window to make designing them easier for you.

However, for development purposes it is sometimes handy to have the particles show up at the right place in the Game View.

To do that use the **Preview In Game View** attribute.



This ensures the particles are shown at the correct location in the Game View. If you enable this then they will NOT be at the correct location in the UI Builder (it's either GameView or UI Builder, not both)

In play mode or in builds this option is ignored and particles are always shown in alignment with the game view.

Debugging the particle system

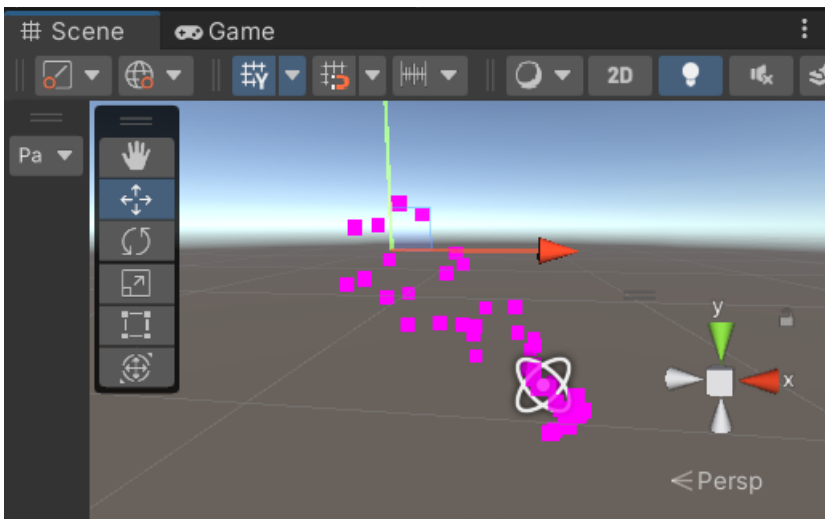
It sometimes helps to see what the particle system is doing. Most of the emitter shapes for example are 3D, not 2D. Observing what they actually do can help understand the behaviour of your particles.

By default the „renderer“ module of the particle systems is disabled. Yet for debugging you can turn it on.



If you then double click your particle system in the scene it should take you there.

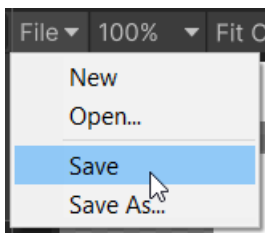
All the particles will be pink because not material is assigned, but you can still see their movement. Especially in combination with forcefields this is a handy trick.



Frequently Asked Questions

What's the „There is no UI Document containing this ParticleImage in the Scene.“ warning?

This means you have no scene opened that contains the ui you are currently editing. You should open the right scene or create a UI Document referencing the ui asset. It may also happen if you have not SAVED your ui. To fix that please save the ui asset and the scene. Then it should create the particle system in the scene.



Hint: You can disable the warning in the settings (Tools > UI Toolkit Particles > Settings).

Please refer to the „How does it work?“ section for more details.

Why are some features not supported?

Sadly what we can render in the UI Toolkit is limited. For example we can not use any custom materials or multiple textures. That's a limitation set by Unity in combination with the approach we use for this tool. Hopefully Unity will add some more features to the UI Toolkit renderer in the future. Once they are there we can patch this tool to support them.

If you really need all the particle system features then one solution is to render the system via a separate camera on a separate layer into a render texture and then use that texture as the background image on your visual element. Though, this is not something this tool currently supports.

There is some memory garbage created in the first couple frames.

This is due to the Unity internal allocation of the geometry for the particles. Sadly this can not be avoided, but after about 5 frames the internal pools are warmed up and then every new frame has zero GC allocations.

Increasing the max particles property of the system will also cause some new allocations as there is more memory required if the number of max particles increases.

The element origin seems off in the GameView window if simulation space „World“ is used.

Yes, this is by design. This happens because the target element can be on different positions inside the UI Builder and the Game View. Therefore we had to decide which one takes precedence and that's the UI Builder since that is where you will spend most of the time designing the ui.

You can force the GameView as the preferred target via the „Preview In Game View“ attribute. Please notice that this attribute is for development purposes only. In build and during play mode this will always be assumed as “on”.

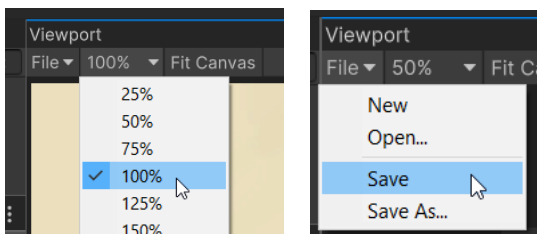


The element origin does not move in the UI Builder if Origin „Transform“ is used.

Yes, this is by design. To match a 3D object there has to be a camera and a 3D world. Both are missing in the UI Builder. Look in the Game View. In there the particles will match the 3D object.

The particles seem off in the UI Builder if simulation space is „World“ and viewport scale is not 100%.

Yes, sadly there is not workaround yet except setting the viewport too 100% or saving the UI layout:

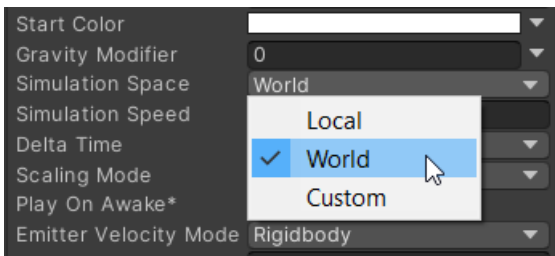


The attractors seem off in the GameView window if simulation space „World“ is used.

Yes, this is by design. Please refer to the “The element origin seems off in the GameView window” question above for more details. - You can force the GameView as the preferred target via the „Preview In Game View“ attribute.

The particles move with the element. How do I make them stay?

Change the simulation space on the ParticleSystem to "World" instead of "Local".



The particles are visible outside my element. I want them to be cropped.

Change the Overflow style to "hidden". This will crop the particles just like any other content.

