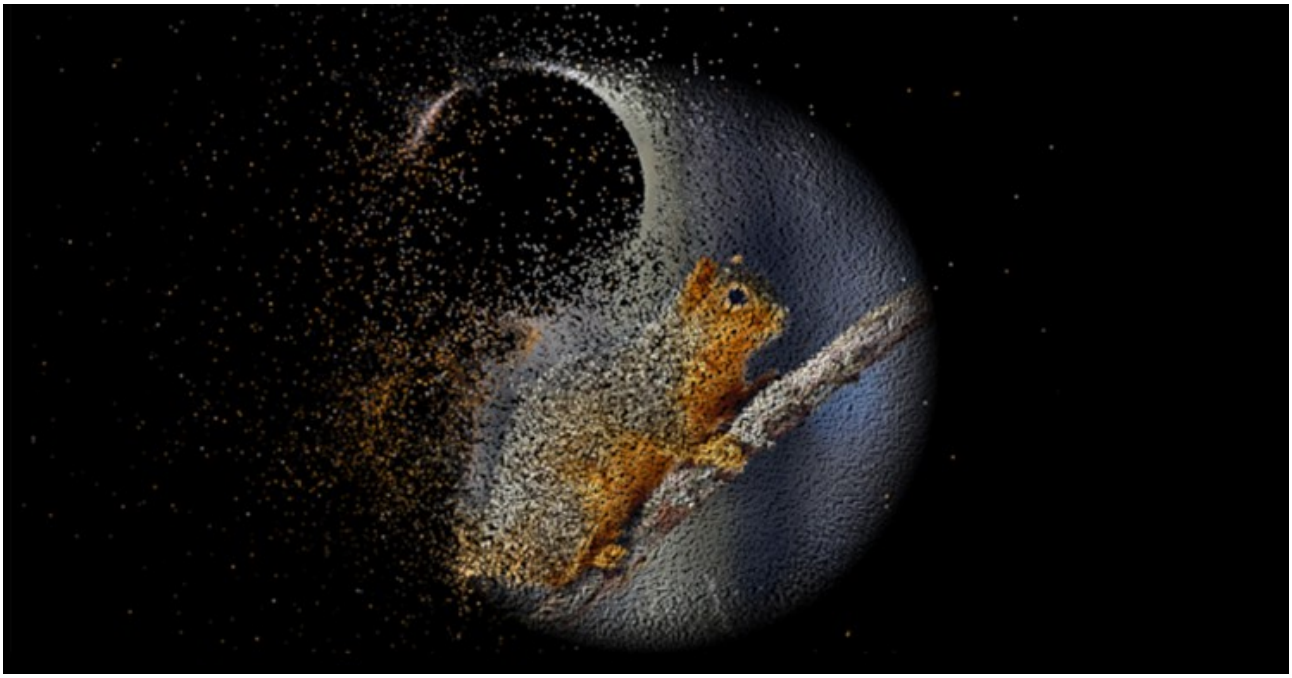
PINGPONG
pingpongtech.com

PPT Particles
v 2.0



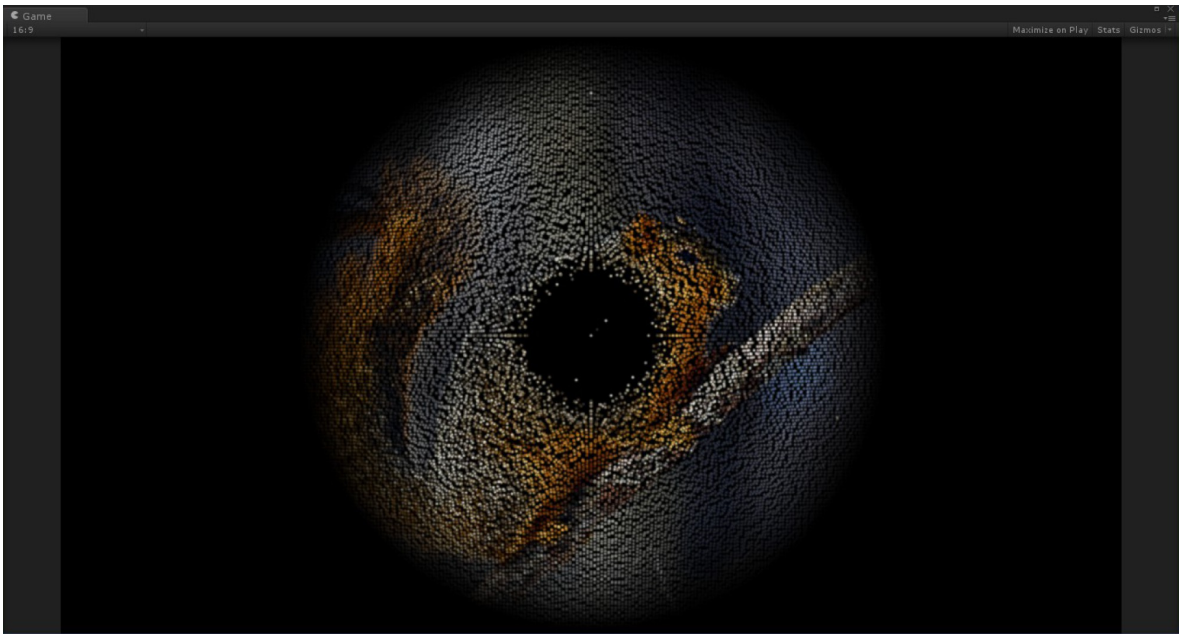
PPT Particle System v 2.0

Table of content

Introduction

Welcome to the PPT Particle System 2,0.
This plugin has been developed in order to convert any kind of image or Render Texture into a particle system that can be affected by external forces. The images that can be converted into a particle system are plain .jpg or .png or any other image file, and they can include an alpha channel. The render textures are rendered from any camera in the scene and therefore allow videos, objects, scenes, etc to be displayed by the particle system.

These external forces are generated by a script attached to an empty GameObject, and can interact with the particles and create motion and effects on them. That means that it is possible to play with the forces, enabling or disabling the GameObject, moving it or increasing or decreasing the intensity of rejection or atraction of the force.
The particle system is created on the start event of the game object and cannot be modified on real time.



Quick Start

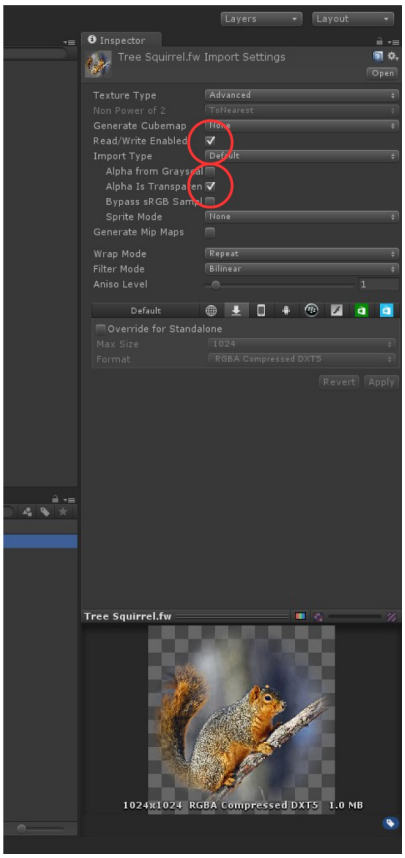
Converting Images

The first step is to import the image that has to be converted into particles.

Importing the image file

Any type of image supported by Unity3D can be used, regardless of its size or wheather they include an alpha channel or not. The package comes with a sample image in PNG format with an alpha channel and a transparent gradient effect:

The system handles images with transparencies drawing a particle wherever it finds a pixel with alpha channel > 0.
However, that can be modified using the Alpha Cut Off property in the PPTGenerator script. More details on how to do this will be given in following sections.



Import settings:

The “Advanced” option must be selected in the “Texture Type” dropdown menu.

For the image to work with the particle system, the “Read/Write” property must be enabled.

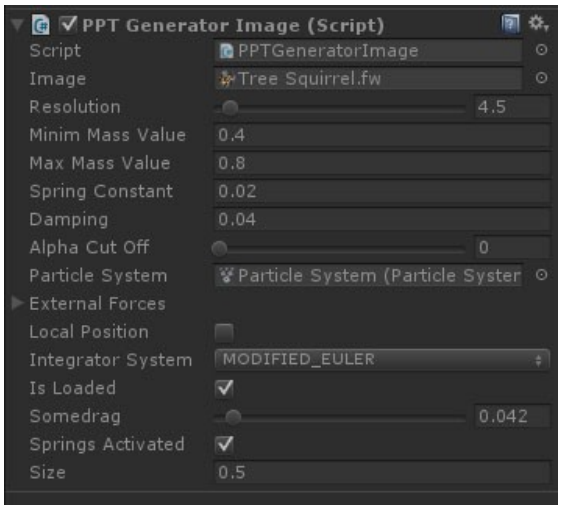
If the image has parts with transparency, the “Alpha Is Transparency” option needs to be checked.

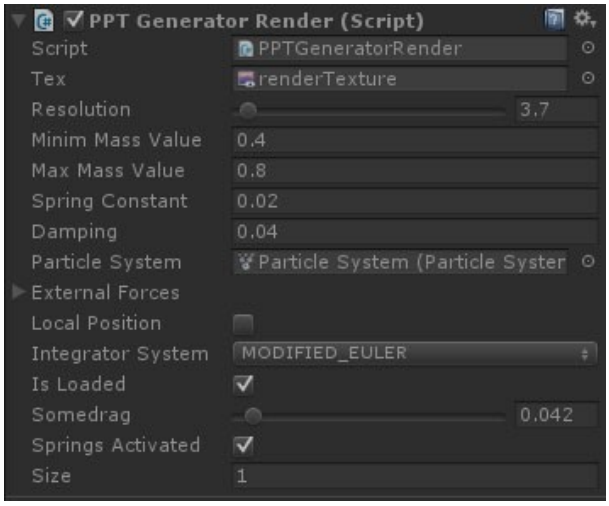
Converting Render Textures

The first step is to set up the content to be displayed inside the Camera To Render GameObject hierarchy. The content can be any material supported by unity: 3D gameobjects affected by lights, quads with movietextures, even other particle systems, etc. The package comes with a sample scene including a quad with a sample video as movieTexture. The main objective of this asset is to allow videos to be displayed as particle systems. However, since Android does not support Movie Textures, it will not be possible to use the asset for this purpose in Android unless you use another system to display videos instead of Movie Textures.

When playing the scene, a new layer named “PPT - ParticleSystem Render Texure” will be created on the first free slot of the layer list. To do so, there must be at least one free slot available. Every GO inside the Camera To Render hierarchy will have its layer set to “PPT - ParticleSystem Render Texure”. The Camera To Render GO will automatically set its culling mask to only rendering gameObjects using this layer. The main camera will automatically set its culling mask to not rendering this layer. This way, the GameObjects will not interfere with the rest of the scene.

PPTGenerator: script properties





There are two different scripts, one used to convert images and the other to convert render textures. There are slight differences between them. These scripts is responsible for the creation of the particle system.

Image

Only in the PPT Generator Image script. The image that we want to use in the system and we have already imported must be attached here.

Render Texture

Only in the PPT Generator Render script. The render texture we are going to display. It is the target texture of the Camera To Render.

Resolution

This field refers to the final resolution of the particle system. That is, the density of particles that form the image. With a low value the resolution increases, with a high value, it decreases.
This is due to the fact that the value represents the leap between the pixels that will be read and converted into particles. A high value means that more pixels of the image are being skipped.

Min and Max mass value

These parameters determine the random range between two values (minimum and maximum) for the mass of each particle.
This entails that every particle will behave in a slightly different way from the others when affected by forces. If the values are equal, all particles will have the same behaviour.

Spring Constant & Damping

These two values modify the different ways in which all particles behave in terms of spring and damping. They are used for the creation of the particle system and can not be modified on real time.
These four parameters just mentioned affect to the physic reaction of the particle system to forces. To achieve an specific behaviour, it is recomended to experiment with different values in all of them to see how the reaction varies until the desired result is obtained.

Alpha Cut Off

Only in the PPT Generator Image Script. This property determines which pixels of the image should be converted into particles or not, depending on the alpha value of every pixel. Given that the alpha channel of each pixel can vary range between 0 and 1 (being 0 total transparency and 1 no transparency at all), a value of 0 means that all pixels with an alpha value greater than 0 will draw a particle. The particle will have the same opacity than its corresponding pixel. A value of 1 means that only the pixels with alpha 1 will draw a particle and particles will have no transparency at all.

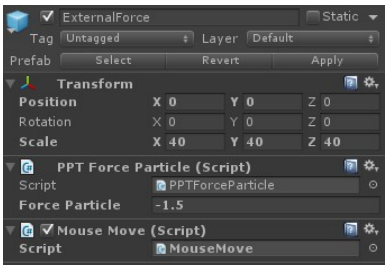
Particle System

Since the PPT Particle System 2D uses the Unity3D Particle System component for rendering, it is necessary to create a new GameObject > Partycle System and attach it to the script.

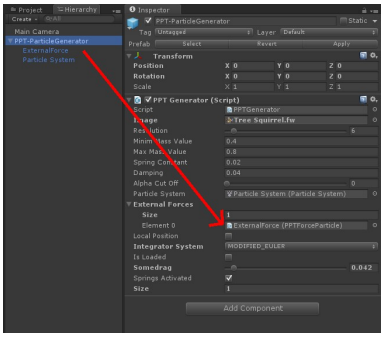
The default configuration of the Particle System Renderer can be used, just changing the “Start Size” value to change the size of the particles.

External forces

This is an array of GameObjects with the script PPTForceParticle attached. This script turns a new empty GameObject into a force able to interact with the PPT Particle System.
As many empty GameObjects as forces desired must be created, with the script attached to them.



The Force Particle property determines how powerful the force will be. If the value is positive, the particles will be attracted by the force. If it is negative, the force will reject them.



These gameObjects must be added to the array “External Forces” in PPTGenerator, dragging them from the scene hierarchy to the field in the script. By default there is one force attached and it is controlled by the mouse.

Local Position

A bool that determines whether the particle system uses the world or the local position of the forces GameObjects.

Integrator System

This drop down list allows you to choose the basis of the PPT Particle System 2D. That is, the system used to create the relationship between the particles and the forces. For the time being two kind of integrators are available. It is recommended to use “MODIFIED_EULER”. In upcoming releases other integrators might be added.

Is Loaded

This is a boolean flag that can be used to check if the field particle system is loaded.

SomeDrag

This value, between 0 and 1, affects the viscosity effect of the particles, that is, the speed they will have when returning to its initial position when affected by forces.

Spring Activated

It is possible to deactivate the spring property of the particles. This will make the particles never come back to their initial position when affected by a force. Instead, they will spread infinitely moving away from the initial position.

Size

The size of the whole particle system. 1 by default.

This is all you need to do to convert an image to a particle system with a customizable physic behaviour using the PPT Particle System 2D.

About Us

Ping Pong Technologies S.L. is a company from Barcelona (Spain), which develops interactive applications for fixed or road show installations. For support, more information, suggestion or other issues, please contact info@pingpongtech.com.