Laser Material

Dragged in the assets provided and that consisted of – a fire distortion texture and fire endpoint texture.

Create a new material called M\_Laser with blend mode as masked and shading mode as unlit. Add a particle color node and attach it to emissive color. To energize the laser i.e. to make it look as if energy is being sent out of it, here we make it look a bit distorted for that added a reference to the fire distortion texture and extended the R(as the texture was created with red as the distortion ref) and added a power (base,exp) node and set the R to the base and set the exp as 0.55. Extended the result and hooked it into the opacity mask. On clicking apply, the distortion should be seen. Instead of using the sphere preview chose the plane preview. From the texture node, extended the UV input pin and added a texture co-ordinate node and set the co-ordinate index to 1. This is important as we are going to be using a ribbon particle and a texture with co-ordinate index default as 0 would’ve made the texture on the ribbon stretch thereby giving it an undesirable layered look but a texture with co-ordinate index as 1 when used on the ribbon would make the texture be repeated instead of stretched in the ribbon and it would look appropriately distorted.

Ribbons and textures

In the ribbon particle we can set the size of the texture used on it by changing the UVTilingDistance parameter. We use a repeated texture instead of stretching it as the length of the laser is going to vary so we need to ensure it looks uniform at all lengths. To make the texture move on the material we can drag the output pin of the texture co-ordinate node and add a Panner node and set the speed X as say -0.8 and then attach the output to the UV input pin. We can also try using the time and speed Y options of the panner node to tweak the results further. Save the material.

Laser System

In the content browser right click > Select FX > Create a Niagra System. From the popup choose the New system from selected emitter option and click next. Select the dynamic beam option and name the file as P\_Laser. On the dynamic beam node check the ribbon render option. In the right hand window, the parameter of the ribbon render are shown, set the material as M-Laser. The dynamic beam looks like a needle but we need the width to be uniform so click the beam width option on the dynamic beam node and in the right hand window there’s a default arrow that would revert the current width settings near the beam with option. Click on the default button so that just the beam with and beam twist amount are shown also in the output the width becomes uniform. Set the beam width to 8. Selected the ribbon renderer and set the UV1TilingDistance to 100. Set the UV1Scale’s X to 0.3 and Y to 2.0 to give it a more streamlined appearance. Selected the spawn burst instantaneous option and set the spawn count to 2.

Where the beam is going to end will vary and here we are going to update the beam end with each frame from the blueprint. In the left window in the user exposed option click the plus button and set the vector variable name to BeamEnd. To set the beam end to value of this variable click the Beam emitter setup and on the right window set the beam end option to the BeamEnd variable. Check the absolute beam end option so as to end it in a world location. Click the emitter state option and set the loop behavior to once. Click on the initialize particle option and uncheck the lifetime so as to disable it. We will be destroying the laser ourselves so we don’t need lifetime. Click the particle state option and uncheck the kill particles when lifetime option thereby disabling it. Delete the solve forces and velocity option as we are not going to use it at all. Click the color option and we see that the scale alpha is controlled using a curve which we don’t need so we set it to default i.e. 1.0.

With this you might notice that the particle is no longer shown in the preview this is because when the beam end was set to BeamEnd it was given its default value 0,0,0. To set another value just click the P\_Laser node and in the parameters you’ll see BeamEnd and set the X to say 300. Click on the color option on the dynamic beam node and on the right-hand window next to the scale color option click the drop-down arrow and select the vector from float option. Below it near the scale option click the drop-down arrow and select the multiply float option and set the A value to 100. Go to the user exposed option on the left side-window and make a float variable called Scale. This variable will be used to gradually dissipate the laser. The preview should look black now. This is because the default value of Scale is 0 so the color becomes black. Click on the Scale variable and set the default value to 1.0. Now the preview should look blue. On the dynamic beam node click on the + button next to the particle update option and add a scale ribbon width parameter. On the right-hand window next to the ribbon width scale option, set it to Scale variable. To do so, just drag the scale variable and place it on the option. With this the initial laser setup is done.

To improve it further, near the ribbon width scale option click on the drop-down arrow and select the float from curve option and set the scale curve value to the Scale variable. Change the curve index from NormalizeAge to Sine. This makes the laser vibrate. There will be 2 points in the graph. Click on the second one ( the one which is below ) and set its value to 1.0 so both the points lie on the same line. Set the bias value to 0.5 and scale value to 0.5. Between the 2 keys on the graph at the center of the line add a new key and tweak the value to get the vibrating effect. When we tweak the value, the line between the keys might get a steep. To change that just click on the key and a line controlling the slope of the key appears. Just change its rotation. For now set the value to the middle key to 0.7. To control the frequency of the vibration set the Period to 0.35.

Now to give a gradual emergence effect to the beam, next to the ribbon width scale option hover on the float from curve and copy it. Click on the drop-down and select the multiply float option. Click on A and paste the float from curve. Click on B and select the float from curve option and set the curve index from NormalizedAge to System Age. The keys of B are positioned in such a way so as to give a gradual dissipation effect. To give it a gradual emergence effect, set the 1st key value to 0 and 2nd key value to 1. To control the time taken for emergence just change the key’s x value i.e., the time value. Also tweak the slopes accordingly.

Laser Implementation

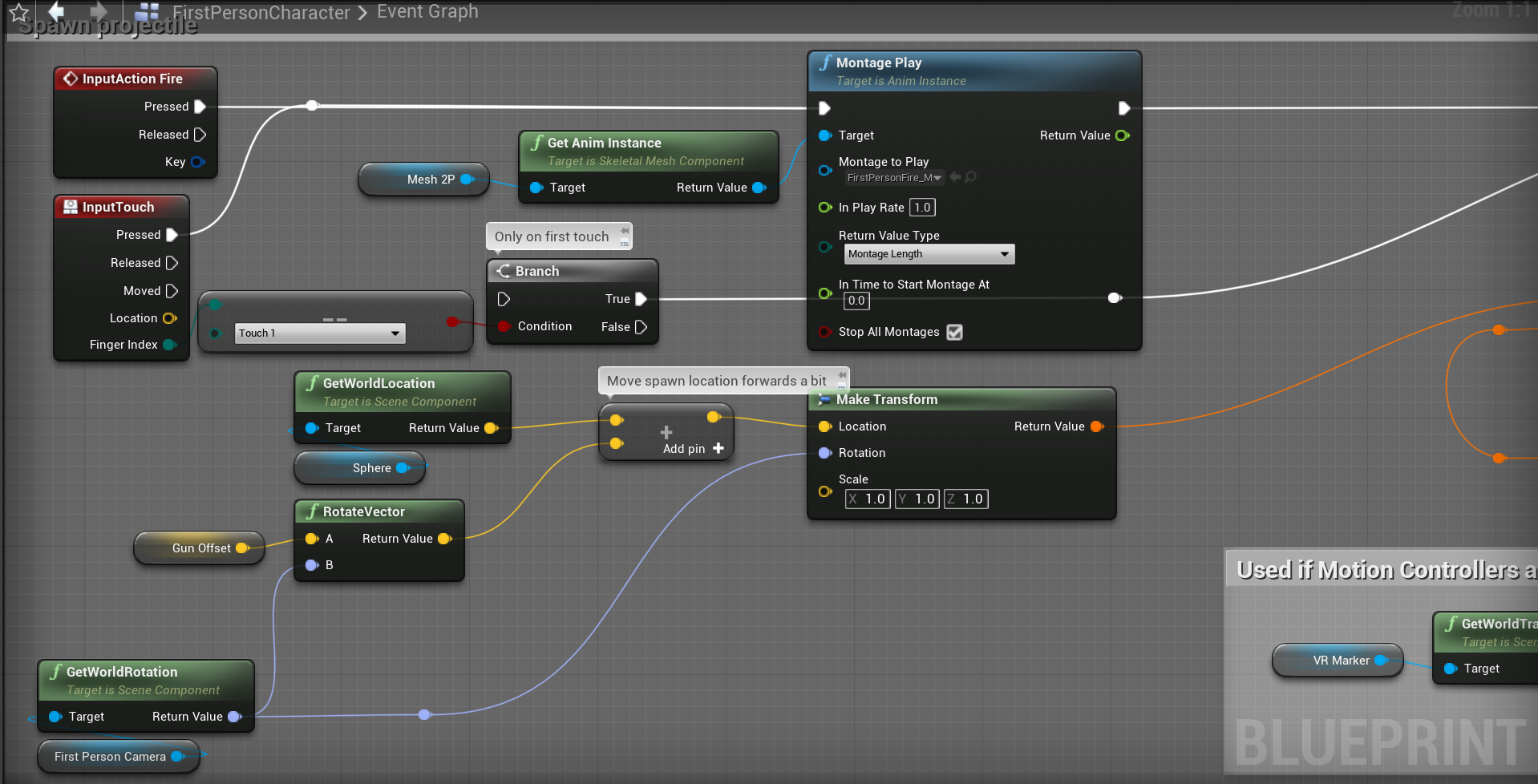
Mechanism: When you shoot a laser the end point keeps changing depending on the direction you shoot. So while you are shooting the end point of the laser needs to be calculated and the laser particle’s end point must be set to it.

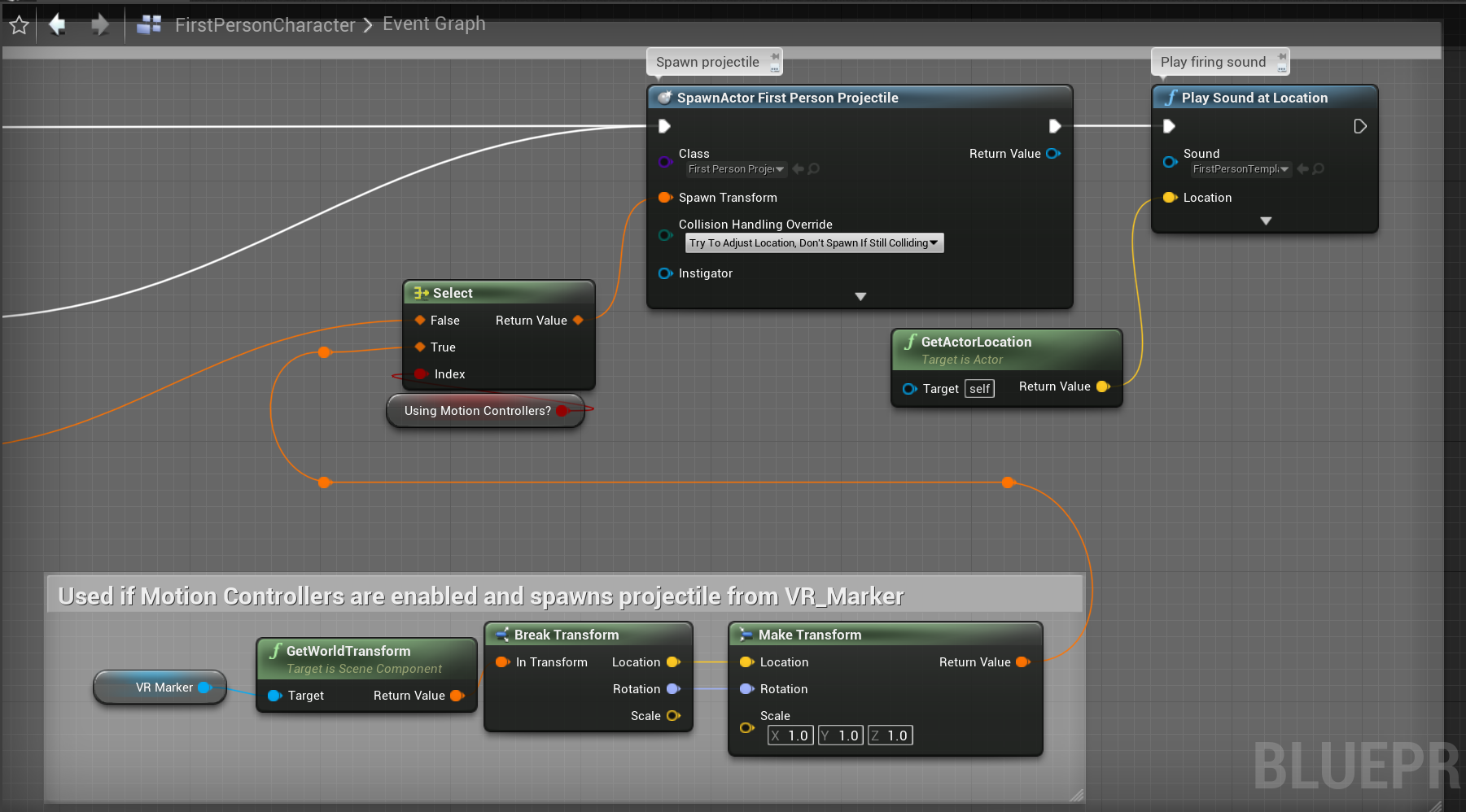
In this tutorial the FPS template is used and we are using the gun attached to the character. But in your project you might be using weapons as a separate object so make the changes accordingly.

Create an actor BP called BP\_Laser. In the viewport add a niagra particle component called laser and set it to P\_Laser. Create a variable of type same as that of the player character for example if it’s an FPS set the type as first person character object reference.

Open the First Person character blueprint and add an arrow component. This arrow will be used to position the system. The FPS template contains code for firing projectile based on where the player is looking. Borrow part of that code(in case you are using a different BP) / modify that code.

The original code:





You could modify it to:

In order to keep calculating the end point of the laser you need to come up with some kind of mechanism that starts calculating the laser’s end point during fire but the simplest way to do this is to use an event tick to keep calculating the laser’s endpoint and showing the laser only during fire. Extend the output exec pin of the event tick node from that add a get player view point node (parent is the first person variable) is added. Extended the location pin and promoted it to a variable called start trace. Extended the rotation pin and added a get rotation X vector node and promoted it to a variable called set shoot dir. Extended the return value pin of the set start trace node and added a vector - vector node and to the second pin added a get actor location node. Extended the return value pin and added a dot product node and to the second pin hooked the return value of the shoot dir node. Extended the return value of the dot product node and added a vector \* float node and to the vector pin added the value of the shoot dir node. Extended the vector \* float node and added a vector + vector node and to the second pin hook the value of start trace variable. Extend the start trace variable and add a vector + vector node and extend the second pin and add a vector \* float node. To the vector pin add the shoot dir node and to the float pin add the weapon range variable/ set the value to what you want the weapon range to be.

Extend the set shoot dir node and add a line trace by channel node and to the start pin hook the vector + vector node to which the start trace is connected and to the end pin hook the vector + vector node that has to do with the weapon range. Extend the out hit pin and add a break hit result node. Extend the hit component and add an is valid node and extend the return value and add a select vector node. Extend the impact point and hook it to A and extend the trace end and hook it to B. Drag in a reference to the laser variable and extend it and add a set vector parameter and set the parameter name to User BeamEnd and hook the return value of the select vector node to the param pin of the set vector parameter node. Hook the output exec pin of the line trace by channel node to the input exec pin of the set vector parameter node.

When you press fire all you just want to render the laser particle effect once as the endpoint is being changed during the event tick node. When the fire button is released, the laser should die down.

Add the fire input action node and extend the pressed pin and if you want you can add a sequence node and use the 1st pin for triggering the fire animation and sound etc. Extend the second pin and add a do once node extend the completed pin of the do once node and add a spawn Actor BP laser node and extend the target pin and add a self-node(self here is the first person character). Extend the return value and set it to the laser variable. Extend the output exec pin of the set laser node and add an attach actor to component node and set the target pin to the value of the laser node. Extend the parent pin and add a reference to where you want the laser to be attached. You could add an arrow component to the weapon and then use the arrow as the parent actor. Set the location rule and rotation rule to snap to target while the scale rule to keep relative.

But there is a problem if you are shooting and if the laser already exists we need to first destroy it before the spawning. To do this between the do once node and the spawn actor BP node add a reference to the laser variable and extend it and add a valid function. Extend the completed pin of the do once node and attach it to the input exec pin of the valid function. Extend the not valid pin and hook it to the spawn actor node and extend the valid pin and add a destroy component node and set the target to laser and hook the output pin to the spawn actor node.

Now on simulation, on fire the laser is spawned and also changes its length but it doesn’t stop. To do this make a new function or event called stop laser. Extend the released pin of the fire button and add a valid function. Set the target to laser and extend the is valid pin and add a stop laser node.

Go back to the definition of the stop laser node and extend it and add a timeline node and name it as stop effect. Open it and add a track and name it Scale. Make a key with time = 0 and value = 1 and another key with time = 0.3 and value = 0. Also set the total time to 0.3 as at 0.3 the value becomes 0 and beyond that we don’t need it to run. To change the interpolation from linear to something else just select the keys whose interpolation you want to change and right-click ON them and select the type of interpolation you want like auto, etc. Go back to the event graph and add a reference to the laser node and extend it and add a set float parameter node and set the parameter name to User Scale and hook its input exec pin to the output exec pin of the stop effect (the timeline) node. Hook the Scale pin of the stop effect node to the param pin of the set float parameter node. Extend the finished pin and add a Destroy Actor node.

Now on simulation when you press fire the laser appears and when you release the laser gradually fades out. It’s a good idea to go to project setting > Engine-Rendering > Uncheck motion blur.

With this you’ll have implemented a basic laser system and you can also implement new effects according to your desire as well as performance requirements.

Add lights to Laser

Go to the Laser\_FX and in the DynamicBeam node click on the + button near the Render option and select the Light Render option. Click on it and on the right-hand window set the radius scale to say 3.0 and expand the bindings option. As the intensity is low we will create a new parameter for it. In the left-hand window click on the + button next to the particle attributes option and select the linear color option and name it Light\_Color. Drag the variable to the dynamic beam node and place it below the Color option such that a new option is created called Set Light\_Color. Go to the left-hand window and click on the + button next to the User Exposed option and select the linear color option and name it Color. Click on it and assign it a color in the right-hand window we give it the same blue color of the laser. Now click on the Color option on the dynamic beam node and set it to the Color variable by dragging the variable and placing it on the Color parameter on the right-hand window. Click on the Set Light\_Color option and set the Light\_Color to Color variable by dragging the Color variable and placing it on the Light\_Color parameter on the right-hand window.

Next to the light color option is a small down-arrow mark. Click on it and select the multiply linear color option. To the linear color option set the Color variable by dragging it and placing it on the linear color option and click on the down-arrow next to the scale factor option and select the make vector 4 from float option. Click on the down -arrow button next to the float option and select the multiply float option and set the A value to say 10000 and the B value to the Scale variable by dragging it and placing it on the B value.

Click on the Light Render option on the dynamic beam node and on the right-hand window set the color binding to Particles Light\_Color.

Now on simulation light spawns where there are particles. To increase the light emitted by the laser just click on the spawn burst instantaneous option on the dynamic beam node and increase the spawn count to say 15. On simulation the light is much brighter.

Gun Sparks

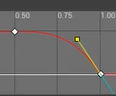
To send off gun sparks when the laser is emitted.

Go to Laser\_FX and in the system overview window – the one with the dynamic beam node, right-click and select the add emitter option and choose the empty emitter option. Rename it to Start\_Sparks. Click on the initialize particle option and on the right-hand window click on the down arrow next to the lifetime option and select the uniform ranged float option. Set the minimum to 0.5 and maximum to 1.0 check the position option and click on the down arrow beside it and select the Simulation Position option. Uncheck the mass option and the color option. Click on the down arrow next to the sprite size option and select the Vector 2D from Float option. Click on the down arrow next to the vector option and select the uniform ranged float option and set the minimum to 0.5 and maximum to 1.0. Click on the Sprite Render option and on the right-hand window change the alignment from Unaligned to Velocity Aligned. In the Start\_Sprite node, click on the + button next to the Particle Update option and select the Scale Sprite Size by Speed option. On the right-hand window set the Min Scale Factor to 1,1 and Max Scale Factor to 1,6. Set the velocity threshold to 500. The decrease in this speed is necessary to deform to the maximum. On the Start\_Sprite node, click on the + button next to the Emitter Update and select the spawn rate option. On the right-hand window set the spawn rate to 200 and check the spawn probability to 0.4.

On the Start\_Sprite node, click on the + button next to the Particle Update option and select the Color option. Set the color to that which you want the gun sparks to be eg: yellow. Click on the down arrow next to the scale color option and select the vector from float option. Click on the down arrow next to the value option and select the float from curve option. A linear curve appears but to make it sinusoidal select both of them and right-click on one of them and select the curve type as auto. Set the scale curve value to 350.

Now on simulation at the point of origin of the laser a yellow light appears. But it is a constant light whereas we want to decrease its size over the lifetime like a spark.

On the Start\_Sparks node, click on the scale sprite size by speed option and on the right-hand window click on the down arrow so as to expand all options. Click on the down arrow next to the initial sprite size and select the Vector 2D from float option and click on the down arrow next to the value option and select the float from curve option. A linear curve appears but select the first key and change the time to 0.3 and use the splines to make it like a quadrant.



On the Start\_Sparks node, click on the + button next to the particle spawn option and select the add velocity in cone option and on the right-hand window set the cone axis co-ordinate space from local to world. Click on the down arrow next to the velocity strength option and select the uniform ranged float option and set the minimum to 150 and maximum to 900 and set the cone angle to 300.

Now on simulation sparks start to fly in a cone. But we need to orient it in the direction of the main beam.

Click on the down arrow next to the curve axis button and select the normalize vector option. Click the down arrow next to the vector to normalize button and select the subtract vector option. Set A option to the BeamEnd variable by dragging it to the beam end option. Click on the down arrow next to B and select the simulation position option.

Now on simulation sparks start to fly and if you move the laser, it orients itself in the direction of the main beam. But if there is air resistance and gravity then we need the sparks to be pulled in that direction i.e., it needs to experience drag as well as the pull of gravity.

On the Start\_Sparks node, click on the + button next to the Particle Update and select the Drag option. A warning might appear saying that there are un-met dependencies on the left-hand window. To solve that click the Fix issues button. If the issue is fixed the warning disappears. Set the drag to 3. Click the + button next to Particle Update option and select the Curl Noise Force option. The warning appears again. Select the fix issue button and the warning disappears if it’s solved. Set the noise strength to 1000.

Now on simulation the sparks appear as if they experience drag.

Click on the + button next to the Particle Update option and select the Gravity Force button. The warning might appear again. Click the fix issues button and if it’s solved the warning disappears. Set the gravity to 0,0,-280.

Now on simulation the sparks seem to move in the direction of gravity too.

Finally if you want you can add a burst of sparks when the laser is first triggered. To do that click on the + button next to the Emitter Update option and select the spawn burst instantaneous option and on the right-hand window set the spawn count to 100.

Now on simulation whenever the laser is first emitted a burst of gun sparks is seen.

Laser Sparks

Similar to gun sparks

Copy the Start\_Sparks node and paste it and rename the node to Laser\_Sparks. Delete the gravity force option and the add velocity in cone option. Click on the + button next to the particle spawn option and select the add velocity option and on the right-hand window click on the down arrow next to the velocity option and select the random vector option and click on the down arrow option next to the vector scale option and select the uniform ranged float option. Set the minimum to 1 and maximum to 30. On the Laser\_Sparks node select the curl noise force and set the noise strength to 300. Select the Color option on the Laser\_Sparks node and set the color parameter to the Color variable by dragging it to the color parameter.

Now on simulation particles with the same color as that of the laser are seen but they are just near the point of origin. We need to make them go along the path of the main beam.

On the Laser\_Sparks node click on the Initialize Particle option and on the right-hand window click on the down arrow next to the position option and select the add vector option. Click on the down arrow next to the A option and select the simulate position option. Click on the down arrow next to the B option and select the multiply vector by float option. Click on the down arrow next to the vector option of the multiply vector by float and select the subtract vector option. Set the A option of the subtract vector to the BeamEnd variable. Click on the down arrow next to the B option of the Subtract vector and select the simulate position option.

Click on the down arrow next to the Float option of the multiply vector by float and select the uniform ranged float option.

Now on simulation blue laser sparks are seen along the main beam. To increase their lifetime, click on the initialize particle option on the Laser\_Sparks node and on the right-hand window below the lifetime set the minimum to 1.5 and maximum to 3.0.

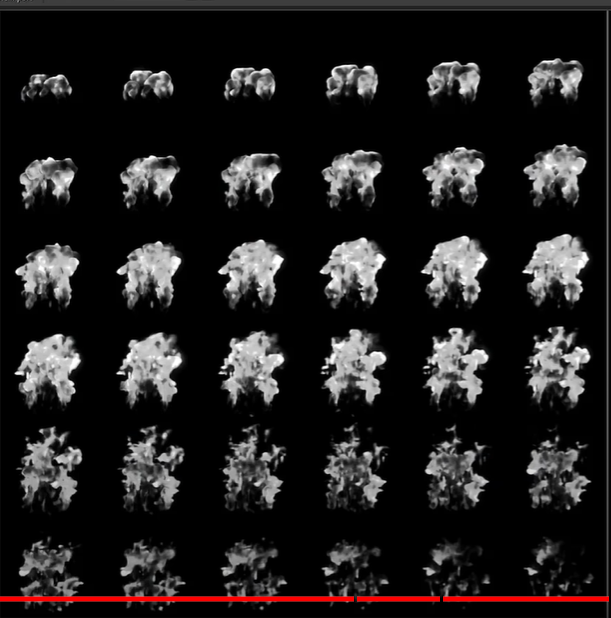
Disappearing Problem

Although all the additional effects have been included when we stop firing the laser the main beam slowly fades away but the effects disappear at once when you destroy the component/actor. To make the sparks linger even after the main beam disappears all you need to do is to increase the time in the Stop Effect timeline we set it to 0.3s before but if you change it to say 3s then even though the main beam fades after 0.3s the sparks linger till the laser actor is destroyed.

Shot Effect

To get a plump shot effect at the tip of the gun while shooting.

T\_Fire\_SubUV has been provided with the starter pack and it contains info that makes a fire blob like structure.



Create a new material called M\_LaserHit and set the blend mode from opaque to masked and the shading mode from default lit to unlit. Drag in a reference of the T\_Fire\_SubUV and hook the R channel to the opacity mask pin. Now on the preview pane parts of it are transparent while other parts are black. Extend the emissive color pin and add a particle color node and extend the exec pin and add a multiply node and to the B pin hook the R channel of the Texture node. Hook the result pin of the multiply node to the emissive color pin of the material node. Now on the preview pane the black parts will become white.

Create a new material called M\_Laser\_Translucent and change the blend mode to Translucent and shading mode to Unlit. Go to M\_LaserHit and copy all the nodes except the material node and paste them in M\_Laser\_Translucent node and hook the result of the multiply node to the emissive color pin of the material node. Extend the red pin of the texture node and add a multiply node and to the B pin hook the alpha value of the particle color node. Extend the result pin of the multiply node and hook it to the opacity pin of the material node. In the preview pane the result is same as the M\_LaserHit except for the fact that it’s translucent.

Working – In the texture we have different images for the states of the flares and we choose one of these for each particle.

Right-click in the content browser, select the FX option, select the niagra system option and choose the new system from selected emitter option and add the empty emitter and click on the finish button. Name the file as P\_Weapon\_Fire. Open it and you’ll see the empty node. Rename it to Weapon\_Fire. Select the sprite renderer option and on the right-hand window change the material to M\_Laser\_Translucent. As the texture used in the material is 6x6 we change the sub image size to 6,6 and check the sub UV blending enabled option. Click on the + button next to the particle attributes option on the left-hand window and select the float option and name the variable to SubImageIndex. Drag it to the Weapon\_Fire node and place it below the Particle State option in the Particle Update section such that a new option is created called Set SubImageIndex. On the right-hand window click on the down arrow next to the SubImageIndex option and select the float from curve option. Click on the 2nd key and set its value to 36. Select both the keys, right-click on one of them and set the curve type to auto.

Click on the + button of the Emitter Update option and select the Spawn Rate option. Set the spawn rate to 5. Click on the Emitter Rate option on the Weapon\_Fire node and set the loop interval to Once and the loop duration mode to infinite. Click on the + button next to the Emitter Update option and select the spawn burst instantaneous option and set the spawn count to 1. Click on the initialize particle option on the Weapon\_Fire node and on the right-hand window click on the down arrow next to the Sprite Size option and select the Vector 2D from float option. Click on the down arrow next to the value option and select the Uniform Ranged Float option and set the minimum to 20 and maximum to 30. Check the sprite rotation option and click the down arrow beside it and select the uniform ranged float option and set the minimum to 0 and maximum to 360. Uncheck the mass option and the color option. Click on the down arrow next to the lifetime button and select the uniform ranged float option and set the minimum to 0.4 and maximum to 0.7.

Now on the preview plane the white fire is visible. Now we need to give it color.

Click on the + button beside the particle update option and select the color option. Set the color to the color you want. To make it brighter and more emissive, click on the down arrow next to the scale alpha option and select the float from curve option. Select the first key and set the time to 0.3 and adjust the splines so that the curve is like a quadrant. Click on the down arrow next to the scale color option and select the float from curve option. Click on the down arrow next to the value and select the float from curve option and click on both the keys and set the curve type to auto and then click on the 1st key and set the value to 300.

Now on the preview plane we see that the fire is like the Ori game color. Some parts of it are bright and some dark.



Now we need to vary the size of the fire.

Click on the + button next to the particle update option and select the scale sprite size option. Click on it and on the right hand window next to the scale factor option click on the down arrow and select the vector 2D from float option. Click on the down arrow next to the value option and select the float from curve option. Select the 1st key and set the value to 0.2 and the 2nd key value to 1. Select both keys and use the splines to set the curve to quadrant.

Just add the niagra system to the Laser BP or whatever you want and just deactivate it after fire. To do that in the viewport add a new particle system and called Weapon\_fire and set it to P\_Weapon\_Fire. Now on simulation the effect always appears. To stop it when the firing stops just go to the stop laser function definition and add a reference to the Weapon\_fire, extend it and add a deactivate node to deactivate the effect.

Now on simulation the effect appears and disappears on starting and ending of fire. However, when you move the gun around the effect does not move with it. To fix this go to the P\_Weapon\_Fire BP and select the emitter properties option on the Weapon\_Fire node and on the right-hand window check the local space option. Now the spawning happens in local space so now it moves with the gun.

Impact/Hit Effect

Right click on the content browser > FX > Niagra system > New system from selected emitter > Next > Empty > + > Finish and name the file as P\_Laser\_Hit. Open it and delete the empty node. From the P\_Weapon\_Fire BP copy the Weapon\_Fire node and paste it in the System Overview of P\_Laser\_Hit. Rename the node to Hit.

If you had checked the local space option in the while making Weapon\_Fire, here you need to change it. Click on the Emitter Properties option of the Hit node and uncheck the Local Space option.

Click on the sprite renderer and on the right-hand window change the material option to M\_LaserHit. Click on the spawn burst instantaneous option and delete it. Click on the spawn rate option and on the right hand window set the spawn rate to 120. Change the color and spawn, lifetime etc. to make it more discernable to the eye. Changing the sprite size by going to the initialize particle and changing the min and max values of the size is also recommended.

Instead of changing the sprite size another good option is to change the sub image index so that it choses only the big images from the texture. Now in the texture 0-5 images are small so we set the minimum index to 6. We do this by clicking on the set sub image index option of the hit node and click on the 1st key and set the value to 6. Now with a small size we still get big renders.

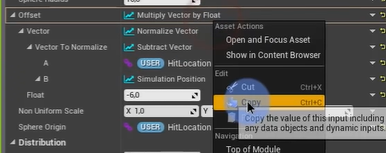
To control the location, click on the + button next to the particle spawn option and select the sphere location option. On the right-hand window set the radius to say 10. To set the location of the sphere, go to the left-hand window and click on the + button next to the User Exposed option and select the vector option and name it HitLocation. Click on the sphere location on the Hit node and on the right hand set the sphere origin to HitLocation by dragging it and placing it on the sphere origin option.

But right now, if you fire the effect is spawned right at the hit point. This point is the one where the cross-hair HUD points to and it might become difficult for the player to see the crosshair if the effect is spawned in the location the crosshair points to. To solve this, we add a small offset such as the cross-hair is distinct but at the same time the offset is not too much of a problem in precision.

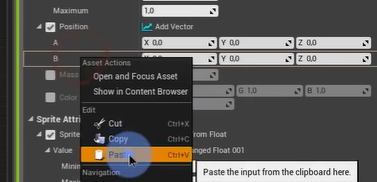
Click on the down arrow next to the offset button and select the multiply vector by float option. Set the float value to -6. Click on the down arrow next to the vector option and select the normalize vector option. Click on the down arrow next to the vector to normalize option and select the subtract vector option. Set the A value to HitLocation. Click on the down arrow next to the B value and select the simulate position option. Now there’s a small offset and you can vary this by changing the float value.

Now if you want to add the gun sparks effect to the hit effect just go to the FX\_Laser and copy the Start\_Sparks node and paste it beside the hit node. Rename it to Hit\_Sparks. You can vary the parameters as you like to get the desired results. Delete the spawn burst instantaneous as it’s overkill. Increase the spawn rate if you want it to be visible and increase the spawn probability. Increase the sprite size. Delete the add velocity in cone. Click on the + button next to the particle spawn option and select the add velocity option. Click on the down arrow next to the velocity option and select the random vector option. Click on the down arrow next to the vector scale option and select the uniform ranged float option. Set the minimum to 75 and maximum to 350. To set the location, click on the initialize position option on the Hit\_Sparks node. Click on the down arrow next to the position option and select the add vector option. Set the A option to HitLocation.

Click on the sphere location on the Hit node and copy the offset info like so:



This is to avoid redoing the whole thing. Go to the Add Velocity option on the Hit\_Sparks node and on the B option of the add vector option just paste the whole thing like so:



If you want more realistic effects provided you meet the performance requirements you can add collision to these sparks. Click on the + button next to the particle update option on the Hit\_Sparks node and select the Collision option. An error might arise which can be fixed by clicking the fix issues button. Beneath the bounce, set the restitution to 0.3 and check the randomize collision option.