

Intro to Particle Systems Lab

LIGHTS, CAMERA, SPECIAL EFFECTS

Introduction

Ah particle systems. What a great invention. This lab will be a brief introduction to Unity's uber powerful particle system component. Although Unity's particle system is great because of all that it can do, beginners will find it a bit scary. When you create a new particle system, you get a game object with this crazy Particle System component that has an endless number of things to it (image on the right).

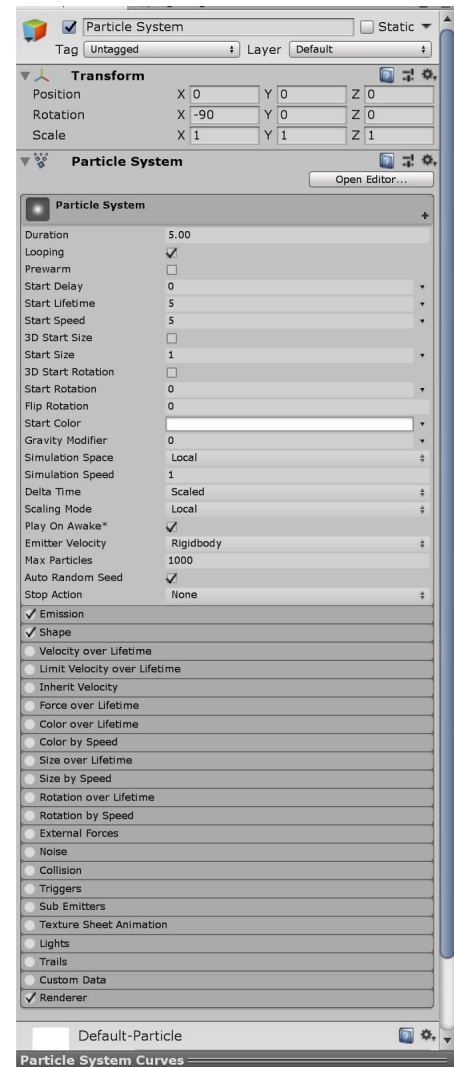
The goal for this lab is to introduce you to some of the things that you see as well as give you some intuition as to how you can go about creating a particle system.

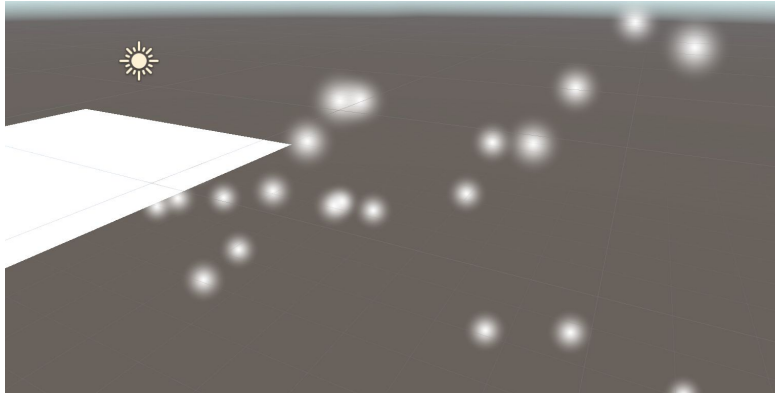
Tutorial

Before we start creating a particle system, we should have some idea as to what we want to make. In this case, we want to make lightning and so with a goal in mind, we can actually start creating it!

One way to create a new particle system is right clicking in the **Hierarchy** view, looking under **Effects** and then selecting **Particle System**. Go ahead and create a particle system in this manner. Now, go ahead and reset its **transform** so that it has a **position** of (0, 0, 0), **rotation** of (0, 0, 0) and **scale** of (1, 1, 1). And feel free to rename the game object to Lightning (as I will do so).

At this point, you should have something that looks similar to this in your editor:





Our goal looks is something more similar to this:

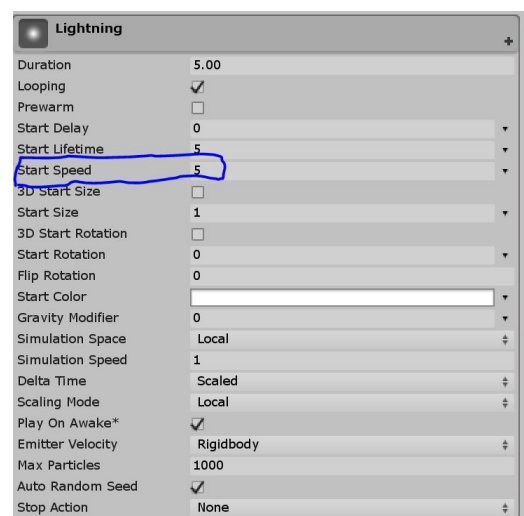


We have a long way to go...

Where do we even start? Well, in general, this depends on the person. For this tutorial, we will actually start by setting a speed for them.

Before we start though, I just have to mention one more thing. **Randomness** is your best friend when working with particle systems. You will see as we work through this that I use randomness nearly everywhere. Randomness makes particle systems feel a lot more realistic in general and I've found that when you don't use it in certain places, it just looks a little bit weird. I suggest you trust me on this one but feel free to try things out.

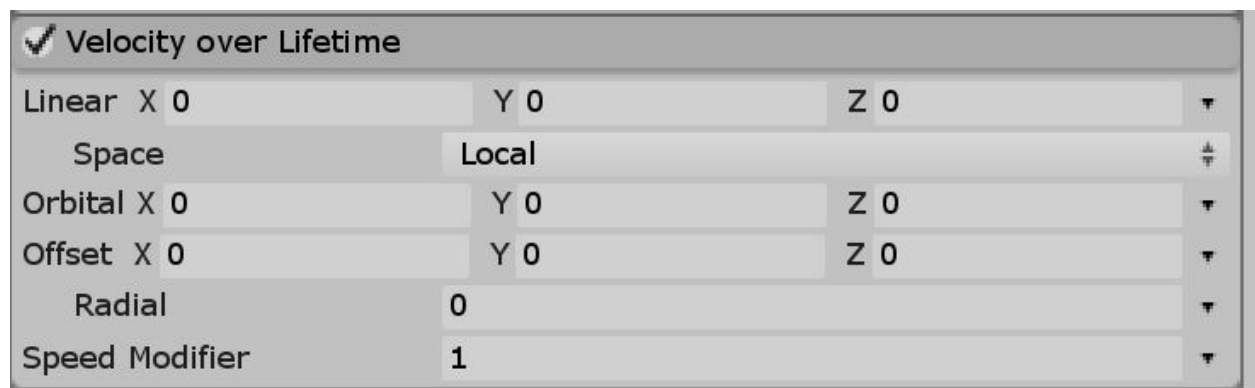
So back to setting the speed. Notice that right now we have a field called "**Start Speed**" set to 5. Let's go ahead and set it to 0 because we will actually use a special **module** to control the speed. What's a module you ask? What a great question! A **module** is one of the these things attached to the particle system component. As of Unity 2018.2, there are 23 modules. Some of the modules are **Emission**, **Shape**, **Velocity**



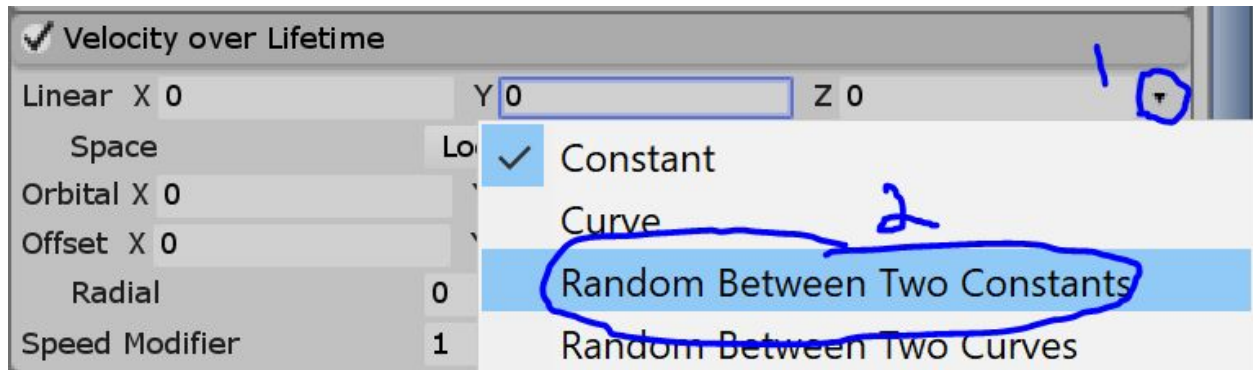


over Lifetime, Color over Lifetime, Noise, Sub Emitters, and Renderer. If you actually count the number of modules in the picture here, you'll notice that there are only 22. The 23rd module is actually nicknamed the **Main** module because it always takes on the name of the game object so in our case, it's called **Lightning**.

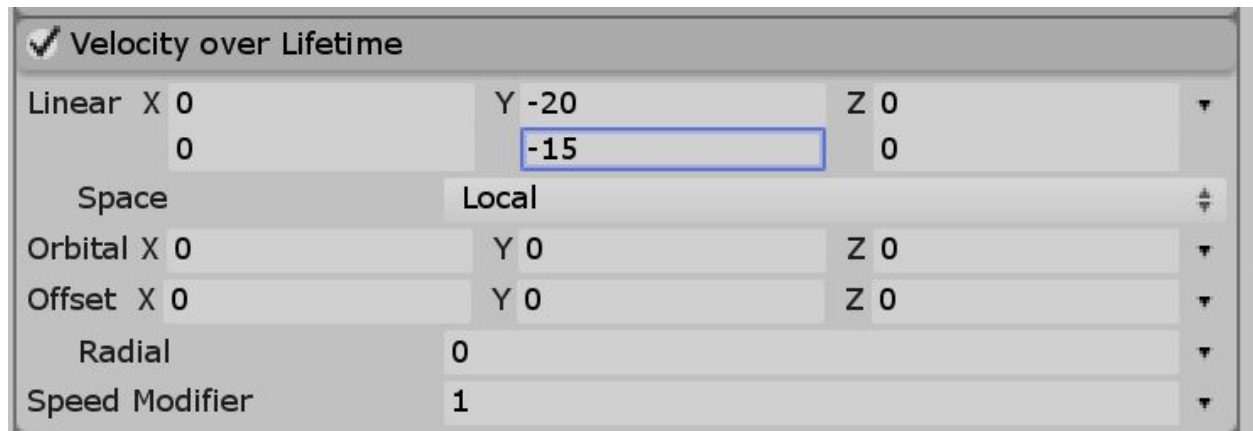
Start Speed is a field in the **Main** module. To open and close a module, just left click on it. Go ahead and open the **Velocity over Lifetime** module. Everything should be greyed out at the moment and that's because we have not activated this module yet. In order to activate it, we press the little circle to the left of the module name and once you do so, the module will activate and its fields accessible. Hopefully you now have something like this:



This might be a little bit intimidating but don't worry. The only field you should care about is the **Linear** field. Remember randomness? Well, let's begin using our new friend. Click on the down arrow at the right of the **Linear Z** field and then select **Random Between Two Constants**.

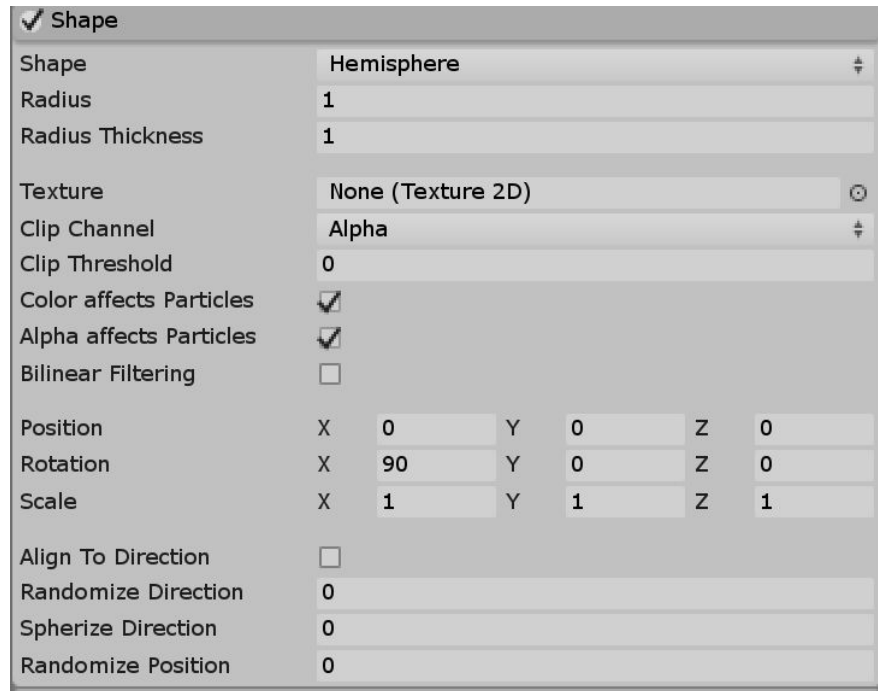


Lightning is fast and travels down. Therefore, let's set **Y** to -15 and -20 for now as they look fast and are going down. It doesn't matter in what order you place the numbers but it helps to be consistent so in my case, I'll try to always put the smaller number before the larger number.



We have a slight issue now where we actually can't see our particle systems from our camera because they are flying below the floor. Let's modify the **position** of the game object to (0, 6.08, -5.33). You can move it somewhere else but I liked the way the lightning looked from this position at the very end. We can now close the **Velocity over Lifetime** module (feel free to come back to it later to tweak the speed as you see fit).

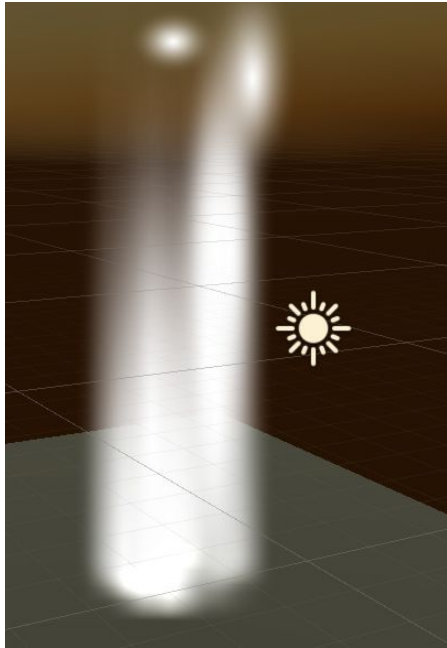
The next step I usually take is setting the **Shape** module. This module allows us to set the type of shape that the particles spawn from. By default it's a cone. This doesn't look very good so let's open the **Shape** module and set the **Shape** field to **Hemisphere** and its **Rotation** to 90.



I think this looks fine for now so I move on to the next bit. I think it's about time to give the particle an actual lightning-like shape. This is going to be a little bit more complicated than the previous parts because it requires tweaking a few modules instead of just one.

The modules we will work with are the **Main**, **Trails**, and **Renderer** modules so go ahead and open them up. Let's start with the **Renderer**. As it stands, we have white balls flying down. That is not what lightning looks like and in fact, lightning doesn't have any sort of spherical head thing so let's just get rid of it. We get rid of it by setting the **Render Mode** field to **None** (by default it should be set to **Billboard**). Oh no! Now we don't see anything! That's fine though. Next up, let's go ahead and set a **Trail Material**. The only reasonable option is **Default-Particle** so go ahead and select that. We still don't see anything but now, if we activate the **Trail** module, we will see the path that our particles take. The path is drawn using the **Trail Material** that we just selected so if we were to pick something else, the path would look different. And if we select nothing, there would be a pink path (that's the default for "no image" lines).

Hmmm...



I don't like it. Let's look at the **Main** module. In it, you will find a **Start Size** field. Like for the **Velocity over Lifetime** module, make this into a **Random Between Two Constants**. Set the first field to 0.1 and the second to 0.2 to make our particles skinnier. Much better.

Next up, we want to give the lightning its signature look and we are going to use one of my favorite modules to accomplish it. Let's open up and activate the **Noise** module. This one essentially modifies particular fields of each particle based on the values provided. Ok so now the particles are wavy. Let's see if we can make them take sharper turns. We will do this by increasing the **Strength** field to 6. A little better; let's actually go back into our **Trails** module and change the **Minimum Vertex Distance** to 0.8 so that the space between each stored vertex is longer (closer to 0 values will make the trail smoother while further from 0 values make the trail more edgy). You can actually close the **Trails** and **Renderer** modules now (if you haven't already).

If you look very carefully, you'll notice that all of the particles are actually taking the same path. If you deactivate the **Shape** module, this will be a lot more apparent. To fix that, we change the **Scroll Speed** field in our **Noise** module. Go ahead and change it to 0.7 and you can tweak it later. You know what, all of a sudden I don't like the shape of the particle spawner so rather than changing it, I will just disable the module completely. The particles seem to be taking a very similar path downward so to fix it this time, we modify the **Noise** modules **Frequency** field to 3. I'm starting to feel satisfied with how it looks but we aren't done yet!

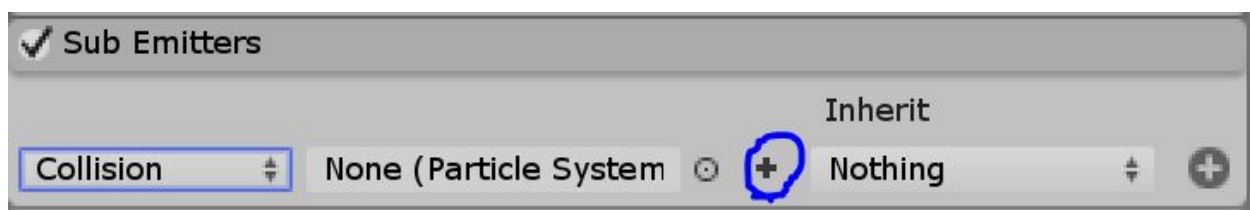
The next change is just to fix something that bothers me as I work on this. If you zoom out in the editor, you will see that the lightning extends WAYYYYYY down. To fix this, we set the **Start Lifetime** in the **Main** module to 1.5 so every particle dies in just 1.5 seconds instead of the default 5 seconds.

Only two things left!! The next bit we'll look at is the color. The lightning in our picture is purplish blue so let's see what we can do about that. I hope you expected this but go ahead and in the **Main** module, to the right of the **Start Color** field, select the dropdown and choose **Random Between Two Colors**. For the first color, I like the color code EAA5F1 and for the second color I like the color code 9DD6E9 but feel free to use your own (in my games I actually tend to use yellow and blue for my lightnings). If you want, you can make red or green lightning. It's all up to how you envision your lightning.

One thing left!! Sadly this one is actually a little complicated. Here's an overview of what I want to accomplish: when lightning collides with something, I want there to be a small spark because to me, that just makes sense!

We will now activate two new modules: **Collision** and **Sub Emitters**. Let's start by looking at the **Collision** module. Since we want our lightning colliding with everything that has a collider, we'll set the **Type** field to **World** and we get weird behaviour...Let's fix it by killing the particles soon after a collision and greatly decreasing how much it bounces off whatever it collides with. So to kill it, we set **Lifetime Loss** to 0.6 (this removes 60% of the particle's TOTAL life, NOT how much longer it can be alive. So in our case, $0.6 * 1.5 = 0.9$ seconds). Let's decrease the **Bounce** to 0.1 and increase **Dampen** to 0.95 (if you want to know what they do, hover your cursor over the each word and Unity actually provides a good explanation as to the behaviour of each field).

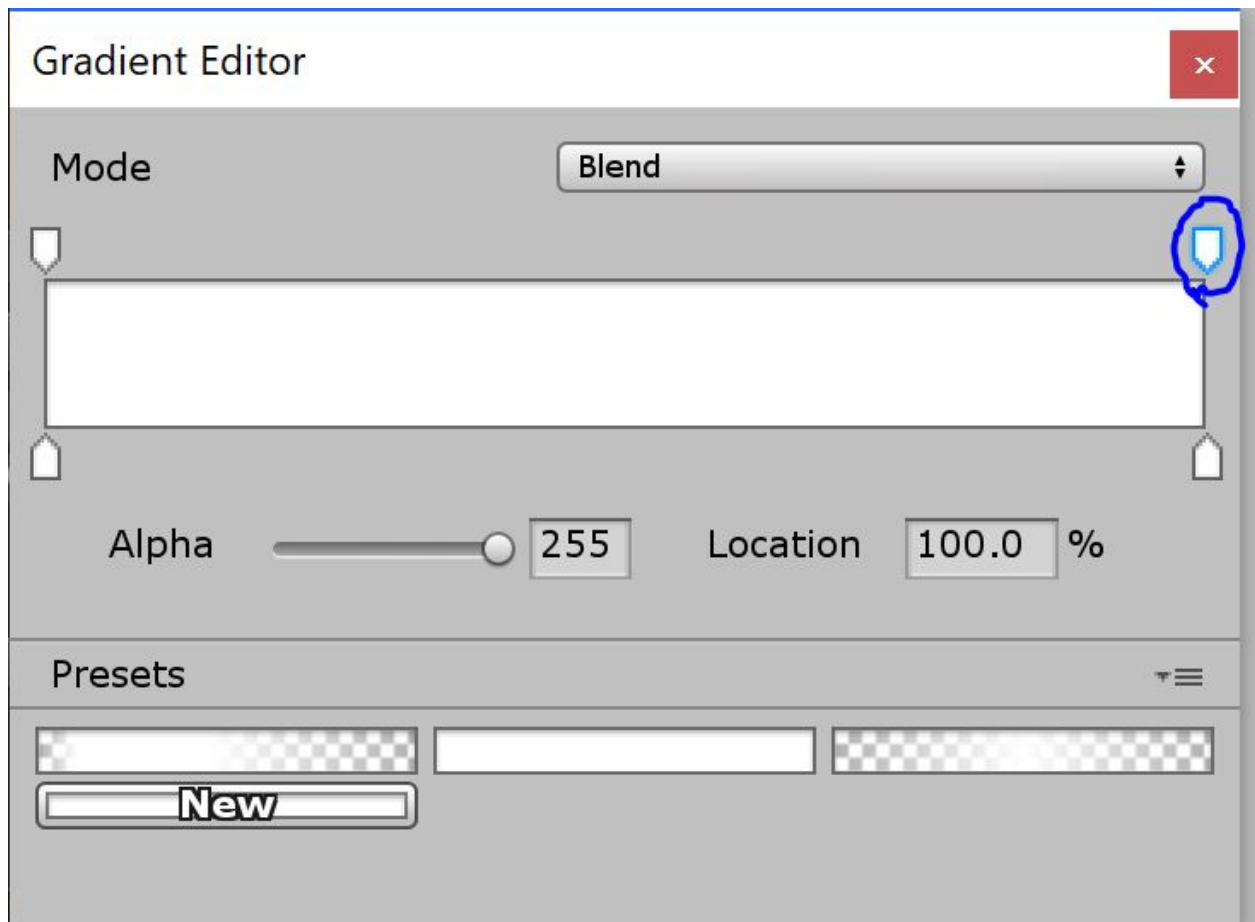
Now for our **Sub Emitter** module. You'll see that it says **Birth**, click that and select **Collision** then press the little plus icon in the center NOT the one on the right.



SO MANY PARTICLES!! Let's go ahead and go into our newly created particle system that is actually the child of the lightning game object (look at your **Hierarchy** view). Let's decrease the **Start Lifetime** to something small like 0.2 and 0.5 (I'm using **Random Between Two**

Constants). My lightning has very small explosions when it collides with something. In fact, let's also set our **Start Size** to be between 0.1 and 0.2. The final module we will go over is the **Color over Lifetime** module and just as you might expect, this one controls the color of each particle is throughout its lifetime.

We actually won't do too much here. Activate the module now and then click on the color to select a new color. You'll notice that this is actually not the usual color picker window we get. This is actually the gradient window. On the left is the color and alpha that the particle starts with while the right is how it will look at the end of its life.



Above I highlighted the pointer I want you to click. After clicking on it, set the **Alpha** field to 0. This way, the particles from the explosion will get gradually more transparent until they are completely gone. This makes it more natural (kind of like smoke) and easier on the eyes (rather than suddenly disappearing as before). In the **Gradient Editor**, the top pointers tell the alpha value of a particle at that point in its life while the bottom pointers tell the color value of a particle at that point in its life.

WE ARE DONE!!! We created lightning and hopefully, you understand exactly why each decision was made.

Unity's particle system may seem a little bit scary because it has so much to it. I think the best way to approach creating a particle system is to begin by having a good image of the end product in your mind and then figuring out which modules to activate and what fields to modify to reach your goal particle. I also demonstrated throughout the tutorial how often times you will work and all of a sudden decide to change your mind about a previous step and that is absolutely fine.

CHECKOFF

- Make a rain effect
 - You are not allowed to set Render Mode to None (you are allowed to use trails)
- Make an atmospheric effect
 - Use this video for reference - https://www.youtube.com/watch?v=xZL_25RY45g
- Create your own effect ****READ ALL BEFORE STARTING****
 - Come up with an idea and run it by a facilitator. The facilitator will do their best to approximate the difficulty level and based on that, either make that effect or decide on a different one
- Challenge:
 - Make a fireball that uses at least 4 particle systems