2D DEVELOPMENT

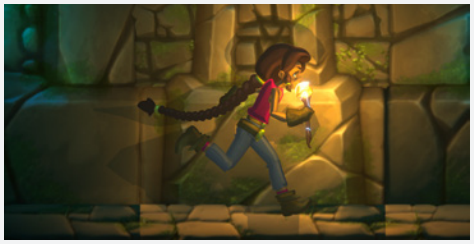
E-BOOK

Leveling up your 2D game

A step-by-step guide to the latest Unity 2D features used to create the popular Lost Crypt sample project

unity

Contents



Lost Crypt game demo

Lost Crypt is a lively scene featuring animation, light effects, organic terrain, shaders, and post-processing, all made natively in 2D. It shows how teams and projects of all sizes, targeting any platform, can now get more engaging and beautiful results faster.

You can download the demo here.

Exploring Lost Crypt

Are you a 2D artist or aspire to create a 2D game? You're in the right place.

While people have been making gorgeous 2D games with Unity for years, we know that we can support them even more – that's why we rolled out many new tools explicitly for 2D game-making.* And to make them super easy to learn and use, we created Lost Crypt, a sample project that showcases the complete suite of integrated 2D tools for creating high-end visuals with Unity.

In this e-book we guide you through setting up the Lost Crypt sample project and then explain how we used the different tools to bring this project to life. We hope this inspires you to try them out in your own projects.

* Available in Unity 2019 LTS and later versions.

Original concept art. We made the project in collaboration with B2tGame.

Setting up the project

Start by downloading the Lost Crypt sample project from the Unity Asset Store. We recommend that you start with a blank New Project and select 2D, then import the project from My Assets in the Package Manager, or by clicking My Assets on your Asset Store page. The project includes all the 2D packages you need. It will then overwrite the new-project settings, and change the rendering settings to the 2D Renderer within the Universal Render Pipeline.

After you import Lost Crypt, restart Unity to implement some changes to the project. When Unity opens, select Lost Crypt in the Project window. In the main Scene, click Play and use the keyboard arrows to move the character and the spacebar to make her jump.

The scripts and game logic are fairly simple, as the main focus when making the demo was to use the 2D tools to materialize the demo's concept art.

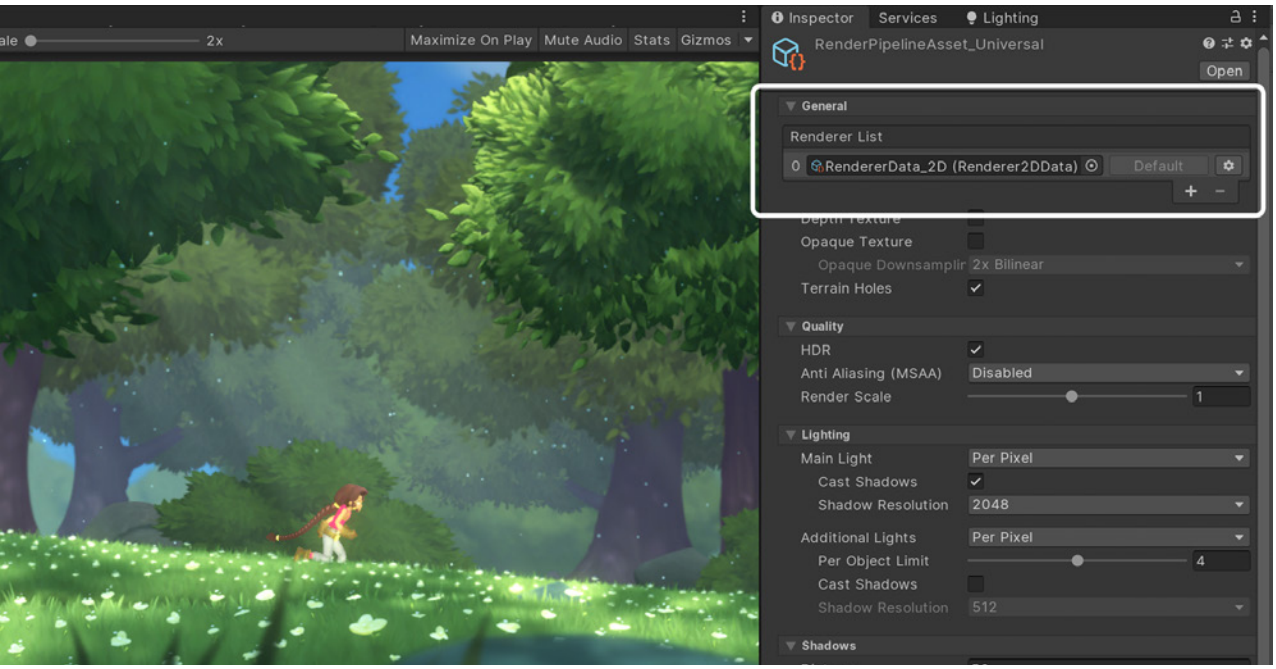
In the following sections, we've broken down the demo creation into discrete steps, and highlighted which 2D tools we leveraged for each unique visual challenge.

Tip!

Working on visually ambitious 2D projects

"Start with the best concept art possible, and keep an open state of mind: everything is possible. Break down your concepts into single elements, and think ahead for what you should do first and what will be reused."

— Michaël (Mika) Renaud,
Art Director, B2tGame



Here you can see Lost Crypt's render pipeline asset with the Universal Render Pipeline and the 2D Renderer.

Step 1 —

Creating a large outdoor organic textured terrain

One of the challenges that 2D game developers had in the early days was to efficiently create organic terrain like hills, slopes or irregular ground, which was only achievable through carefully crafted tile sheets. Years later, this was easier to achieve using multiple Sprites representing different parts of the terrain, but the workflow and performance were still not ideal.

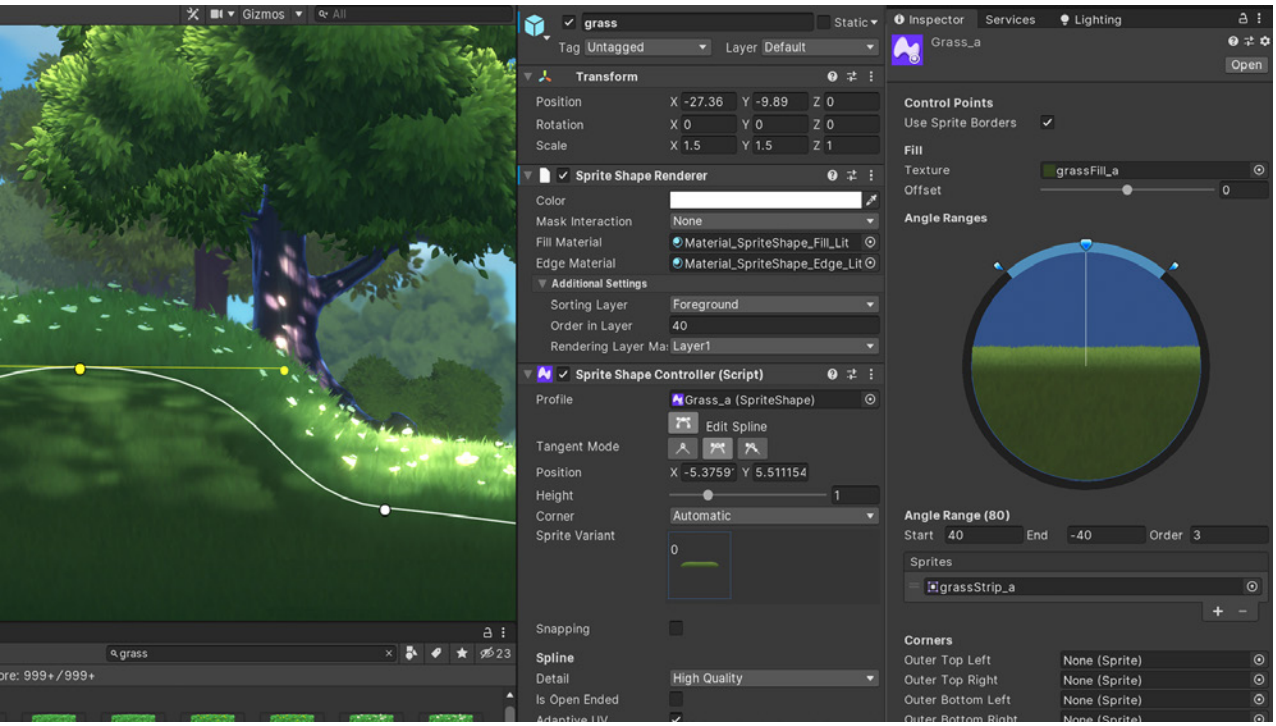
To solve some of these challenges, we developed Sprite Shape, which is a flexible world-building asset. It features Sprite tiling along a shape's outline that automatically deforms and swaps Sprites based on the angle of the outline. With 2D Sprite Shape you can generate terrain and

Tip!

Creating and formatting 2D assets

"I start pasting my concept in one big document. It helps me keep a consistent level of detail across all the Sprites. Then I make a list of all the different elements I will need to compose the Scene. This helps me see what must be painted, and what can be reused. After that, I start cleaning everything for integration. Once the asset is in Unity, it's a lot easier to polish."

— Michaël (Mika) Renaud,
Art Director, B2tGame



2D Sprite Shape was used for the ground layers and in one of the Sprite Shape Profiles on the right.

colliders similarly to how you would do it in a vector-based drawing application. You can adjust the brushes (Sprite Shape Profiles) and start creating without worrying about adjusting many Sprites or colliders as you iterate on the environment.

Select one of the handles to see how the shape and textures react to your changes. For example, you can play with different settings, select a node and change its Tangent Mode to create broken tangents or smooth tangents, change the height of the border and so on.

Lost Crypt also uses some of the Sprite Shape extras (available in the Package Manager) like the NodeAttach script to attach elements to the spline, so they follow the spline. In this demo, the rocks use this script and the ConformingSpline in the flowers layer follows the shape of the grass spline. You can use this feature for gameplay or for decorative elements like we did in the foreground grass layer.

Step 2 —

Building the crypt with a grid-based layout

Tilemaps is probably one of the most essential 2D tools, not only to save memory space with small graphics that can be “tile-able” and repeatable, but because it’s also crucial for level design.

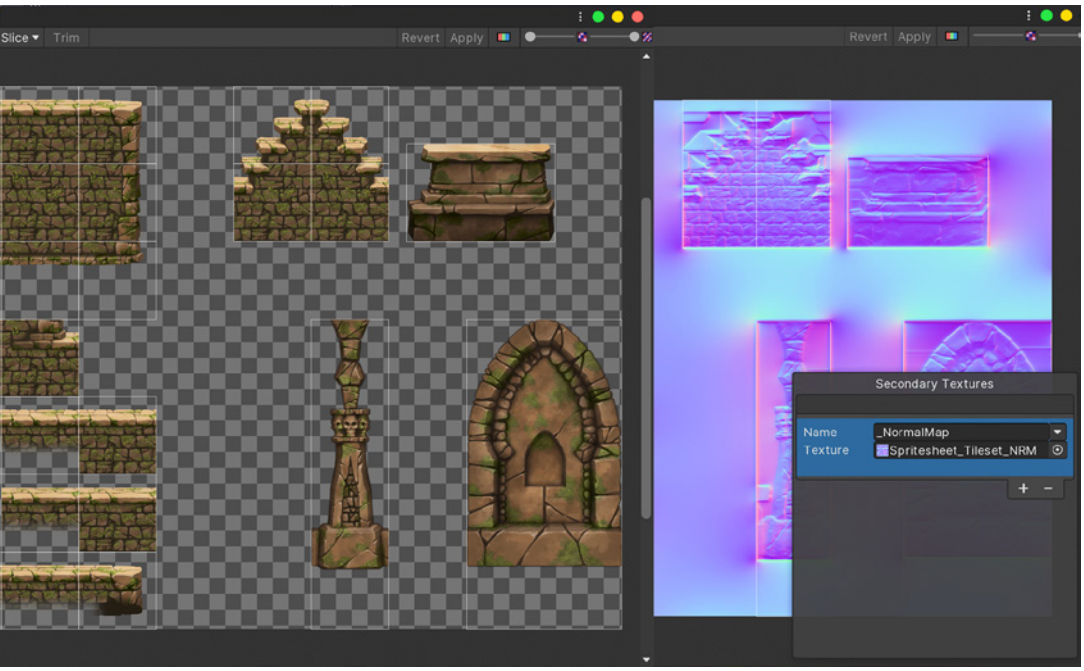


Image of the tile and normal maps sheet used in the demo

In Lost Crypt, we used the 2D Tilemap Editor (available via Package Manager) to recreate the crypt interior where it uses some additional 2D Tilemap Extras tools (available in the Package Manager in Unity 2020.2 and later) to make the

Tip!

#1
Preparing assets for the Tilemap tool

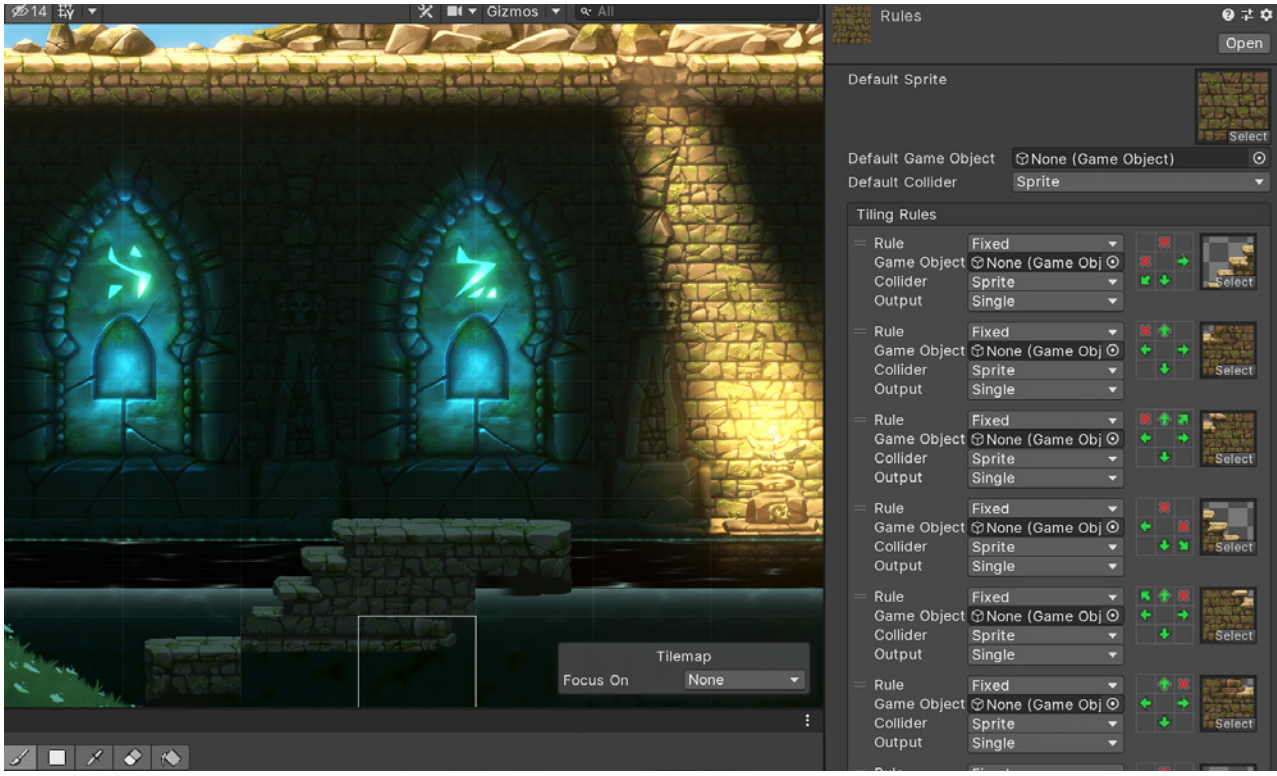
“Go to the Asset Store and use an existing tileset as a starting point if you can. Otherwise, I would encourage people to use some smart objects in Photoshop. Creating a tileset involves repeating some patterns a lot. Whenever you need to alter the overall look of the tileset, having a good portion of the tiles updated simultaneously saves a lot of time.”

#2
Preparing assets for the Tilemap tool

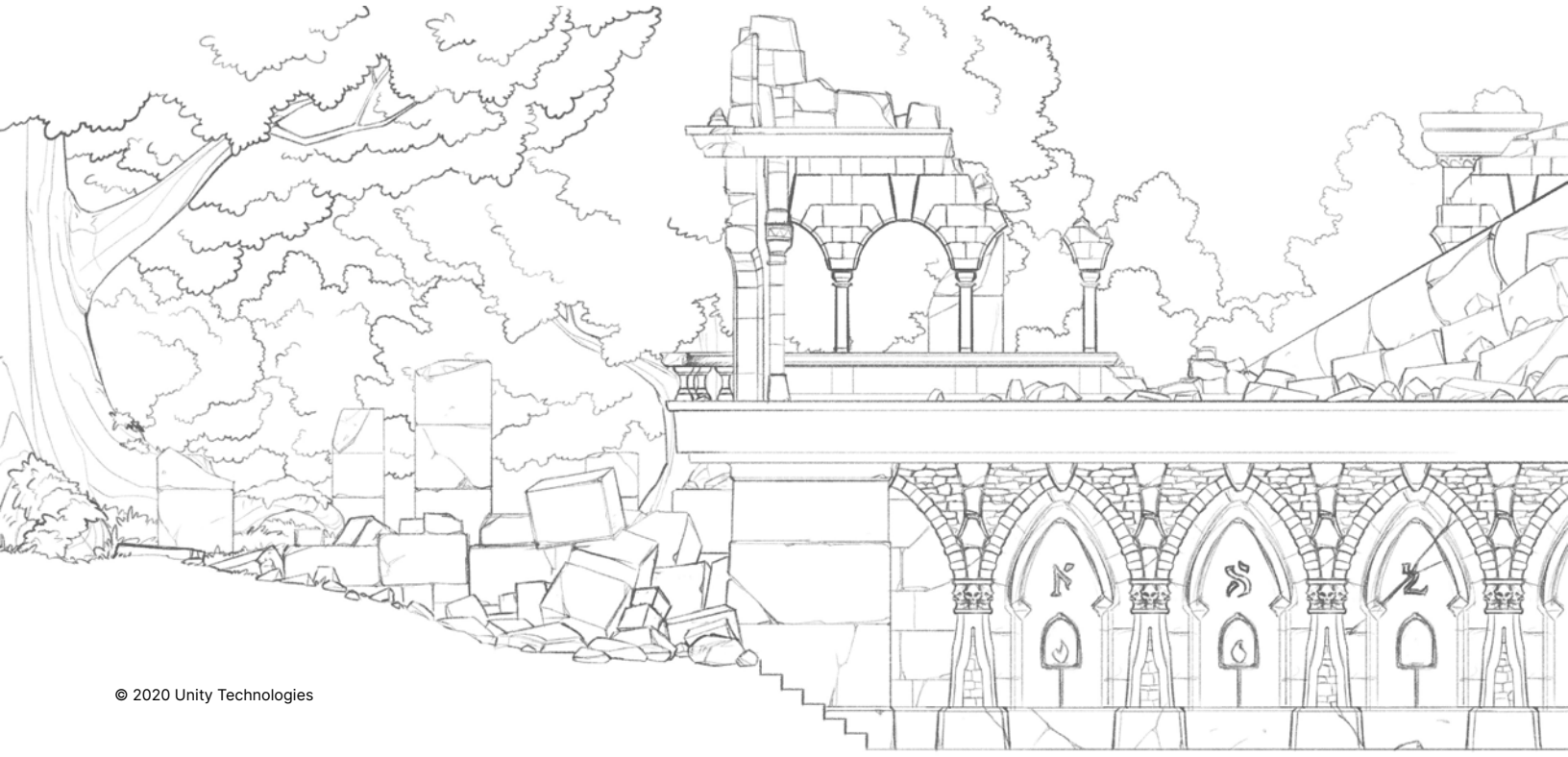
“There’s a built-in filter in Photoshop to generate normal maps from an image. Try it first on some assets you know, like a sphere, for example, to make sure it works well in Unity. There are different conventions on how the tool encodes the normals’ information, so you may have to experiment.”

— Michaël (Mika) Renaud,
Art Director, B2tGame

To improve the workflow, we used Rule Tiles in the crypt from the 2D Tilemap Extras package.



level-design process more efficient. For example, we used Rule Tiles, a tile type that lets you paint tiles like they're brushes. It automatically selects the right tile based on the neighboring tiles or ends, speeding up the workflow when you're creating scenes.



Step 3 —

Animating characters

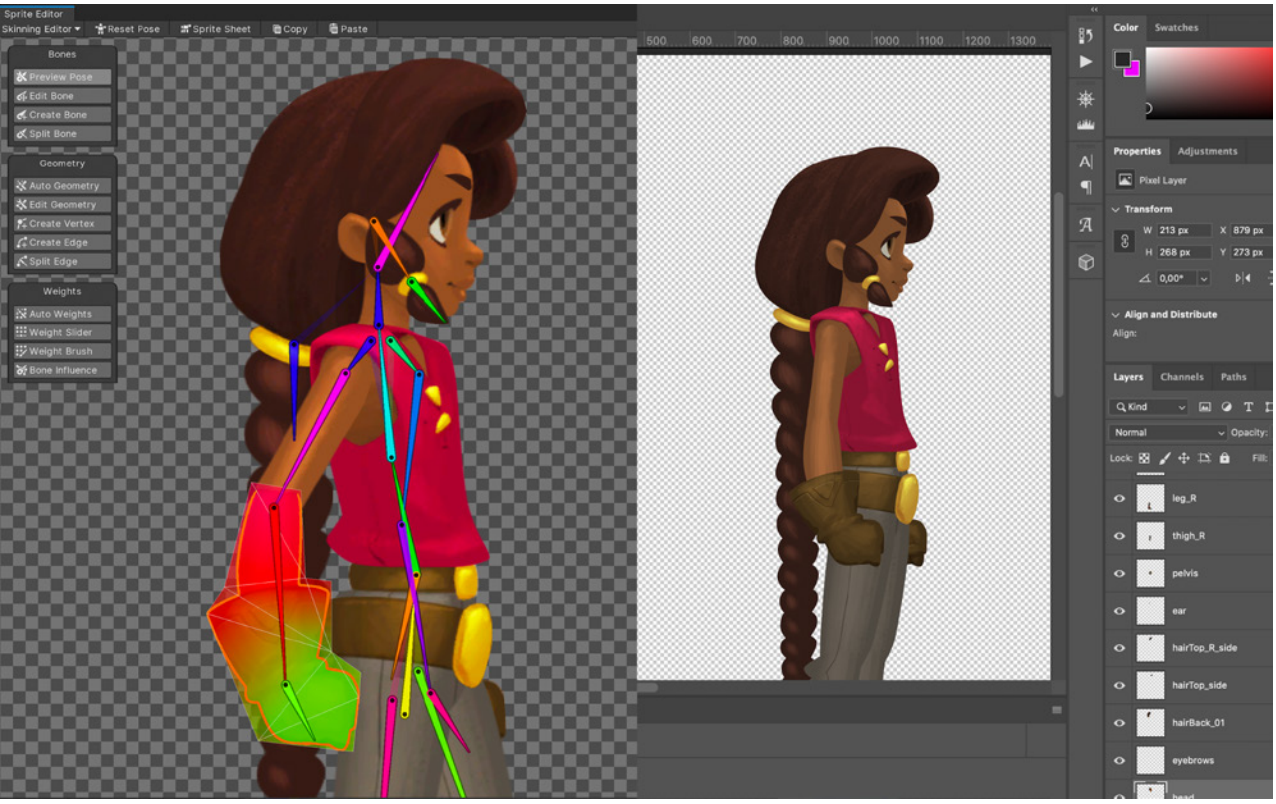
We designed the character in Photoshop and imported “Sara” directly with the 2D PSD Importer. Open the Sara.psb file with the Sprite Editor to see the character setup and rig. If you open the file with Photoshop, you will see how we kept the different parts and layer names intact.

Tip!

Leveraging the PSD Importer package

“It’s very convenient being able to import a PSD directly into Unity while retaining the layer structure. PSD Importer removes a step for artists and makes further modifications easier to maintain.”

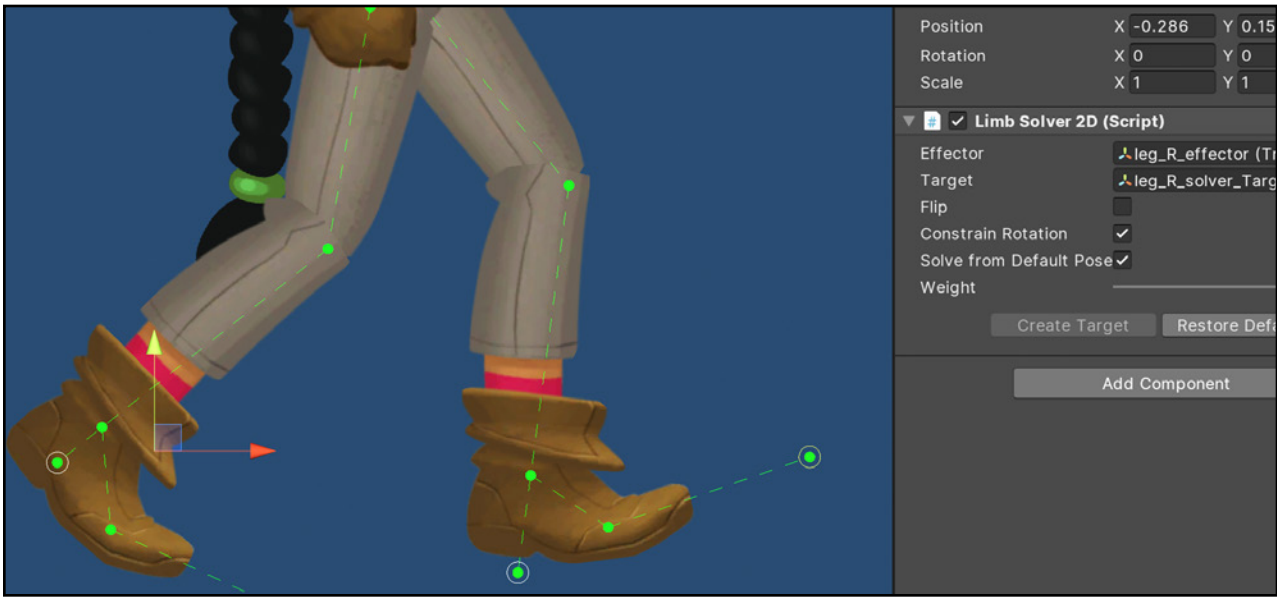
— Ivano (Ivan) Mansueto, Technical Director, B2tGame



The PSD Importer lets you use Photoshop files directly in Unity without having to export separated Sprites and reassemble them later.

Once we rigged the character, we made the different animations with the Animation tool and Animator to control those states. You can see how the tool works in a talk we gave at GDC 2019 (see the Additional resources section at the end).

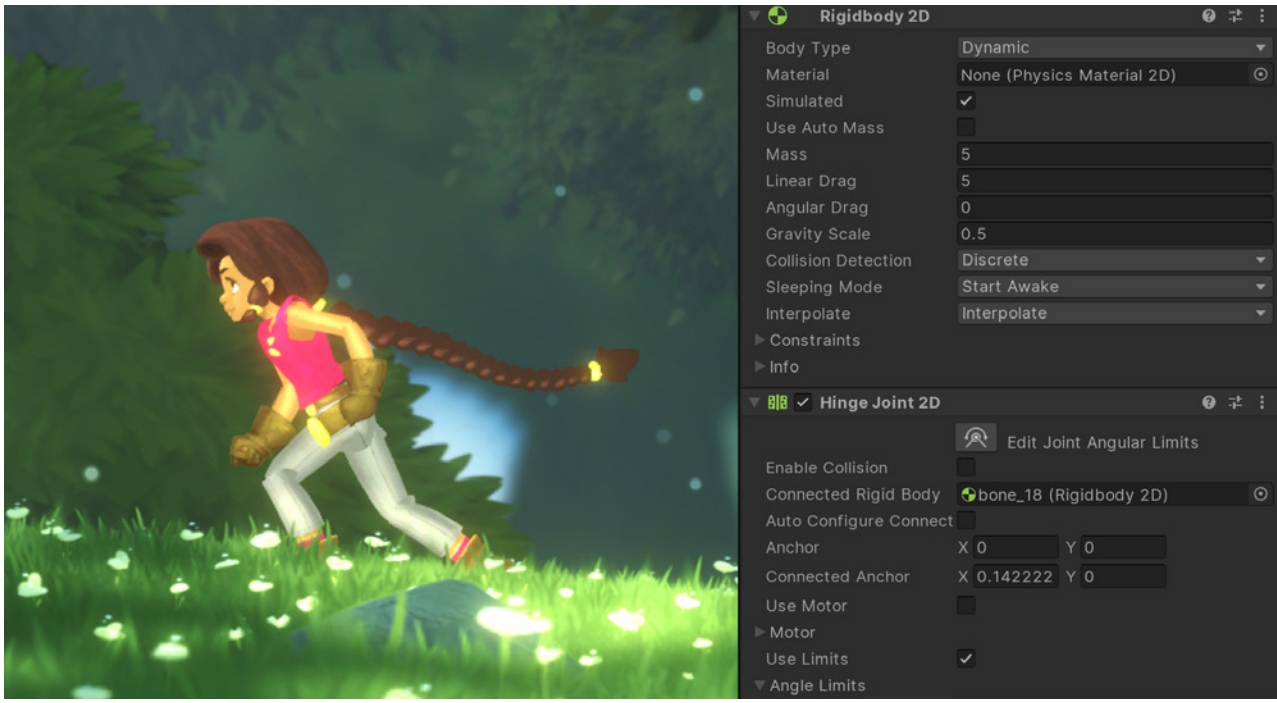
We used 2D IK to facilitate the animation process of the legs and feet.



The character has 2D IK solvers in her legs to help position the ankle and foot tips correctly, then the legs will follow realistically.

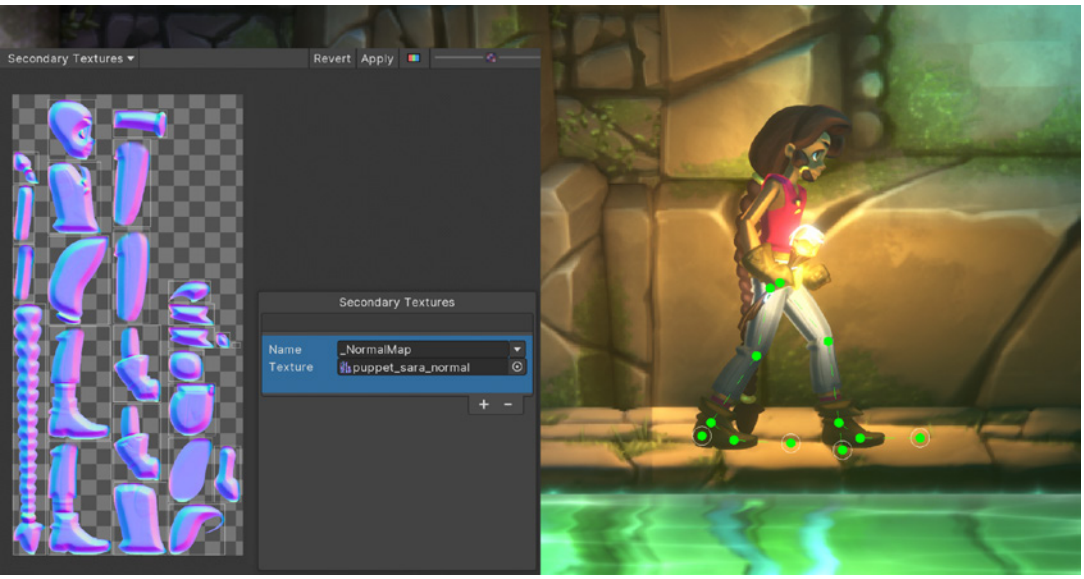
The character's ponytail is a different child GameObject and is controlled by 2D Physics. It reacts realistically to movement because every bone of the ponytail has a Hinge Joint 2D component with some restrictions. That allows her hair to move freely without curling too much or overreacting to the character movement.

The ponytail uses a 2D physics chain configuration with some weight at the end of the chain.



One of the features available with the Universal Render Pipeline is the new Sprite-Lit material. Compared to the usual Sprite-Default material, this one allows Sprites to react to 2D lighting conditions.

We imported the character Normal maps in the Sprite Editor, using the Secondary Textures drop-down menu. You can add Normal and Mask maps to 2D animated characters, regular Sprites, tilemaps, and Sprite shapes.



The Normal map for the character indicates which areas of the Sprite should reflect more or less light.

Tip!

Using third-party tools besides Photoshop

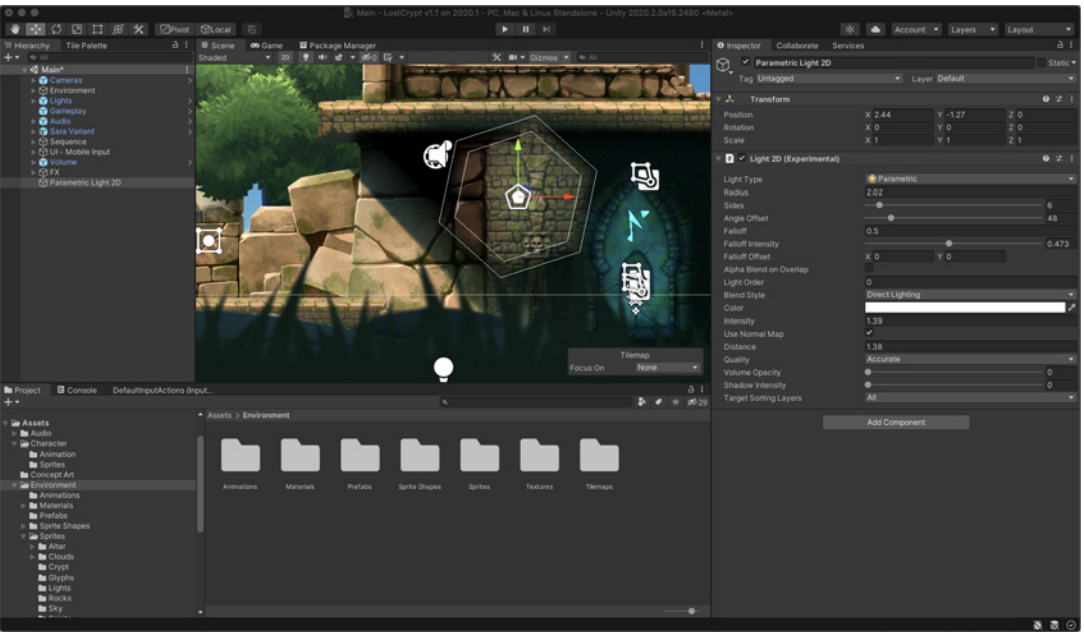
"Crazybump and all similar tools can be really helpful to generate some maps for very detailed objects. Blender is definitely an excellent tool to integrate into your production pipeline, even in 2D. It requires a small learning curve, but overall it gives you more solutions than any other 3D software and it's 100% free."

— Richard Rispoli, Creative Director, B2tGame

Step 4 —

Adding day/ night light cycles

One of the best reasons for having dynamic lighting on Sprites is to alter the appearance of the environment. Using the same Sprites, you can change the mood, time of day or darkness of an area, which opens up a wealth of gameplay mechanics from stealth mechanics to lively rich worlds.



You can quickly see how easy it is to add new lights and understand their options by going to the Scene view and creating a new 2D light in the top menu (GameObject>Light>2D). Select any type (you can change it later in the Inspector window). Make sure “All” is selected for your light Target Sorting Layers, then try different configurations or add more lights.

Tip!

Working on visually ambitious 2D projects

“Favor some more ‘neutral’ or local colors, as opposed to fully lit sprites. This will help you make sure your assets work well with a vast array of lighting scenarios.

Use the mask channels (RGBA) to integrate all the different lighting information: rim lights, subsurface scattering (SSS) lights, ambient lights, FX lights, etc. Then use a shader to get the most control on each element.

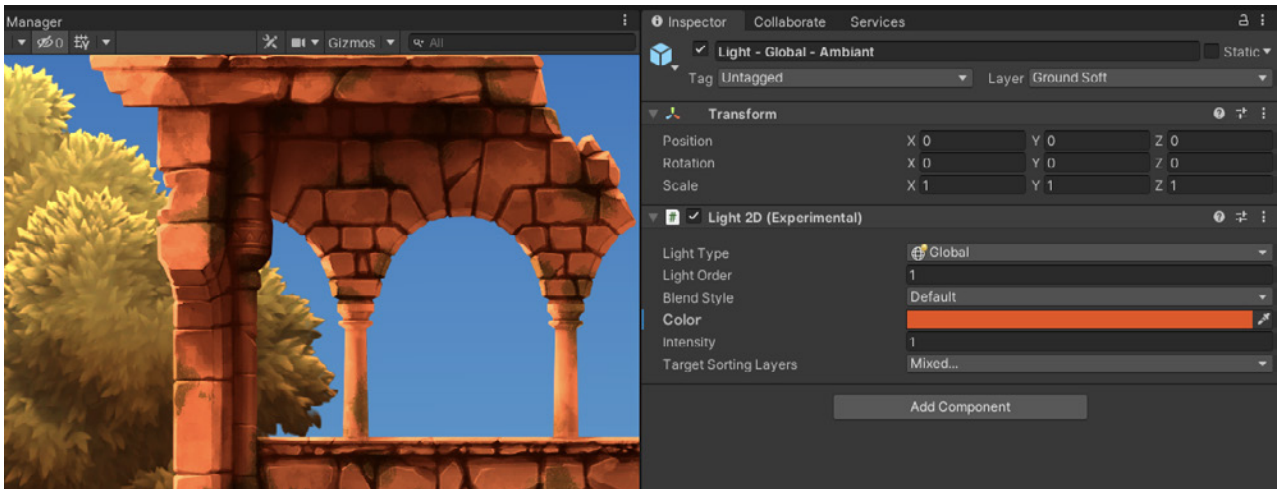
Remember that normals will give you texture and grain, but only work close to some dynamic light sources.”

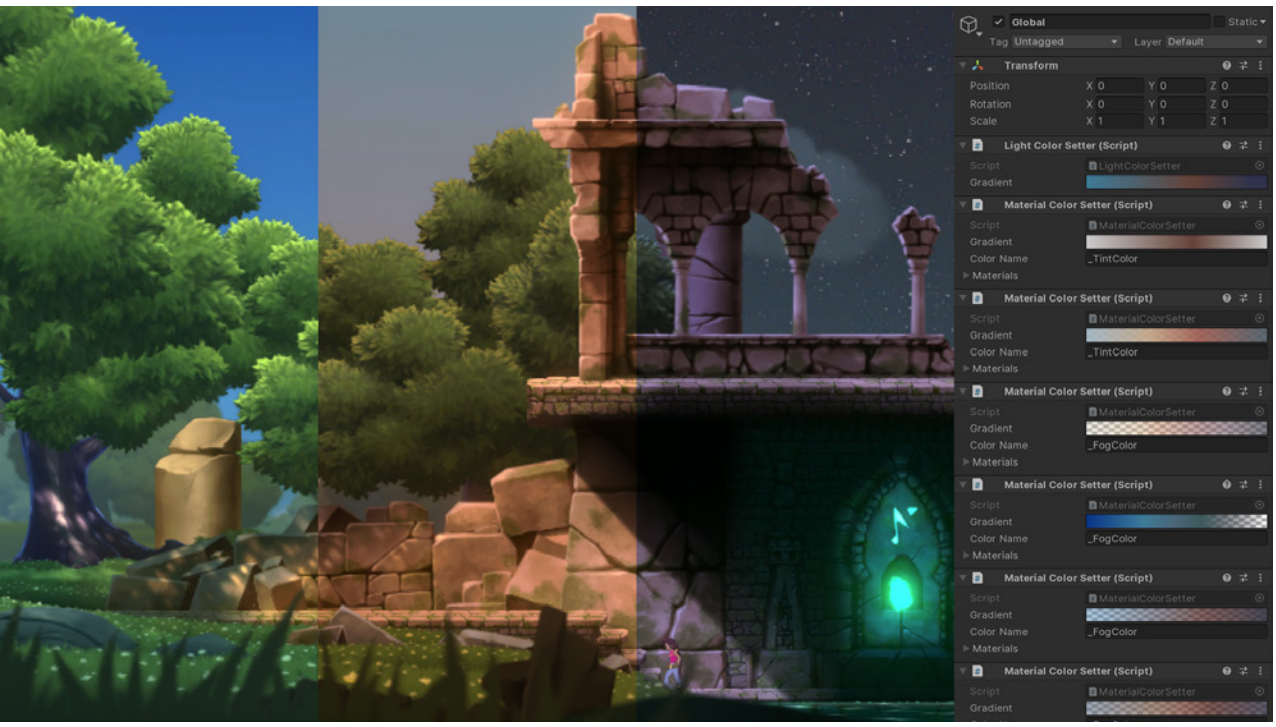
— Richard Rispoli, Creative Director, B2tGame

2D Lights setup (above) and off and on (below). You can use the same Sprites to create different weather conditions or moods.

During Play mode, you'll notice that when the game changes the scene lighting, we manipulate the Global light color, which affects the Sprites in the Scene (you can find it and change its color in the Hierarchy window under Lights>Global), while we change the background Sprite.

A global light affects all sprites in the targeted sorting layers.





A global light affects all sprites in the targeted sorting layers.

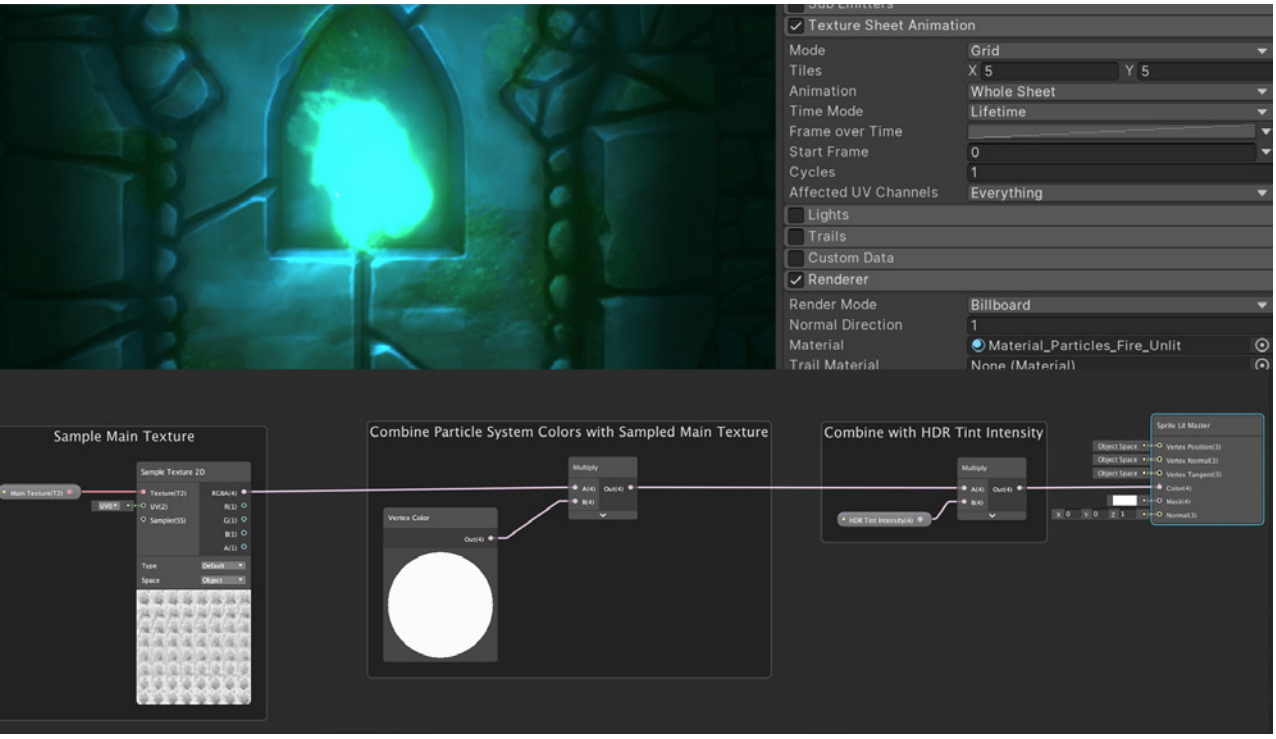
We orchestrated the light changes in the Scene with simple scripts that hold a Color gradient value (light color from daytime to nighttime) and the lights and materials change color following the Time parameter in the parent GameObject. With this kind of setup, you can visually control how different lights blend with precision.

Step 5 —

Adding fire effects

Effects, such as fire, can add a great deal to player experience and create more immersive environments. To add one in Lost Crypt, we started by creating some GameObject torches using the Particle System and Shader Graph for 2D, and then used the Sprite-Lit Master node for the output shader. We made the fire animation in a traditional Sprite sheet that the Particle System uses to play the animation.

The shader we made for the flame utilizes an HDR tint color to increase the intensity of the object's glow using the Volume post-processing. The parent GameObject contains some sparks particles and some lights that illuminate the alcove.



We used an animated 2D shader and particles to create a fire effect.

Step 6 —

Adding water reflection

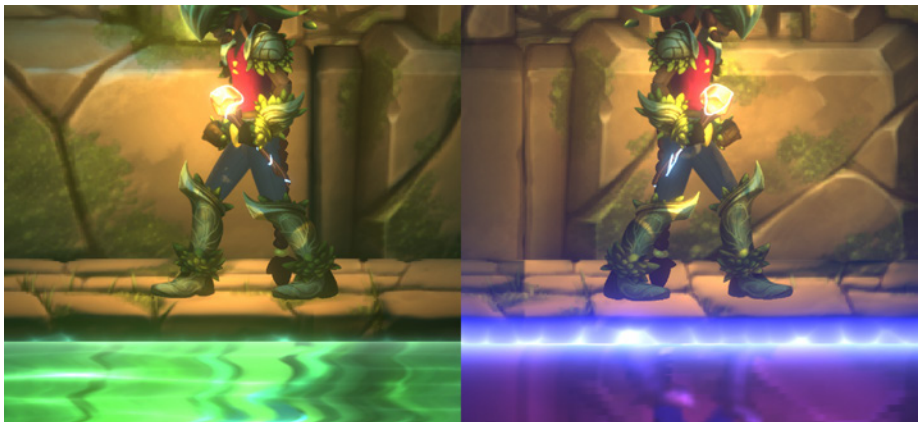
Another everyday use case for shaders is reflections and refractions (e.g., water, ice, mirrors or monitors that display another area of the level) are quite common in games.

Tip!

Using Shader Graph

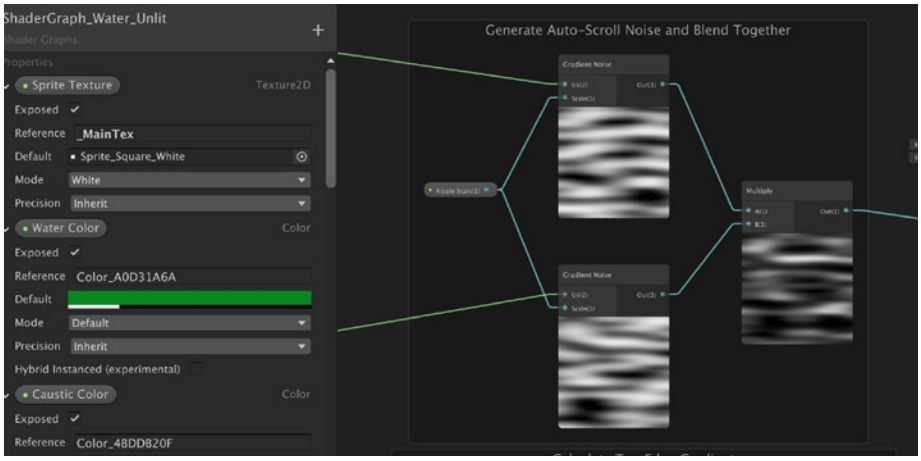
Shader Graph really helps you push the overall quality. If you look at the Night Ghouls, for example, their shape and style is quite neutral – all the magic is in Shader Graph."

— Richard Rispoli, Creative Director, B2tGame



By changing different parameters in the water shader, you can create unique effects.

We achieved the water effect in the crypt entirely with the Shader Graph. We exposed several parameters (like water color, waves speed, and distortion, ripple effect, etc.), which allowed us to adjust the final appearance in the Scene. In order to project a mirrored image of the environment, we added an additional camera that would output the image in a texture to be read by the Shader Graph. Additionally, we added a pass of post-processing bloom to make the water caustics glow, which gives the water surface a nice effect.



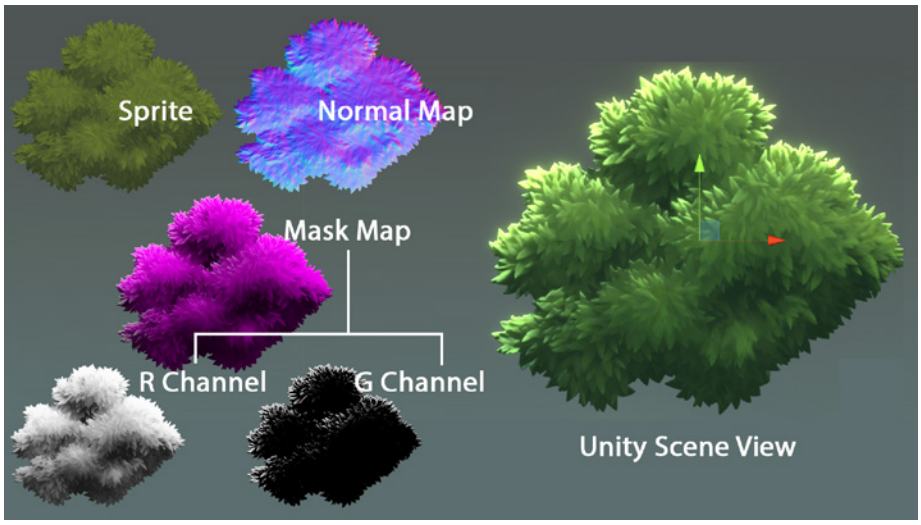
We used groups and clear labels in the graph to make the shader logic easier to follow.

Step 7 —

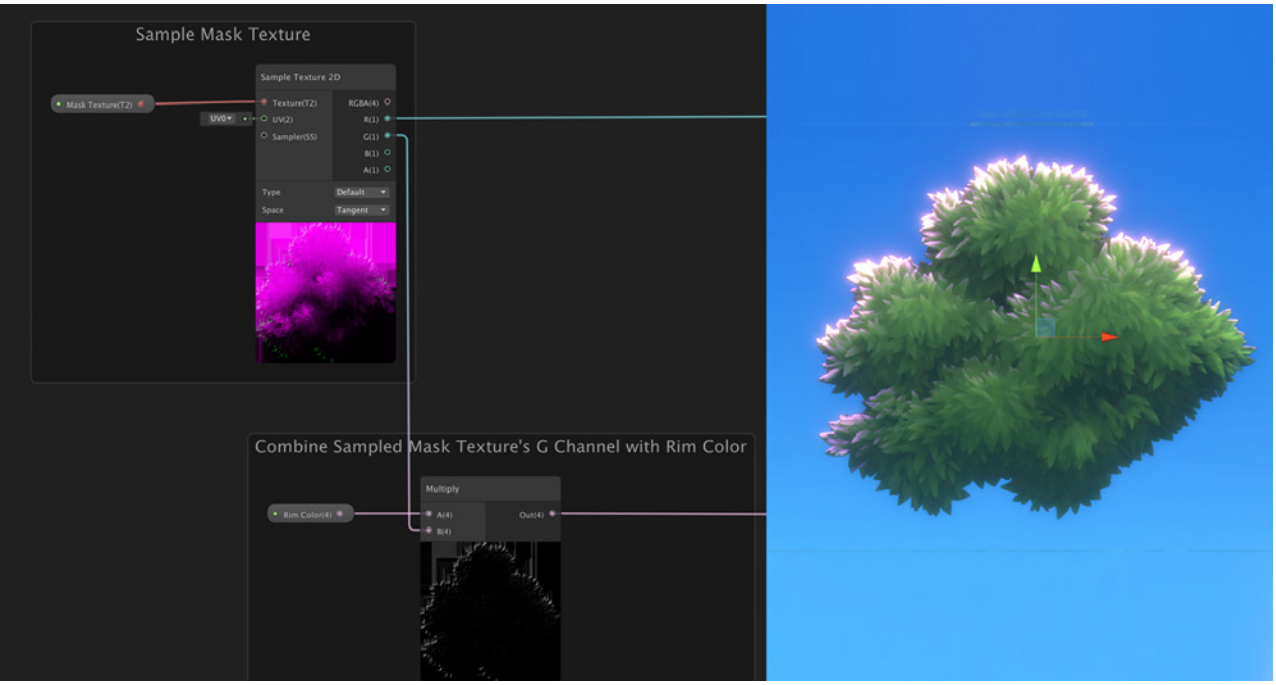
Making trees move in the wind

One way we animated the environment was to make the tree branches sway in the wind. To achieve this effect, we applied just one shader to each tree's foliage Prefab – to create variety and avoid repetition.

Sprite Shape, 2D Lights, and Shader Graph help bring lush foliage and grass to life in Lost Crypt.

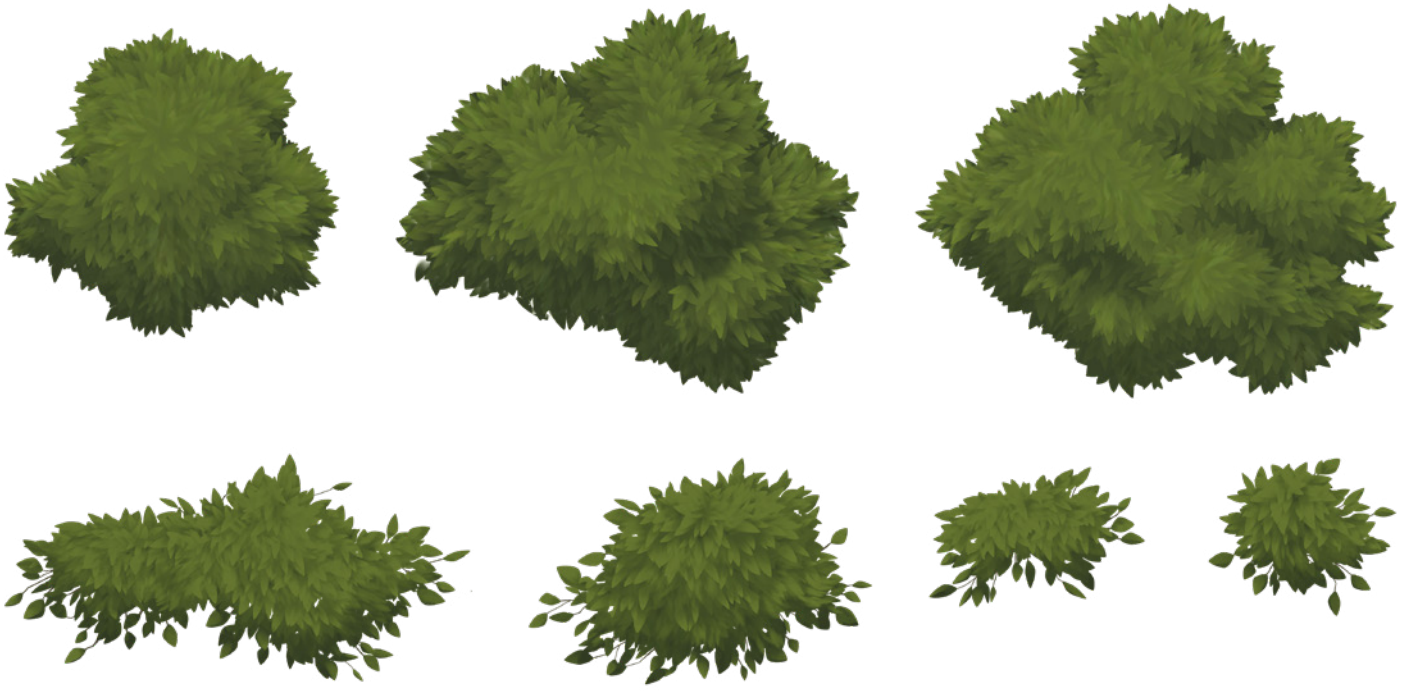


In the Sprite window we added the Normal and Mask maps to be used by Shader Graph.



In the Sprite window we added the Normal and Mask maps to be used by Shader Graph.

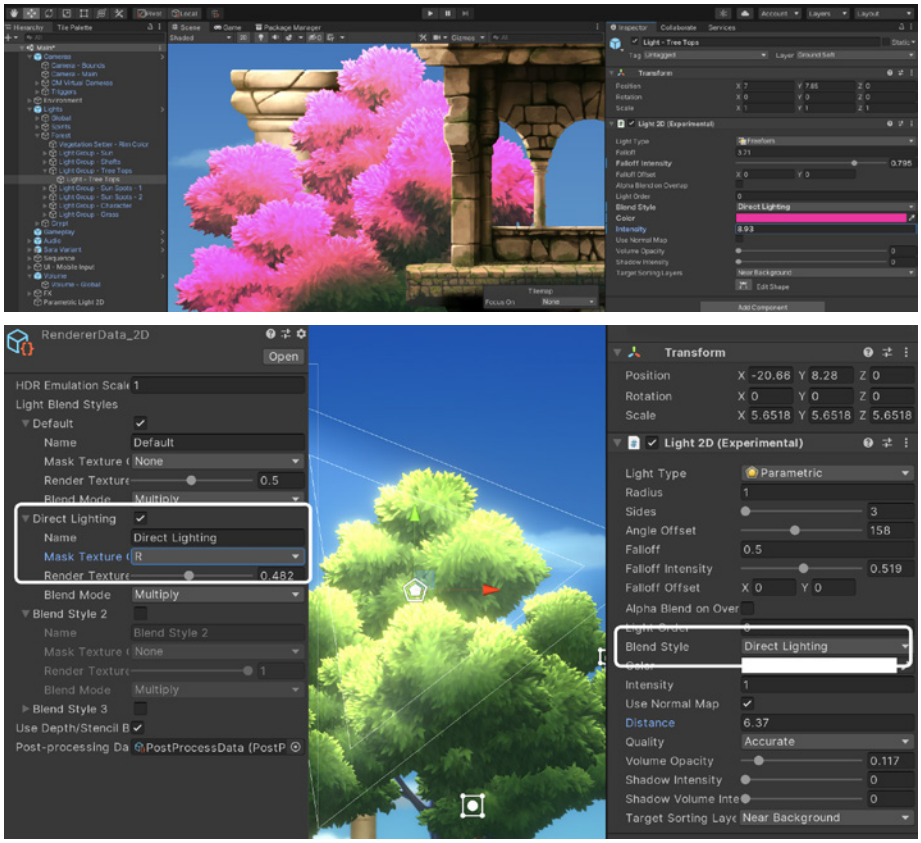
On the Vegetation Wind-Lit graph, you can see how we created two effects. One is the animation or sway effect, which we created by displacing the Vertex position following a few parameters that modify a noise pattern. The second effect uses the G or green channel to adjust the rim light's tone around the foliage.



Step 8 —

Applying blend styles for light effects

Light Blend Styles are a collection of properties in the 2D Renderer that describe how lights should affect Sprites. For example, you can create a blend style that will only affect a particular channel. When you add a light in the Scene that uses that blend style, it will only affect the areas of the Sprite that the Mask map channel information dictates.



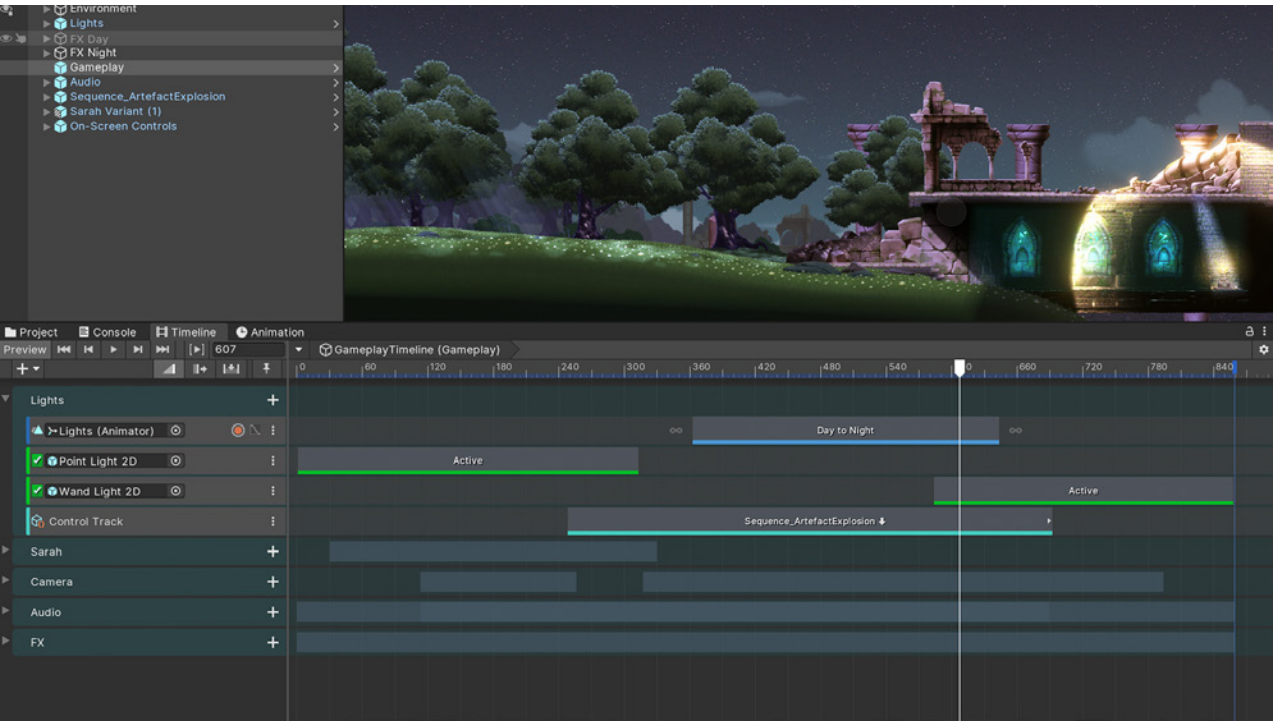
You can see Light Blend Styles in the 2D Renderer (above), and a 2D Light using a blend style (below).

In the example above, the parametric light uses our Direct Lighting blend style, which only affects the areas indicated in the R channel of that Sprite's Mask map. If you find and select the 2D light in the Hierarchy called "Light – Tree Tops," you can change the intensity of the light or color in the Inspector window. Notice how the Mask Map of the tree Sprite prevents the bottom part from being affected by the changes keeping the original green color in that area.

Step 9 —

Creating a cinematic moment and adding particles

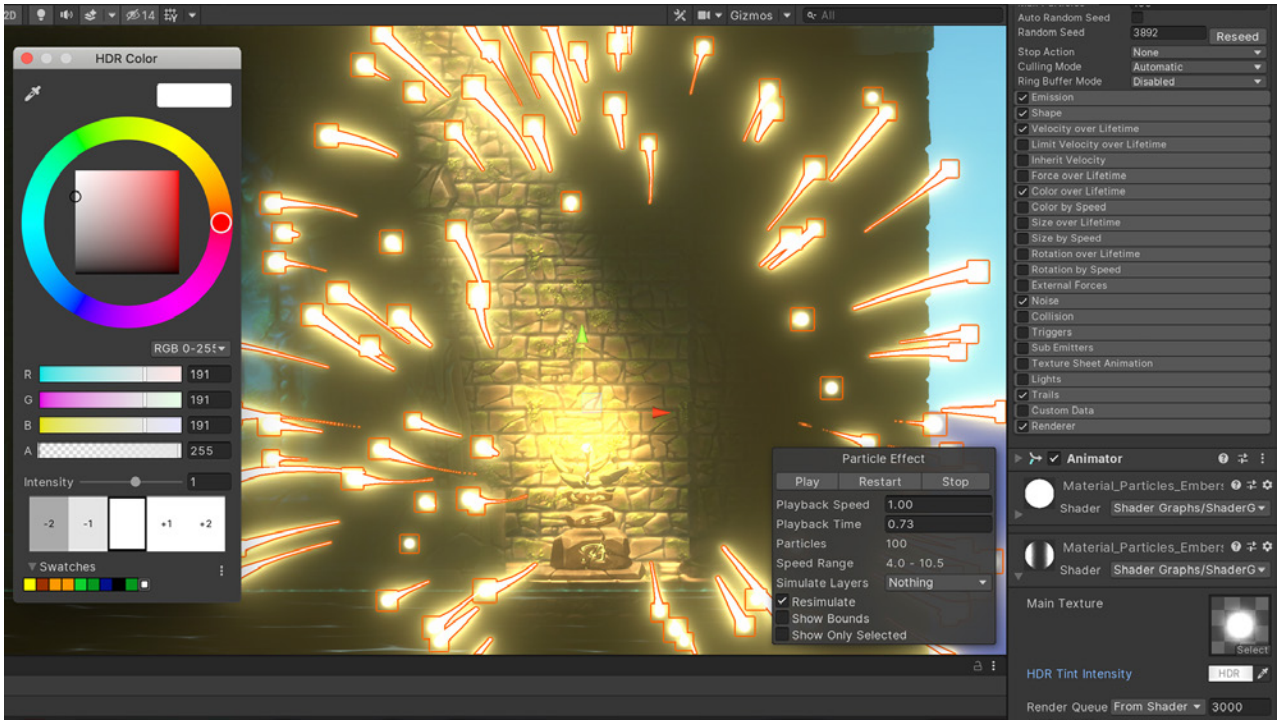
Lost Crypt has a short cinematic when our adventurer grabs the magic wand in the crypt. To make the moment a bit more special, we changed the mood of the environment and spawn particles at the right time with Timeline, since we want to observe the change to nighttime. To follow the particles flying into the woods, we switched Cinemachine cameras by also having an animation track blending cameras.



We used Timeline to create a cinematic moment by coordinating camera, audio, and visual work.

The glowing ring that appears when you grab the wand uses Sprite-type lights. The ring graphic simply expands and fades, creating an aura that lights up the environment.

The particles use the Random Seed feature to always animate in the same way.



We achieved the particle glow effect mostly with the Bloom effect in Volume post-processing. Also, the material/shader for the particles and trail uses an HDR color to define how much intensity the post-processing effect should apply to this object.

Step 10 —

Creating animated ghouls

Look closely at the woods – you can see some spectral creatures in there. To do that, we created a shader that can be used for other ghosts. They are transparent but a fresnel effect adds some shine in the Sprite's edges, making them wobble like floating creatures.

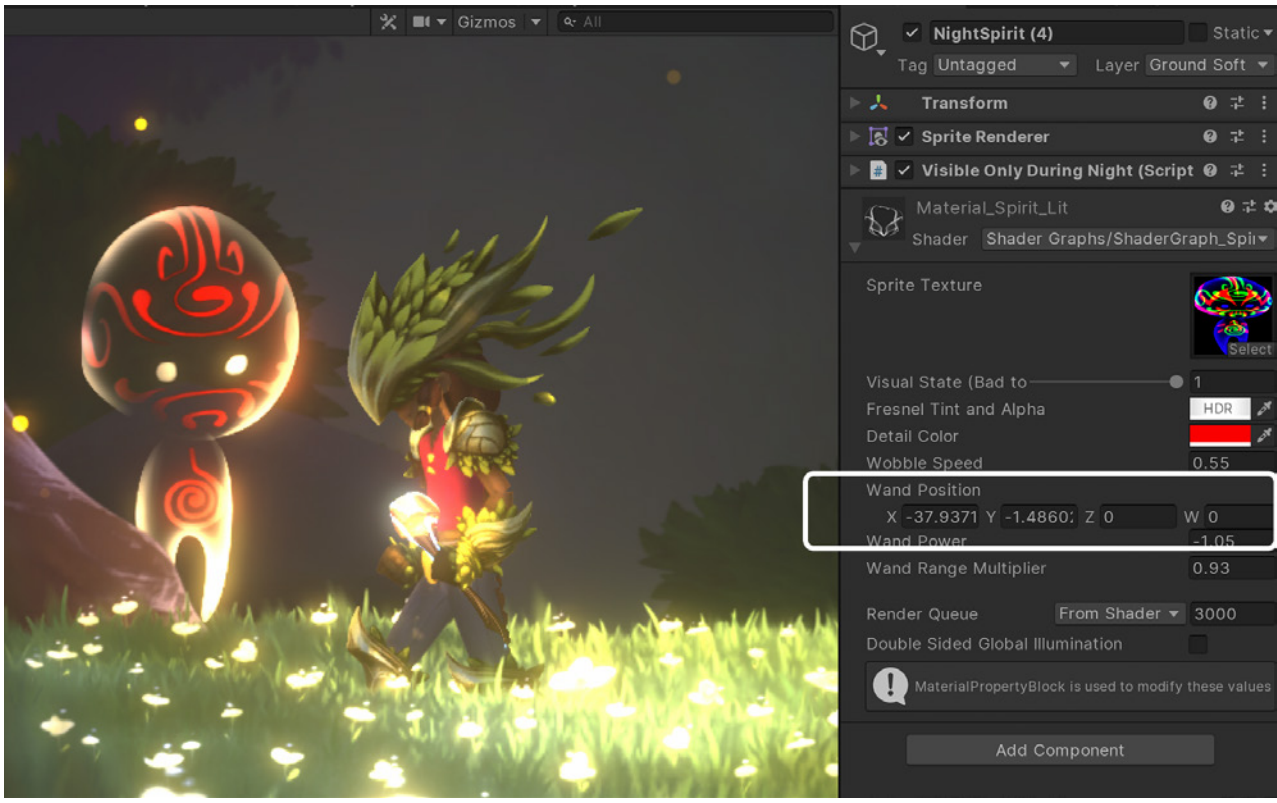
One interesting effect in the shader is the tracking of the wand GameObject transform position. For example, when the wand is close to the ghouls they become brighter. In order to achieve that, we attached a small script to the wand that updates its position in the material shader.

Tip!

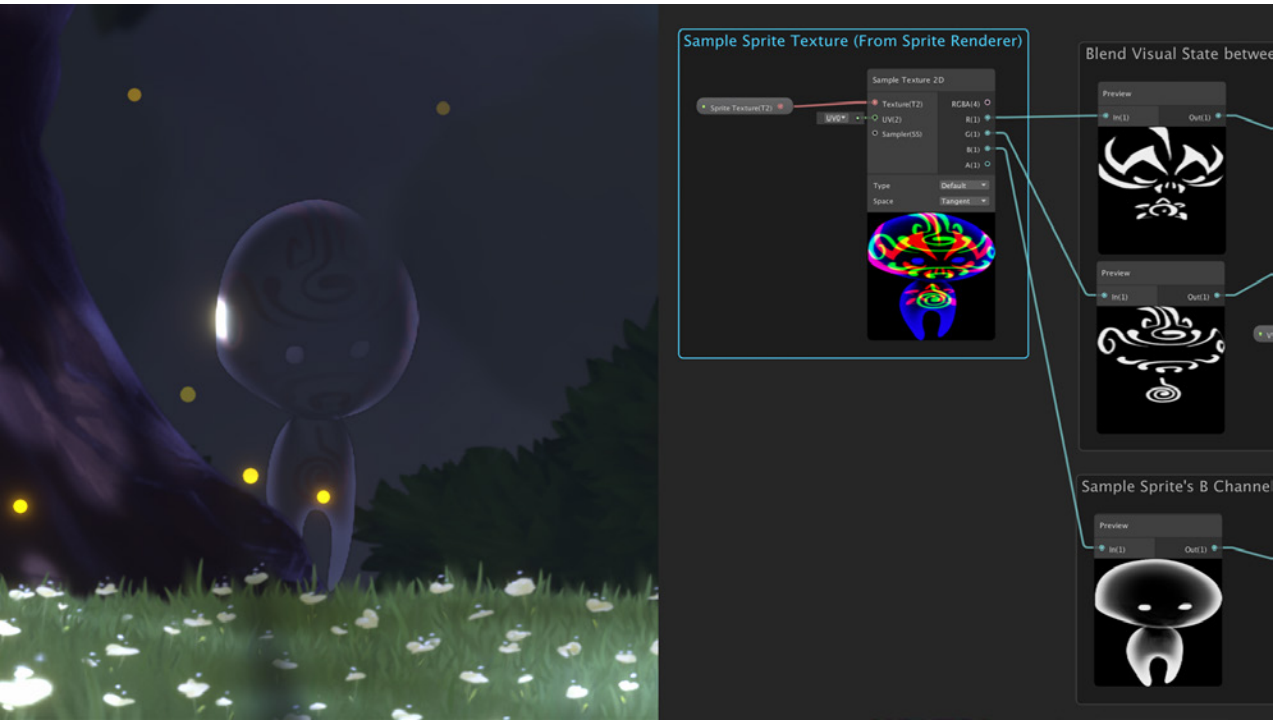
Boosting visual quality with Shader Graph

Shader Graph really helps you push the overall quality. If you look at the Night Ghouls, for example, their shape and style is quite neutral – all the magic is in Shader Graph."

— Richard Rispoli, Creative Director, B2tGame



By exposing values in the shaders, you can adjust visuals based on gameplay mechanics.



Here you can see the ghoul when the wand position is far from it, and the Mask map is on the right.

The ghouls also have a small animation that swaps from one graphic to another. In order to do that, we created a Mask map with different graphics in each channel: R with one visual, G with the alternate visual, and B for the fresnel effect.

Step 11 —

Adding post-processing effects

For a final layer of polish, we added some post-processing effects included in the Universal Render Pipeline. For example, we created an empty GameObject and attached the Volume component to it. In Lost Crypt we use bloom, white balance, and vignette, but there are many other effects that you can use in 2D projects like motion blur and color correction of film grain effects.

You can try other effects by finding "Volume – Global" in the hierarchy and then add them with "Add Override" or adjust the existing effects. Note, however, that post-processing can be demanding on your players' hardware, so try to minimize its use, especially if you're targeting low-end mobile devices.

Tip!

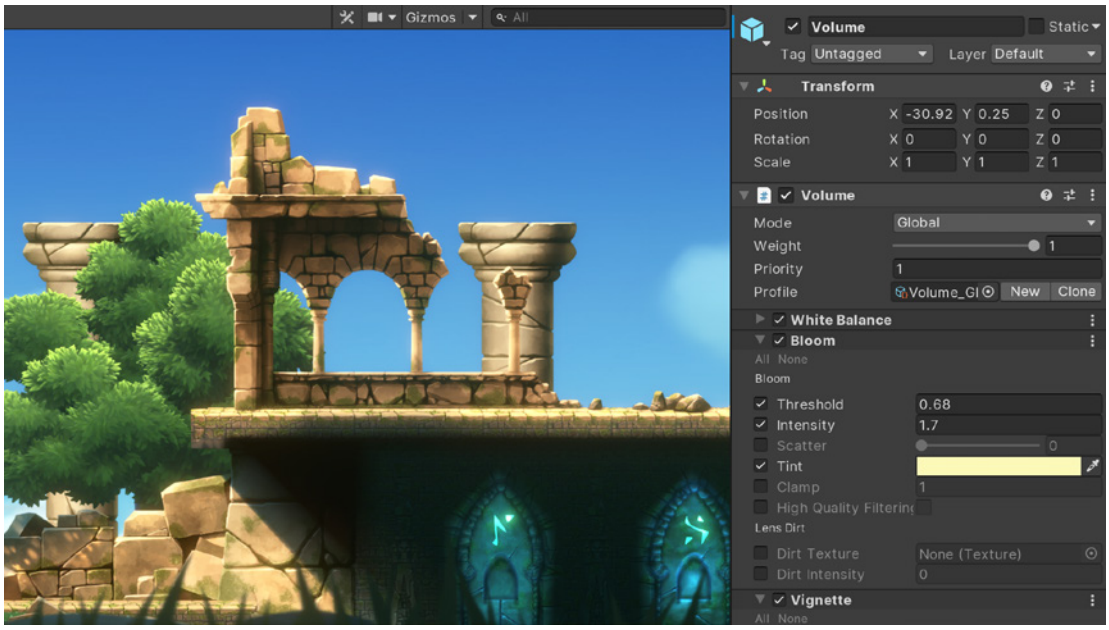
Optimizing art-asset settings for performance

"The biggest performance costs we ran into were due to overly complex puppets, so make sure to reduce the number of Sprite Skins in puppets by having as few elements as possible. Also, lower the geometry in the Sprite Skin Editor as much as you can.

Lights had minimal impact on performance, so you can pretty much use as many as you want, but try to use as few sorting layers as you need.

If you're looking to optimize even further, try reducing the Render Scale of your RenderPipelineAsset and/or the Render Texture Scale of your custom 2D Renderer Data (if you have one)."

— Ivano (Ivan) Mansueto, Technical Director, B2tGame



Here you can see the Volume post-processing settings in Lost Crypt.

Wrap-up

We hope that by seeing how we put together the Lost Crypt demo, you have gained valuable insight into using our integrated suite of 2D tools for creating your own visually stunning games. With them, you and your team can get engaging and beautiful results in areas like animation, light effects, organic terrain, shaders, and post-processing. To recap, use these flexible tools to:

1. Build your world

— 2D Tilemap Editor supports grid, hex, and isometric tilemaps

— 2D Sprite Shape lets you create levels quickly with organic shapes and terrain that dynamically use your Sprites for outline and filling effects

— Sprite Atlas automatically organizes your assets into Sprite sheets to optimize performance

2. Create and animate your characters

— PSD Importer brings your layered Photoshop characters into the Editor for quick and simple character rigging

— 2D Animation provides native 2D skeletal animation in Unity with IK and setup bones for creating character geometry and rigging

— Sprite Swap enables you to change a GameObject's rendered Sprite while keeping the same skeleton rig and animation clips

— 2D Physics lets you create accurate and advanced 2D physics simulations

3. Bring your world to life

— 2D Lights and Shadows helps you create more vibrant environments and characters

— Shader Graph for 2D lets you build 2D shaders visually and see the effects in real-time

— 2D Pixel Perfect enables you to achieve sharp and consistent pixel-based motion and visuals

— Post Processing takes your art to the next level with full-screen filters and effects

If you have any questions about Lost Crypt or our 2D tools, you can reach us on our dedicated forums (see the *Additional resources* section on the next page).

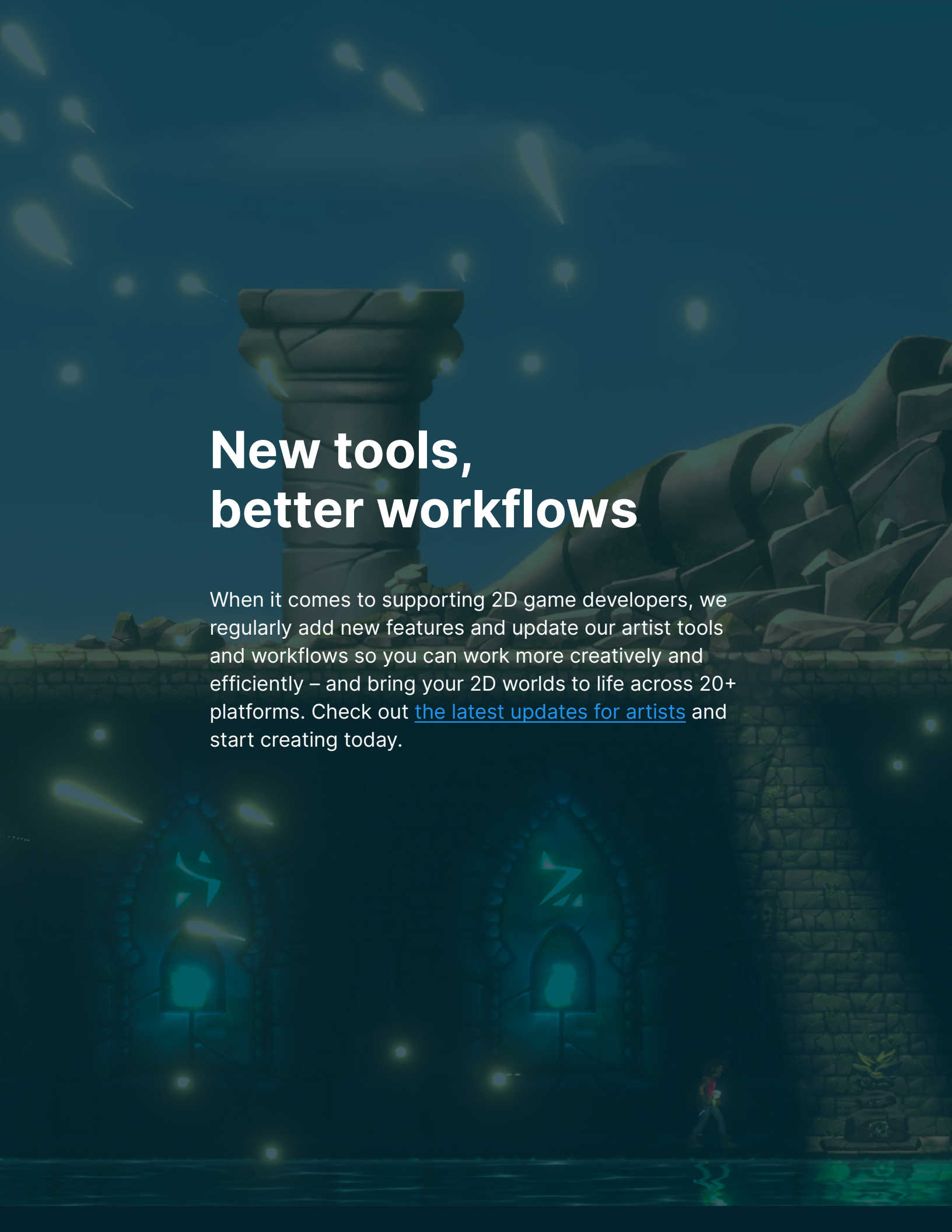
Additional resources

Lost Crypt Forum

2D Forum

New Sprite Rigging and Lighting for 2D – Unity at GDC 2019

Online documentation – Lost Crypt includes a tutorial window to help you navigate through the different features used to build the project.



New tools, better workflows

When it comes to supporting 2D game developers, we regularly add new features and update our artist tools and workflows so you can work more creatively and efficiently – and bring your 2D worlds to life across 20+ platforms. Check out the latest updates for artists and start creating today.


unity