Tef@LiveCraft ver1.0

Overview

Tef@LiveCraft is a Unity editor extension app that allows you to harvest prefabs that can add a little extra to your world. It generates various autonomous butterflies (Tef).

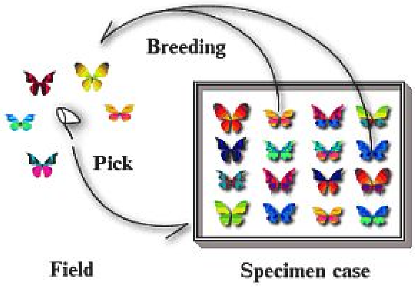
System Requirements

Unity 2021.3.14 or higher
Rendering Pipeline: Built-in, URP

Description

Release LiveCraft's butterflies into your world for a little added flavor. These butterflies fly freely, rest, and flee when people get too close. Additionally, the butterflies can be easily grouped into thousands of flocks. They are also attracted to objects named "Honey," which can be used for small performances.

This editor extension app is designed as an easy-to-enjoy arcade game. First, different-colored and shaped butterflies appear on a stage called "Field." You collect your favorite butterflies to create a collection. The butterflies that appear on the Field are generated by crossing the genes of collected butterflies, so your selection is reflected in this circuit, and gradually the ones that match your preferences will appear.



LiveCraft system 1999 model.

You can generate three types of prefabs from the harvested butterflies:

1. **Static**: A static prefab that doesn't include any scripts.
2. **Holographic**: A prefab that does not include scripts, but can be animated through shaders. It cannot exhibit collision or reaction behaviors, but can be used in metaverse environments where scripts are not allowed.
3. **Live**: A prefab that includes scripts and allows for realistic reactive actions.

Also, at each generation, you can select from three levels of **LOD** (Level of Detail) for each entity.

Demo & Sample

A simple scene is included that contains five prefabs (3 Live + 2 Holographic). When you play the scene, 1,200 butterflies will appear. Depending on your rendering pipeline, open the **sceneField** from the **Built-in** or **URP** folder in the **Sample** folder and play or build it on the editor.

Package Contents

```
Tef_LiveCraft
 |-Documentation
 |-Editor
 |-Resources ... Folder for storing harvested materials.
 |   |-Materials
 |   |-Prefabs ... This is where the prefabs are located.
 |   |-Shaders
 |-Runtime
 |-Sample ... Demos and samples.
     |-Built-in
     |-URP
```

Important Notes

1. If you're using the **Built-in** rendering pipeline, make sure that the **Mathematics** and **Shader Graph** packages are installed. To check, go to **Window > Package Manager** and click on the **In Project** tab. If you don't see Mathematics and Shader Graph, install them from the **Unity Registry** tab.

2. In platforms like the **metaverse**, script usage may be restricted. However, even in such cases, prefabs generated in the **Holographic** type can still be used for animation using shaders. However, they are unable to perform collision detection or exhibit reactive behaviors.

Usage

- Be sure to back up your project before running it.
- Import this product package into the Unity editor.

▶ How to Run the Demo

1. Open **Asset/Tef_LiveCraft/Sample** from the Project window, and double-click on the **sceneField** from the **Built-in** or **URP** folder depending on your rendering pipeline.

2. Once the demo scene is displayed, press the play button.

▶ How to use the app

1. Select **Tools** > **LiveCraft** > **Tef** from Unity's Top Menu to open the app window. There are descriptions provided throughout the app, so refer to them when necessary.

2. Each property has default values, so it will work even without resetting them. Here are the minimal steps to get started:
3. In the **Field** tab, click on the butterfly you like from the ones displayed, collect it, and repeat by clicking the **Refresh** button to request new designs. Over time, designs closer to those you have collected will appear. It will be easy to see if you collect ones of the same color.
4. Choose one design from those collected in the **Collection** tab.
5. Check the animation in the **Anime** tab, and change the settings if necessary.
6. Output as a prefab in the **Harvest** tab. Select **Live** or **Holographic** as the prefab type and make sure **Put in scene** is checked. Click **Create Prefab** and specify a name in the file dialog box, then save it.
7. Now, the butterfly prefab will be added to the current scene. At this point, please close the window of this application. If you open the object named **Lc_Tef** from the Hierarchy, you will find the prefab with the specified name inside. Select that prefab and ensure that its Position is where it should appear in the camera's view. Then, try playing the scene. Butterflies will take flight in your world.

Our Website

Visit site

Support

Contact Form

▶ Detailed Usage Instructions

1. You can adjust the display of each control in the bottom status bar to make it easier to see. Please make changes as appropriate.

2. In the Settings tab, you can select the **LOD**(Level of Detail) for the prefab. High (0) provides the highest level of detail, while Low (2) results in a more simplified representation. Increasing the level of detail incurs additional memory and processing costs.

3. The **Range(X,Y,Z)** in the Anime tab's status bar represents the butterfly's movement range (± meters). By turning on the **Range** button, you can display the movement range on the scene and confirm it. The **view magnification** is for making small butterflies easier to see and does not affect the actual size of the generated object.

4. In the **Live** type prefab, you can attract butterflies to an object named **Honey** or an object with the tag Honey placed in the scene. You can try it out in the demo. You can see a sample script in Player.cs

5. If you want to delete a generated prefab, using **Delete Prefab** in the Harvest tab is convenient. You can delete multiple related files such as materials linked to the prefab at once.

6. For prefabs generated in the **Live** type, you can modify several settings later in the Inspector. (For prefabs in the **Holographic** type, please do not modify them in the Inspector; instead, regenerate them using this application.)

   - **Instances**: The number of butterflies. Maximum 1000 per prefab.

- **Visible Distance**: If the distance from the player is greater than this (In meters), it will become invisible and the instance will be destroyed.
- **Shadow Distance**: Shadows will not be drawn if they are greater than this.
- **Reaction Distance**: If the distance is closer than this, the butterfly will fly away.
- **Moving Range**: It flies around freely within this range. This range is displayed in the Gizmo when the prefab is selected in the Hierarchy.

7. If you want to control the butterflies with your own script, you can change the angle of the wings using the **WingAngle** property of the wing material in your script. Please refer to Tef.cs and AnimeTef.cs in Runtime > Script.

▶ Troubleshooting

1. If you cannot control the demo with the keyboard, please set the **active input handling** to "**Input Manager (Old)**" or "**Both**" in the project settings.

2. If the screen turns **pink**, it may be due to the rendering pipeline not being compatible. This product supports **Built-in** or **URP**. Demos are prepared for each rendering pipeline.

3. If the harvested prefab exists in the hierarchy but the butterfly **cannot be found** in the scene, please make sure it is placed in a position visible to the camera. It may also be too small, so temporarily increasing the scale of the prefab in the inspector may help locate it.

4. If you change the scale of the **Holographic** type prefab in the Inspector, there is a possibility that it may suddenly become **invisible** depending on the position of the camera. Please set the scale again using this application and regenerate the prefab.

5. If it doesn't work properly, please show the console by going to Window > General > Console and check if any errors are displayed. If you think that the error is causing the issue, please fill out the form below with as much detail as possible, including the error message, Unity version, machine used, O.S., target O.S., etc. It would be very helpful.

▶ Self-introduction

I have been developing art-related applications such as VirtualPainter, LiveCraft, and MusicTapestry. My first work was presented in 1987, which was a long time ago, but my experience with Unity is still relatively new and this is my first 3D software. I feel excited about the potential of projecting dreams into the metaverse and VR, and I believe that more and more people will be attracted to this charm in the future. However, currently, game engines and 3D modelers that professional creators use are very difficult to handle, and it is not easy for the general public to enter the field. I thought that there is a need for more user-friendly applications. Although I may have some shortcomings, I will work to improve them one by one, so please let me know if you notice anything. Also, please feel free to contact me for any development projects. Please refer to the following URL for my career. Thank you very much.

profile

Contact Form