

Purring Engine Editor Guide

Module Code: CSD2451/UXG2450

Team: Pawsitive

Engine Name: Purring Engine

Welcome to the Purring Engine Editor Guide. This document provides an overview of the editor.

Contents

Contents.....	1
Editor Controls.....	3
Toolbar.....	3
Folder Structure	4
Usage Guide.....	5
First Startup Notes	5
Object Hierarchy Window	6
Component window	10
Scene Window	11
Resource window	13
Scenes	16
Animation Editor.....	17
Animation Events.....	20
Layer Settings Window	21
General Guides	23
About Entities	30
Render Order	30
Enabling/Disabling Entities	30
Prefab Editor	32
Canvas Objects.....	33
Creating Canvas Objects	33
Text Objects	34
Creating Text Objects.....	34
Editing Text Objects.....	34

Particle Emitter	37
------------------------	----

Editor Controls

The editor has a few controls, as listed below.

Control	Description
F1	Toggles console window.
F2	Toggles object hierarchy window.
F3	Toggles debug log window.
F4	Toggles scene view window.
F5	Toggles resource window.
F6	Toggles performance view window.
F7	Toggles property editor window.
F8	Toggles physics window.
F9	Toggles animation editor window.
F10	Toggles debug drawing.
Left shift + F10	Toggles layer settings window.

Toolbar

The toolbar is located at the top of the entire editor window and has the following functions:

Function	Description
File	Allows for saving and loading of scene files.
Window	Toggle which windows are visible or reset default layout.
Theme	Switch between different editor themes.
Settings	Toggles the layer settings window.

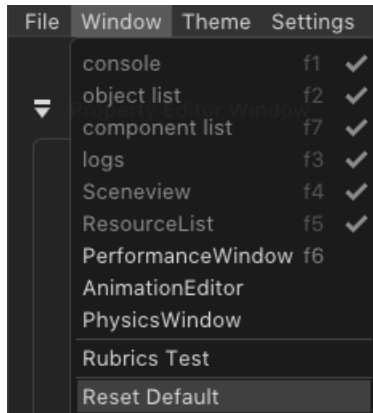
Folder Structure

- Application
 - Contains the engine executable
- Assets
 - **Animation**
 - Contains animation files
 - **Audio**
 - Contains audio files
 - **Defaults**
 - Contains default files used by editor
 - **Fonts**
 - Contains fonts
 - **Icons**
 - Contains icons used by Resource window
 - **Scenes**
 - Contains scene files
 - **Prefabs**
 - Contains entity prefabs
 - **Settings**
 - collidercfg
 - Broad phase collision settings
 - config
 - Editor config settings
 - **Textures**
 - Various textures for use in the engine
- Purring_Engine
 - Contains the Engine as a static library (.lib) for the application to run.
- Shaders
 - Contains shader files for the engine
- Vendor
 - Contains various external libraries & includes

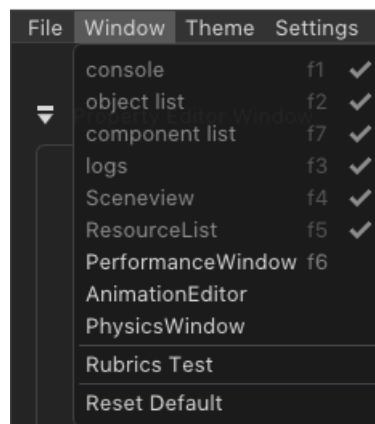
Usage Guide

First Startup Notes

In the event the window does not have anything displayed (first launch), under the **Window** drop down menu, select the **Reset Default** option. This will reset the editor to its default settings.

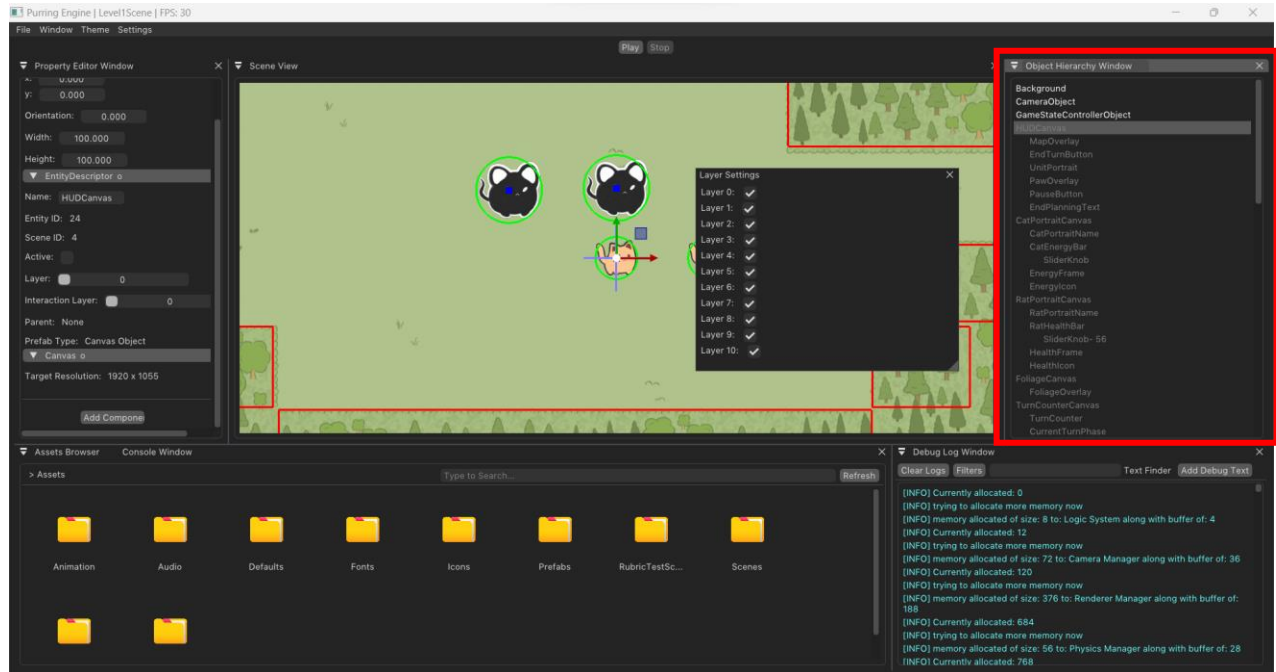


To open new windows, the user can open them from the **Window** drop down menu, select from the list. Or press the Fn keys to open them up. (In this case F1-F6)

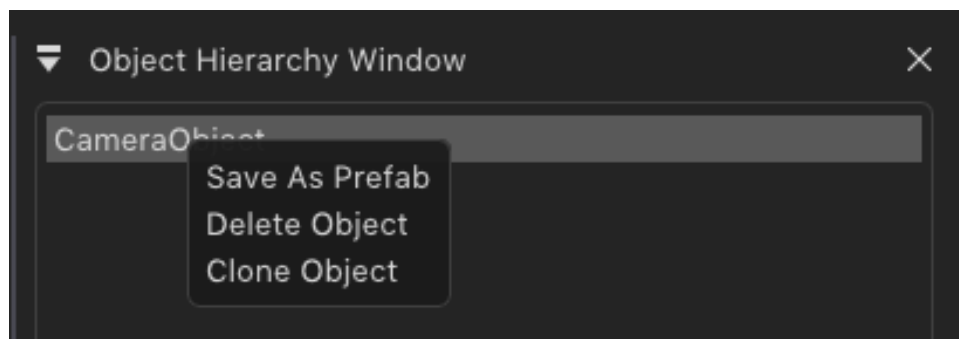


Object Hierarchy Window

This window displays the scene hierarchy and allows the user to select and edit Entities.



Right clicking an entity will select it. The user can then save the entity as a prefab, delete it, or clone it.

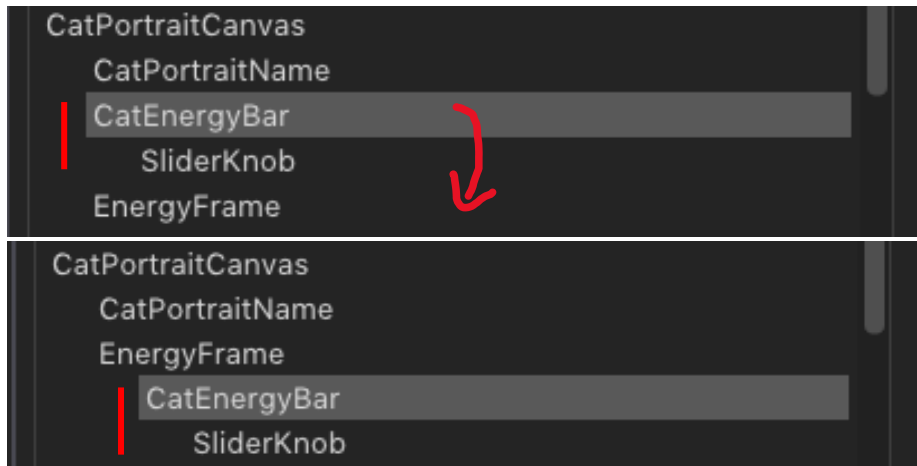


Left clicking and dragging an entity on top of another will make that object a child of the entity.

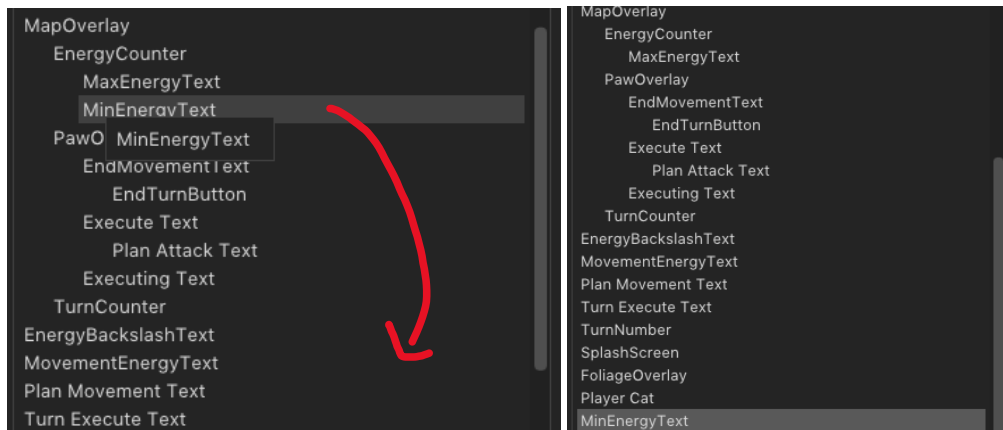


In this manner, children of entities can also be parents of other children.

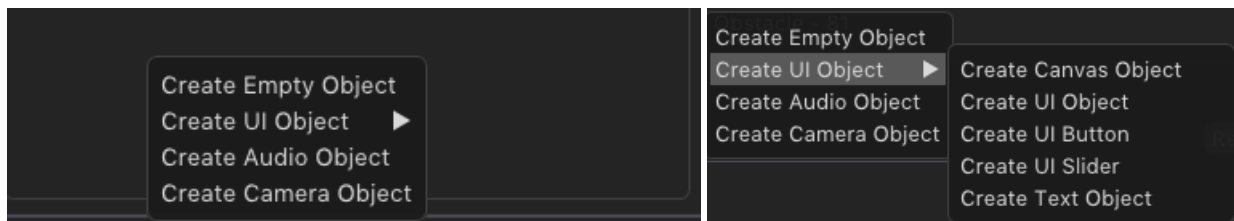
Dragging entities will bring their children along. So if entities with children are dragged in/out of their parents, their children will also follow.



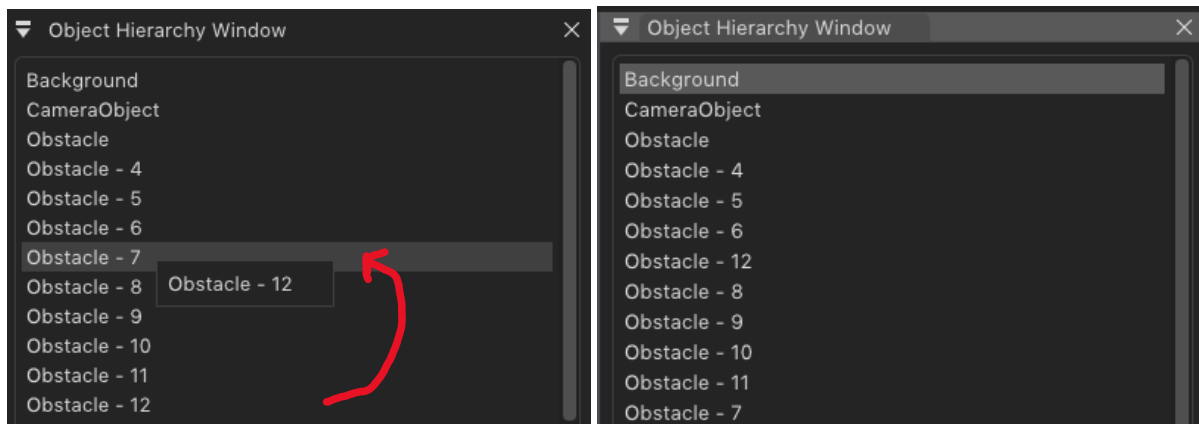
To un-parent the child object, simply drag it out of the parent.



Right-clicking empty space in the hierarchy will open a popup, allowing the user to create an empty object, UI object, Audio Object or Camera Object.



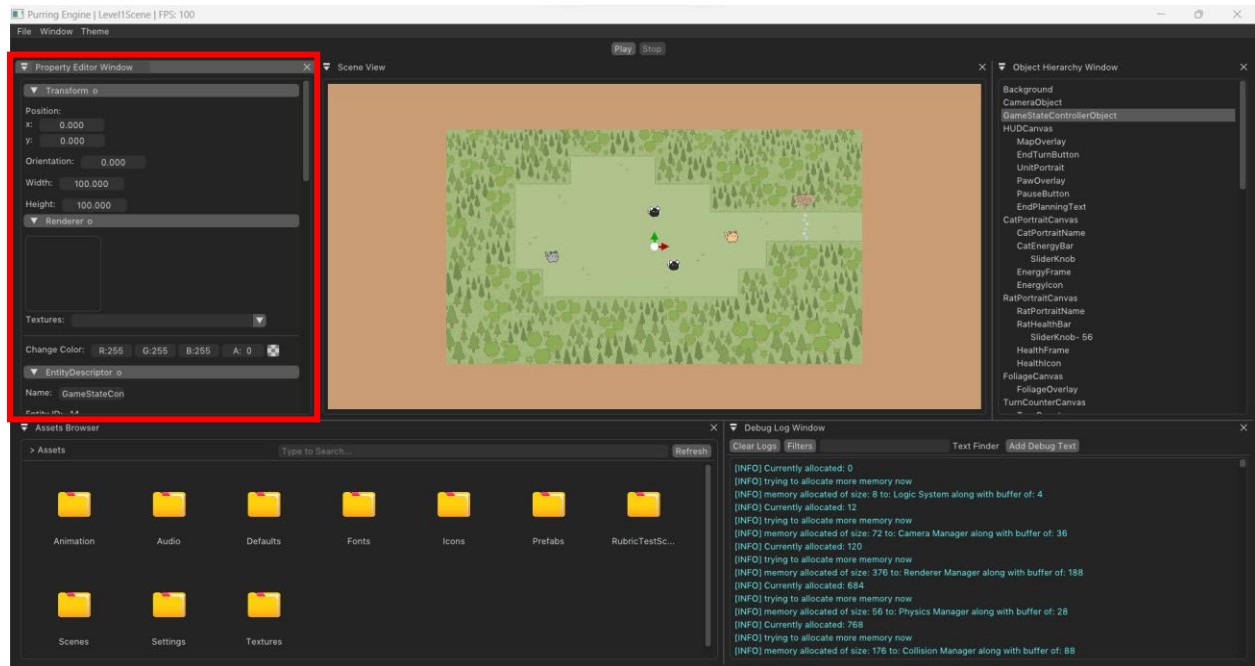
By holding left control when dragging an entity and letting go of left mouse button at the desired location, the order of the objects can be modified.



This is to affect the render order of the entities, more information can be found under the Entities section.

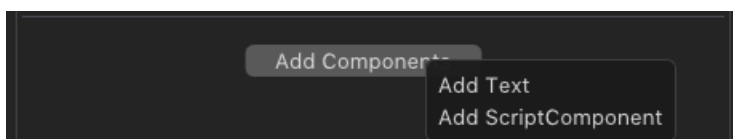
Component window

The selected entity will have its components reflected/displayed on the extreme left, in the **Component Window**. Here the user can add new components and modify existing ones.

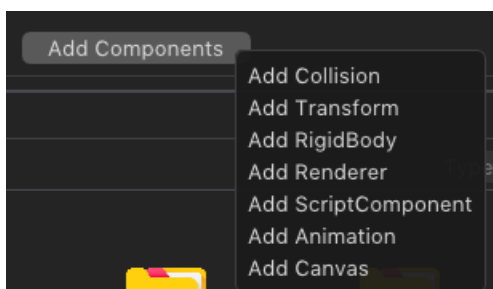


All objects will have Transform and Entity Descriptor components by default. Depending on the type of object it is, different components can be added to the object.

For example, UI objects have a GUI Renderer by default, and only allow for Text and ScriptComponents to be added to them.



Whereas other objects allow for many different components to be added, such as Collision, Transform, Rigidbody, Renderer, ScriptComponent, Animation, Canvas etc.



Scene Window

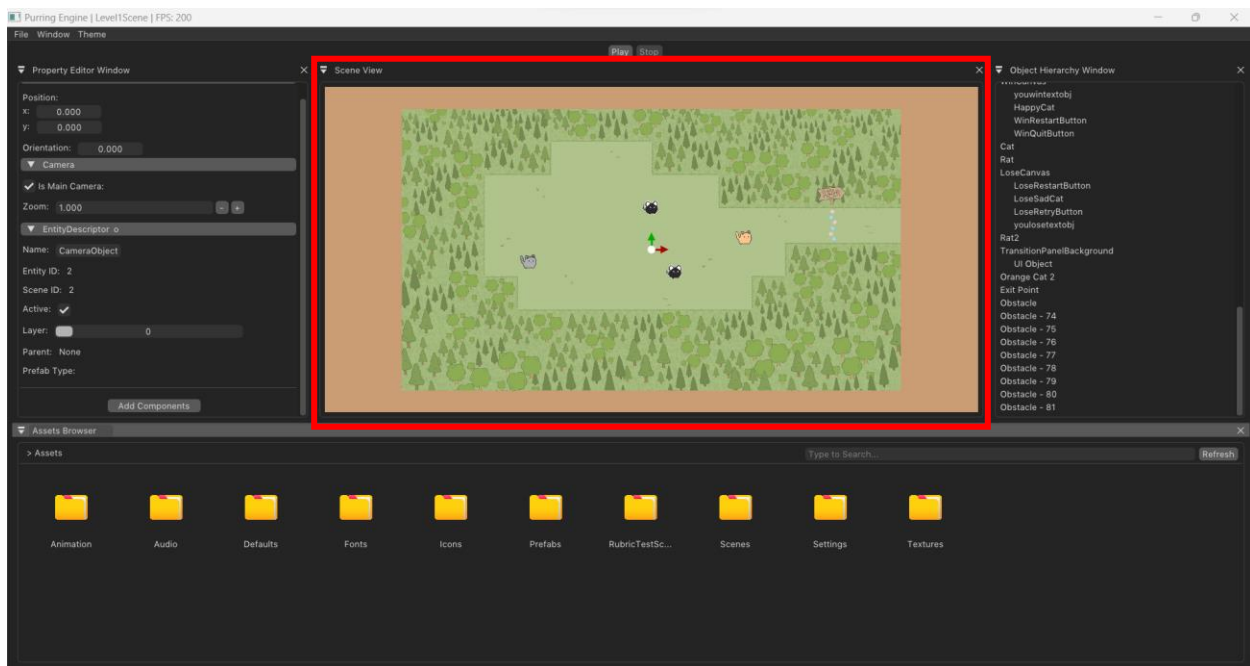
Entities can be seen (if they have a renderer) inside the **Scene View** window.

Gizmos will be displayed upon clicking on an object in the scene view. The user can use the gizmos to modify the entity's position, rotation, and size, which can be toggled by using the below keys.

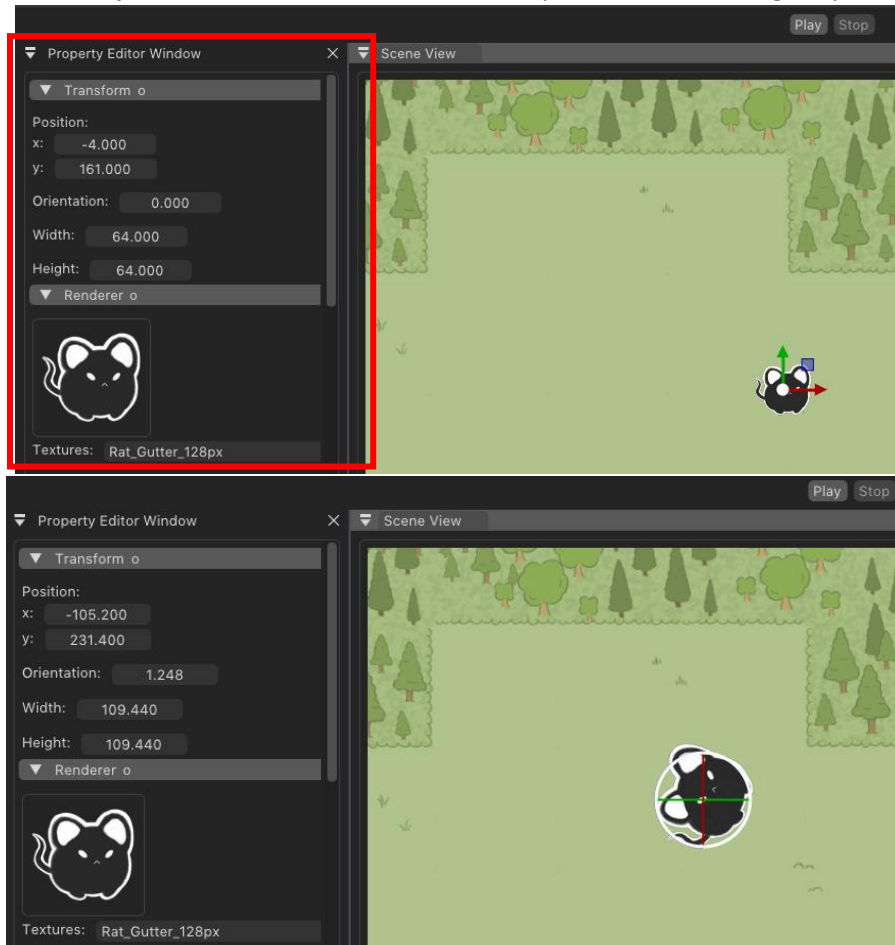
Key	Description
S	Toggles the gizmos to scale the object.
T	Toggles the gizmos to position the object.
R	Toggles the gizmos to rotate the object.

Holding X and Y while pressing the 3 buttons will make the changes aligned to the respective axes only.

Note that the user has to click into the scene view first before the keys can be used.



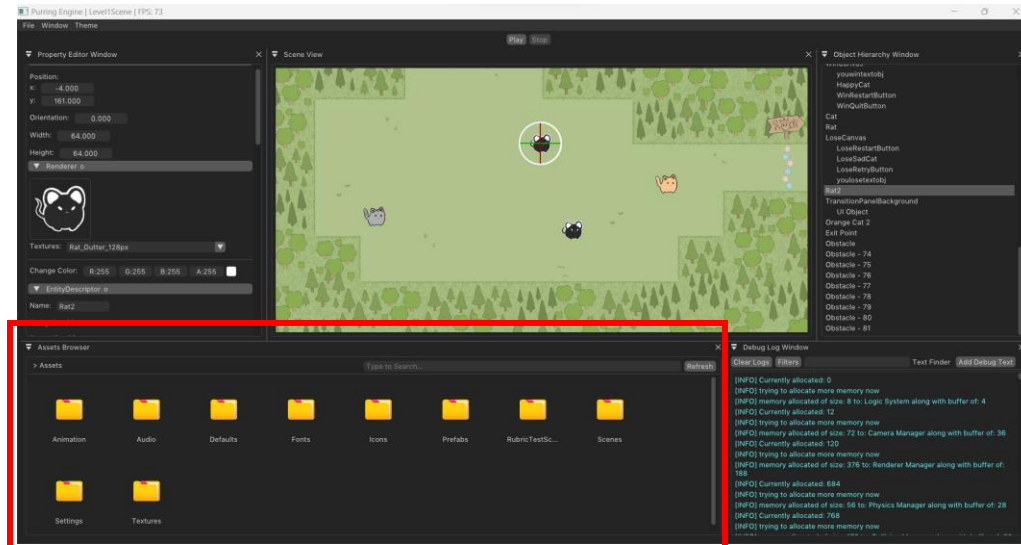
As the object is modified, the Transform component values also get updated accordingly.



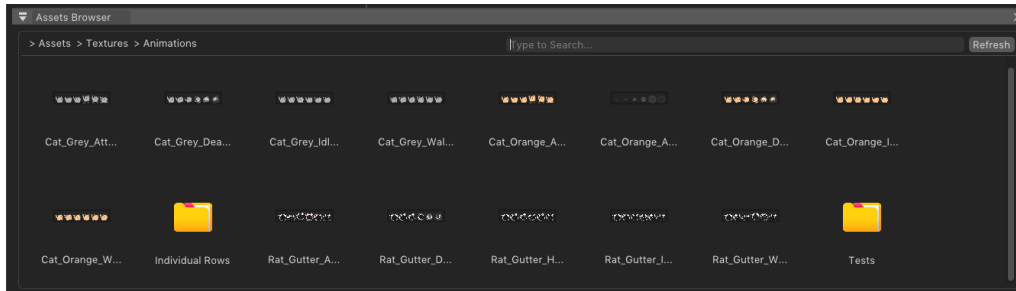
Resource window

The Resource window (Assets browser) contains the resources the user can add to the scene.

Note that it may not be open by default, press F5 to open it.

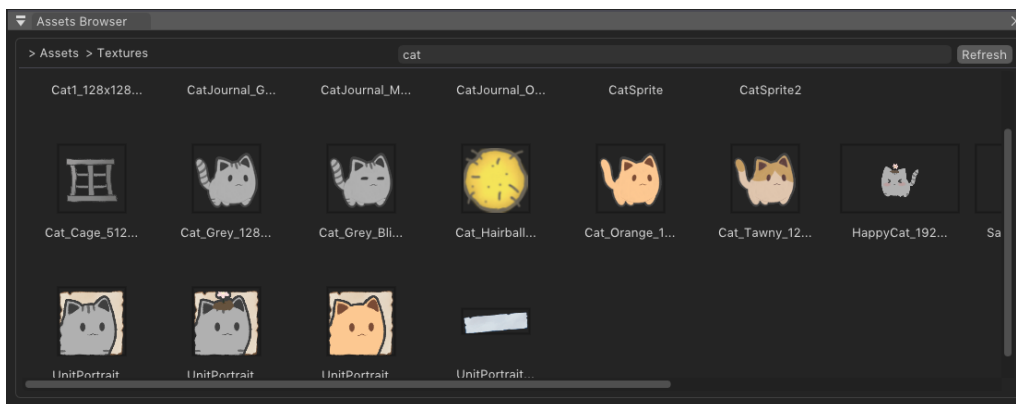


Double clicking on any folder shown will navigate into the folder and show the files within.

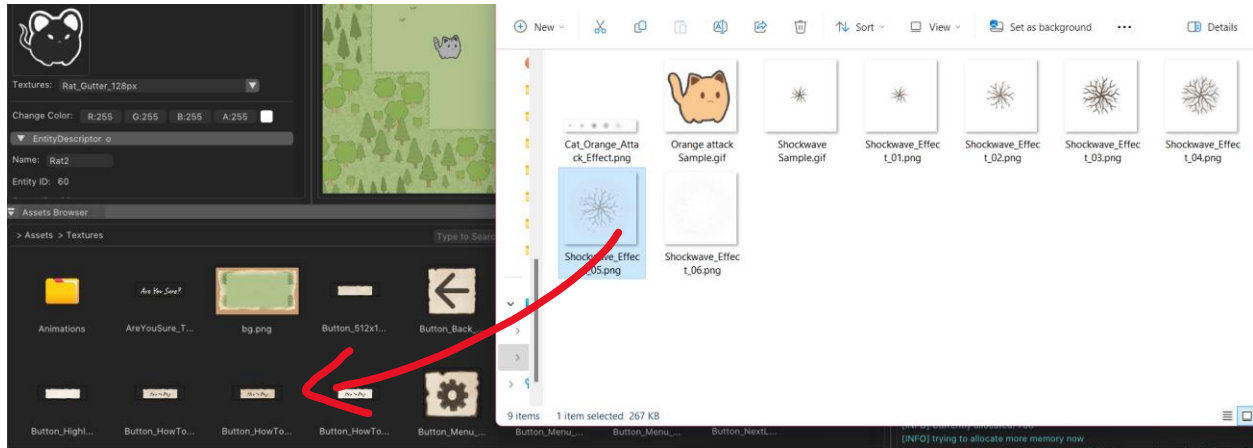


The user can filter files using keywords by typing in the search bar.

The search only returns files within the specific folder selected, and only works with files, not folders.

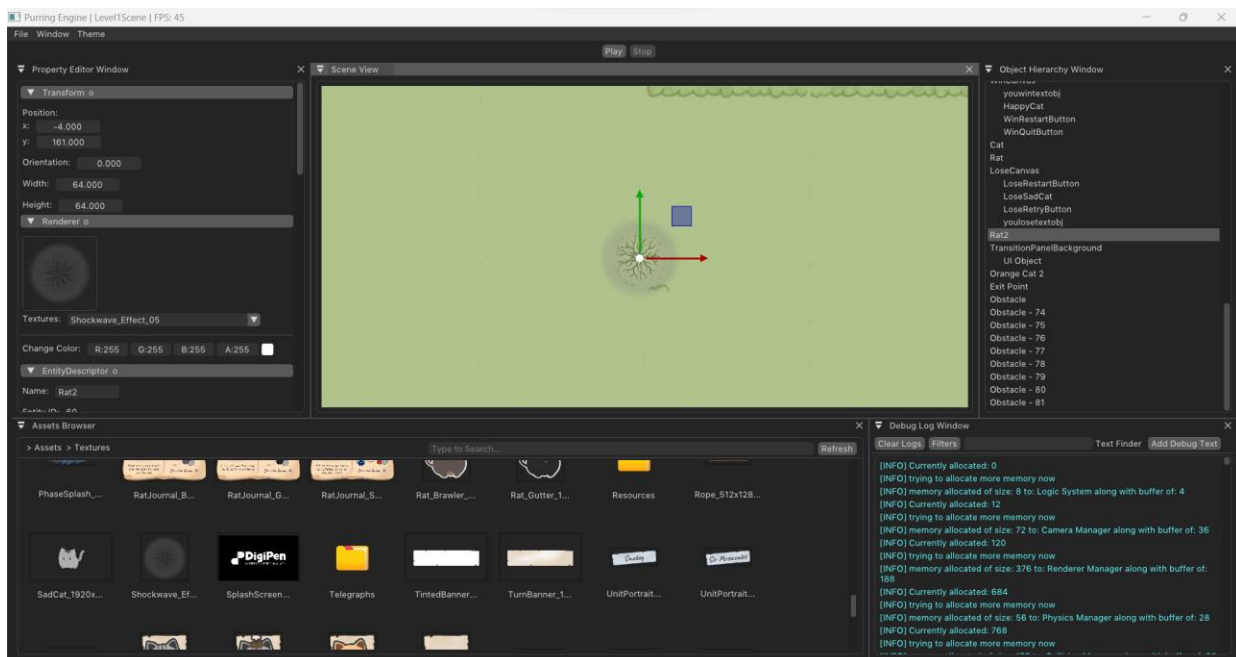


Files can be added by dragging from an external windows explorer into the resource window.

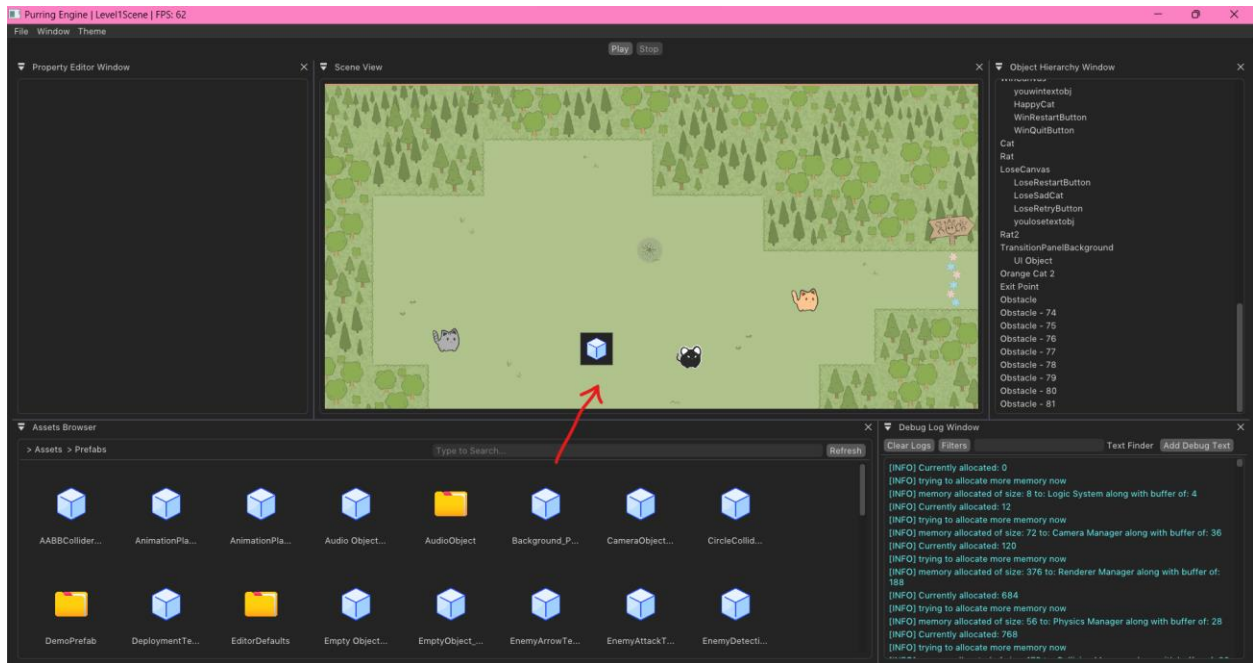


The new file (if supported) can now be used in the project. In the above example, a texture .png file has been added.

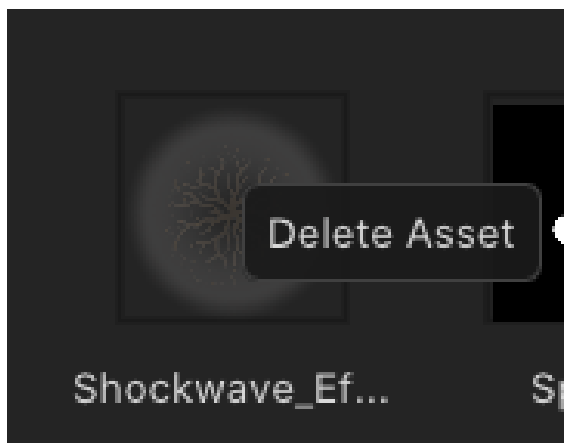
Textures can also be dragged into the renderer component to change the texture to the dragged texture.



The user can drag prefabs (found in the prefabs folder) into the scene to add them to the scene.

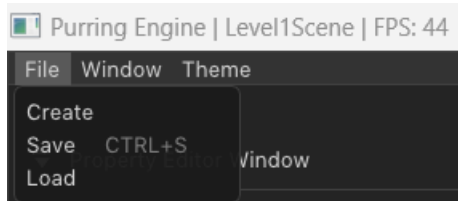


To modify any prefab file or delete an asset from the asset browser, right click on the asset and select the desired option.



Scenes

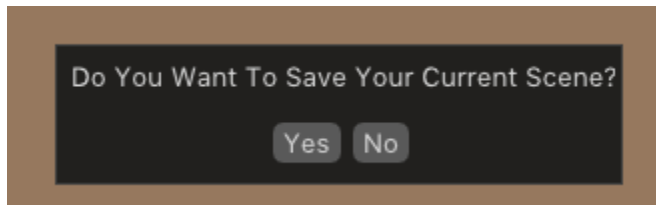
Editor Scene Create/Save/Load



The current scene name is displayed on the top left corner of the window.

Create

A popup to save the current scene will appear when Create is clicked.



Pressing yes will save the current scene. If the current scene is the default scene, it will open the file explorer to save a new scene. After the scene is saved, the default scene is loaded.

Pressing no will not save the current scene and load the default scene.

Save

Clicking this will open up the device's explorer window and the user can select a location to save the scene.

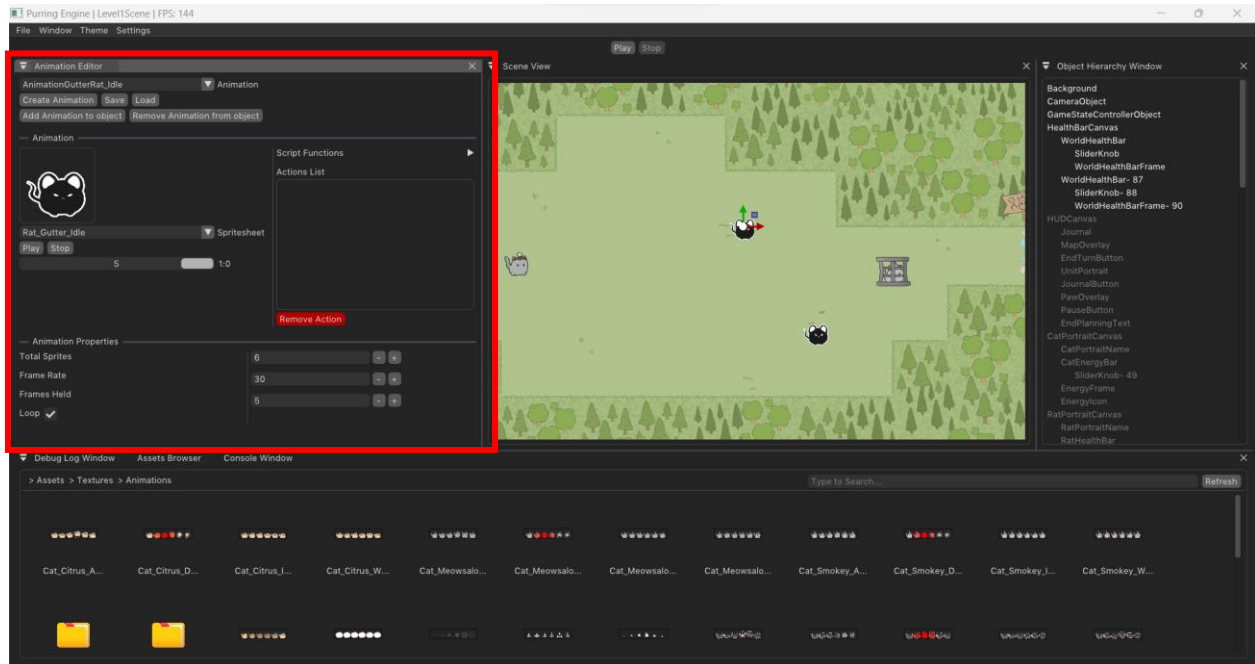
Load

Clicking this will open up the device's explorer window and the user can select an existing scene to load.

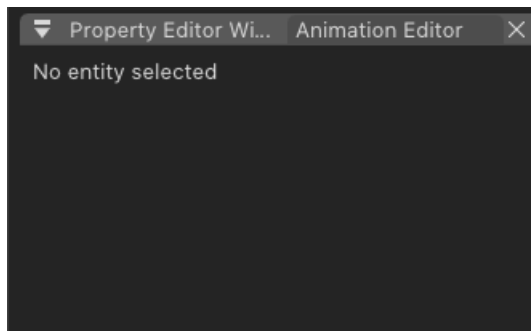
Animation Editor

The Animation Editor window contains the resources the user can add to the scene.

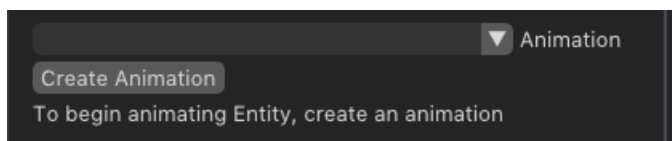
Note that it may not be open by default, press F9 to open it.



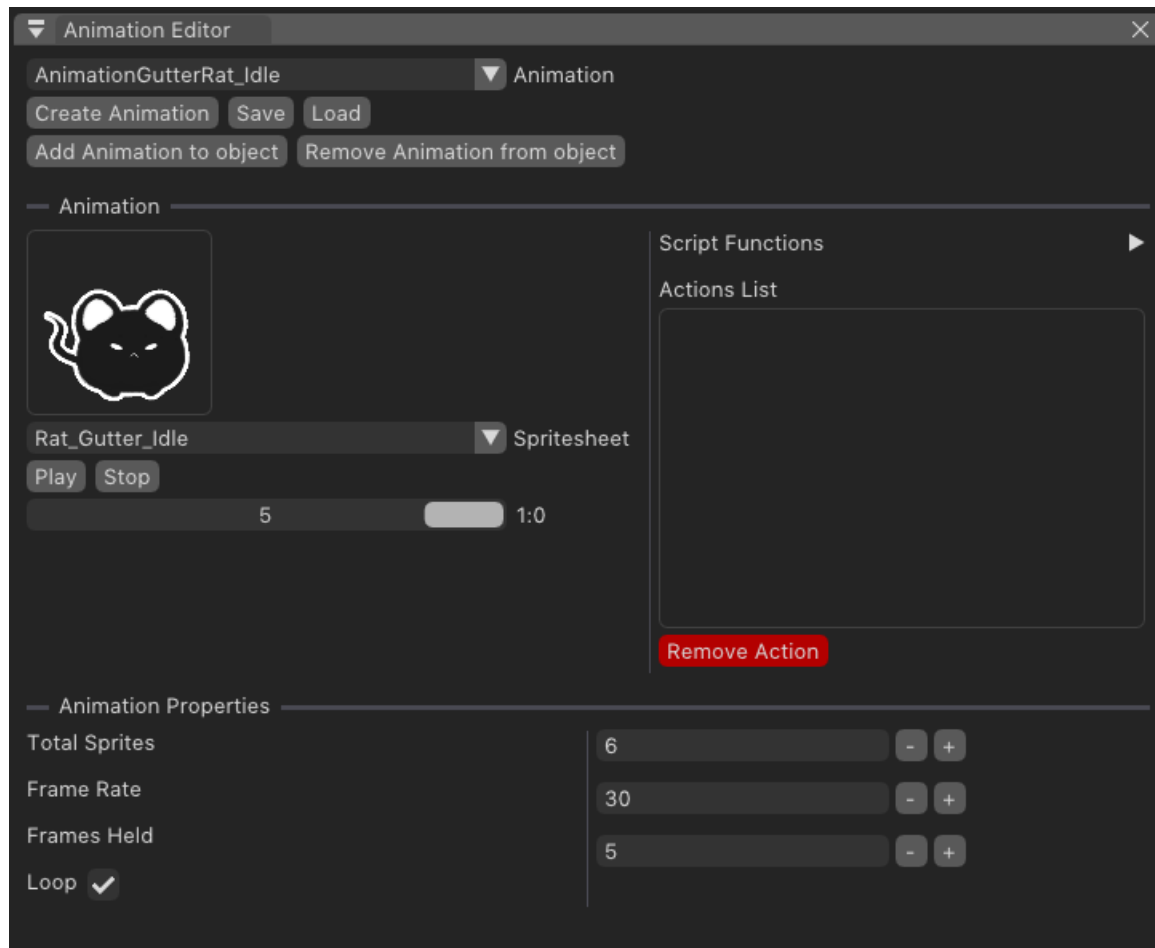
If there is no entity selected, the animation editor will display “No entity selected”.



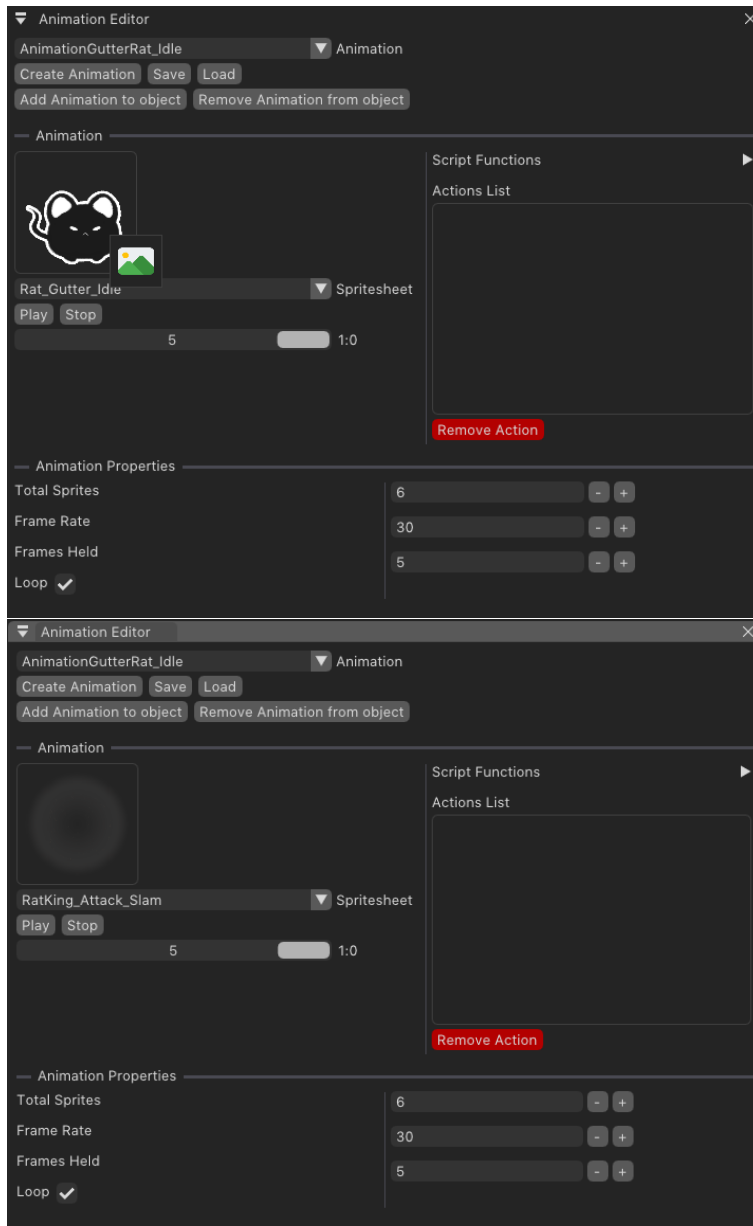
If an entity is selected but it does not have an animation component, it will display the ‘Create Animation’ button. Upon clicking this, the user can create an animation and an animation component will be assigned to the selected entity.



If there is an animation selected, it can be previewed, added/removed from the currently-selected entity, and its other properties can also be edited.

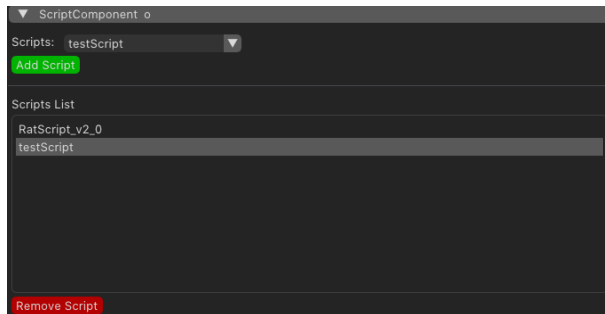


The user can also drag and drop spritesheet textures from the assets browser into the spritesheet window to change the sprite sheet the current animation uses.

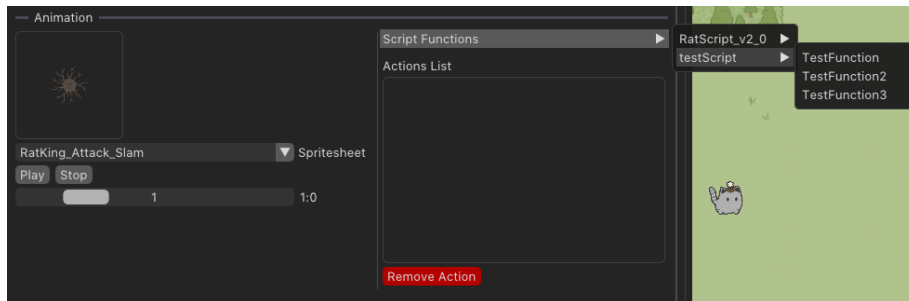


Animation Events

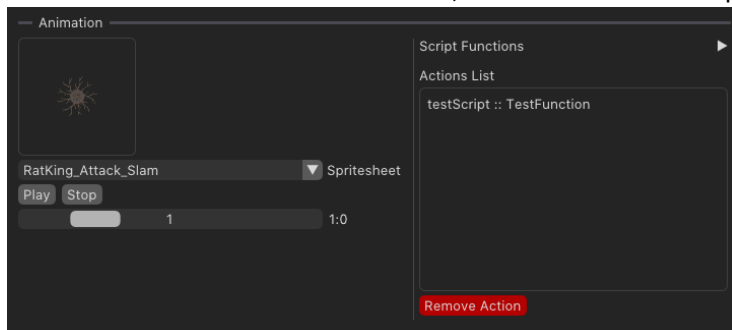
Animation events can be added to specific frames of any animation. The user first has to add the desired script to the object's script component from the property editor window.



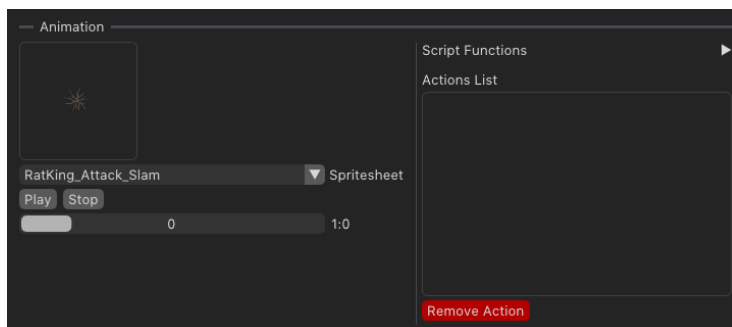
The user can then go into the animation editor window, and hover over Script Functions, and select the corresponding script and functions they would like to call. The selected function will only be called during the selected frame of the animation.



The function is now added to frame 1, as shown in the example below.

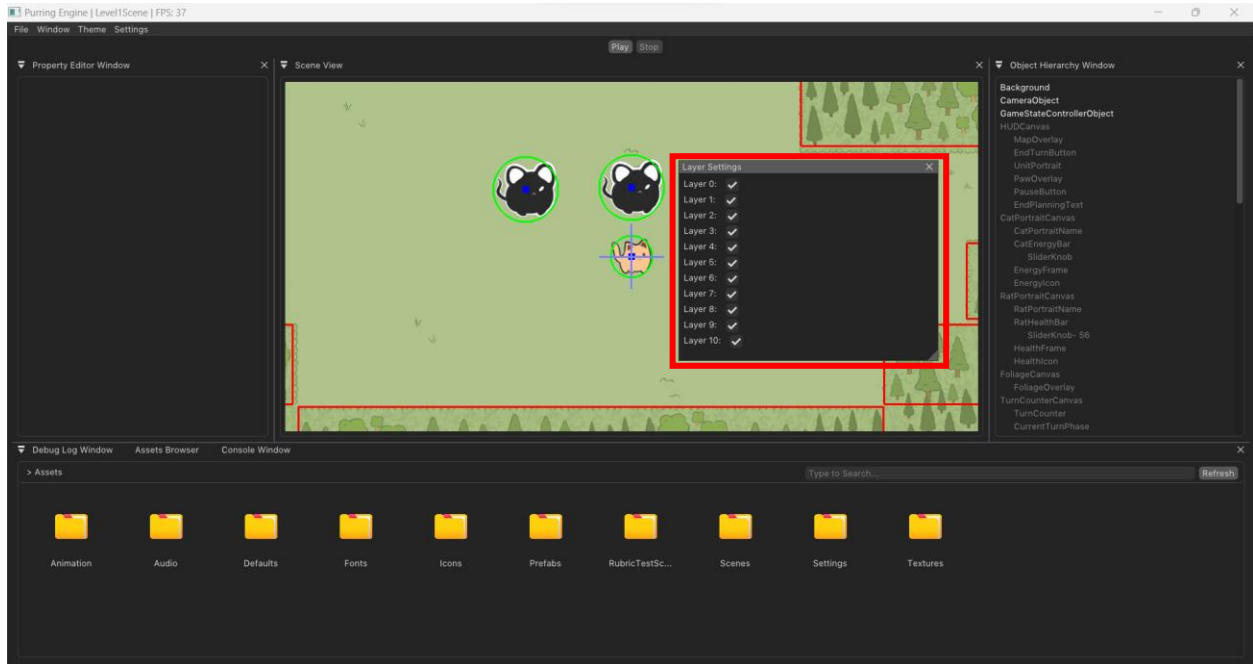


All other frames will not have this function unless it is also added to the other frames.

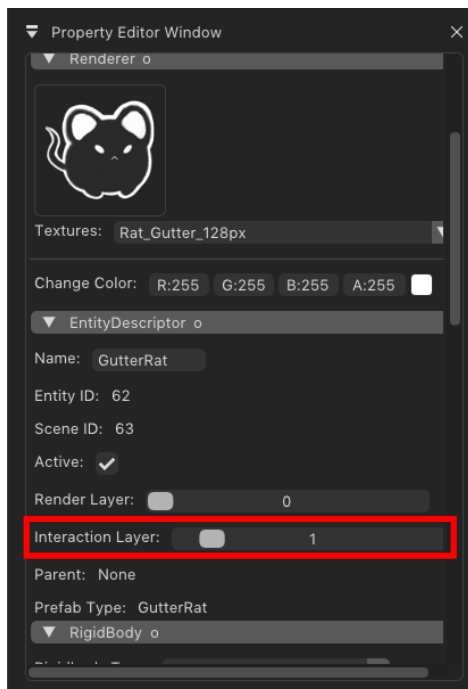


Layer Settings Window

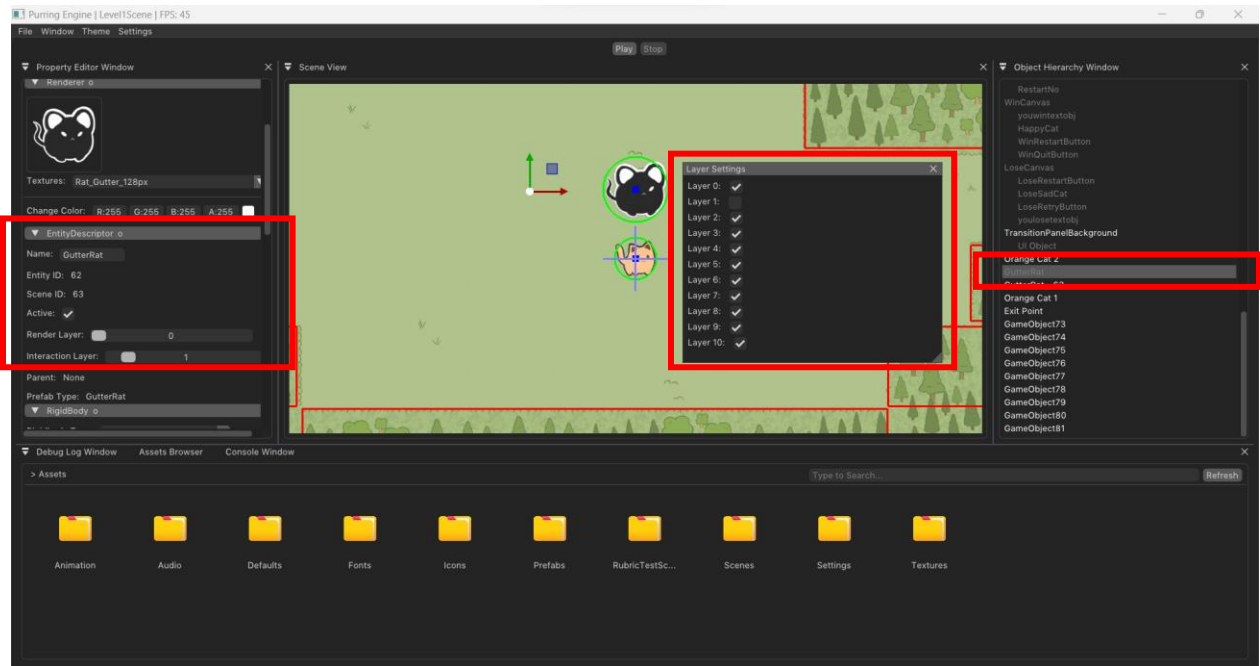
This window gives the user access to the different layers and allows for enabling/disabling all objects grouped under their respective layers.



In the example below, this object has been set to an interaction layer of 1 using the slider.



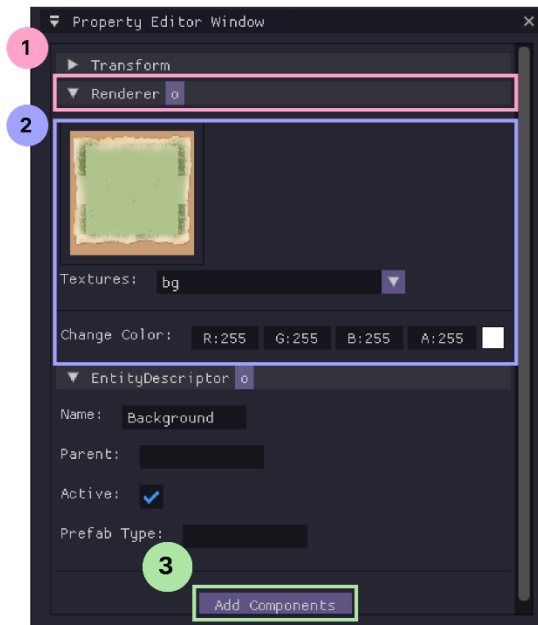
If the user disables Layer 1, all objects under layer 1 will be disabled and appear greyed out in the hierarchy, even if they are still 'Active' in the Entity Descriptor Component. Layer settings take priority over individual objects' active or inactive states.



General Guides

The following are some general user guides for how to use the different parts of the editor.

Property Editor Window:



1. Header

- Component can be expanded or collapsed here
- Reset/Remove components here
- Some components cannot be removed, only reset

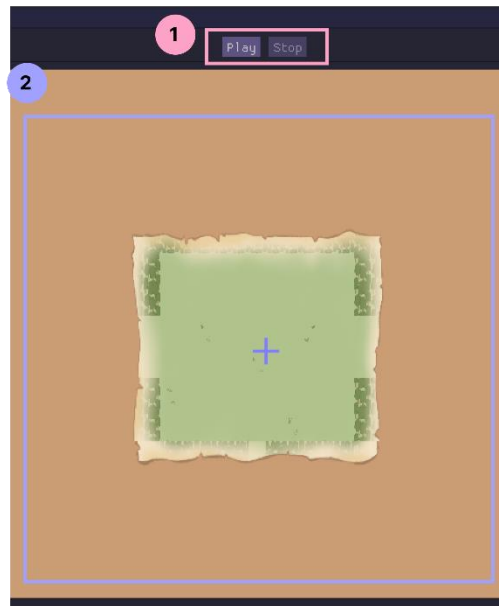
2. Contents

- Values specific to the component
- View and modify variables here by either inputting values or selecting from dropdown boxes

3. Add Component

- New components can be added to the selected entity here

Scene View:



1. Play/Pause/Stop

- When play is clicked, it plays the scene as a full screen and runs all scripts on entities present
- Stop is only clickable in play mode, it stops the scene and returns to editor mode
- Pause is only visible and clickable in play mode, it pauses the scene immediately and resumes when 'Play' is clicked

2. Scene View

- The user can view the entire current scene here
- The user can also move around the scene or rotate the editor camera

Physics Config Window:



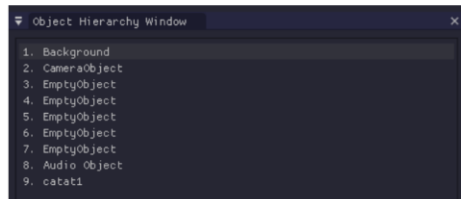
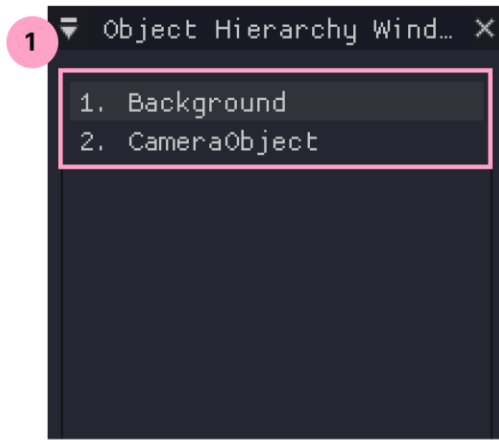
1. Grid and Debug toggles

- The user can adjust the grid size of the scene here
- The user can also choose to toggle debug lines or grid on/off

2. Layer Matrix

- Contains all possible collision layers
- Controls which layers can collide with one another
- If the checkbox is checked, those 2 affected layers can collide
- For example, unchecking the topmost left checkbox will disable collision between layer 0 and layer 9

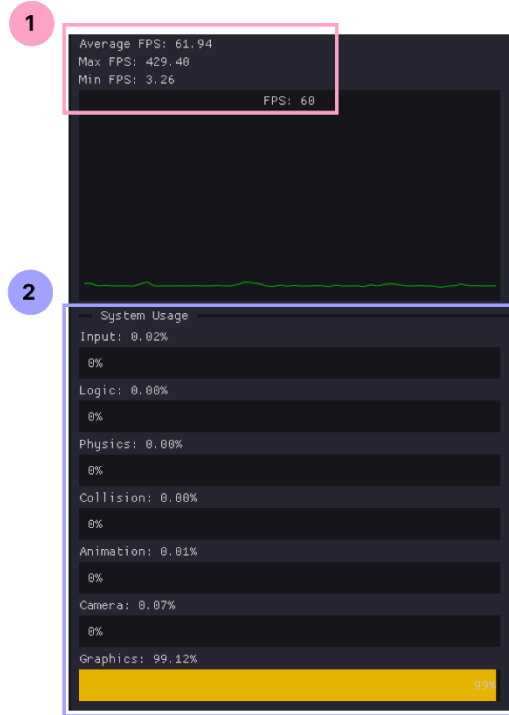
Object Hierarchy Window:



1. Objects

- Contains all objects currently in the scene
- Objects have an ID attached to them in the order that they were created in
- Entity A can be childed under entity B by clicking and dragging entity A onto entity B
- Right-clicking on any empty space in the window will bring up a box
- The user can then select what type of entity they would like to create

Performance Window:



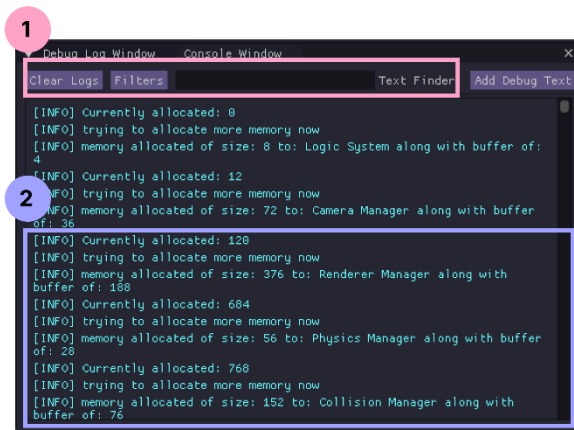
1. FPS Tracker

- The current average, minimum, and maximum FPS can be calculated here

2. System Usage

- Percentage of total memory being used is displayed via a gauge
- The gauge is black by default, but has an orange fill if it is being used

Debug Log Window:



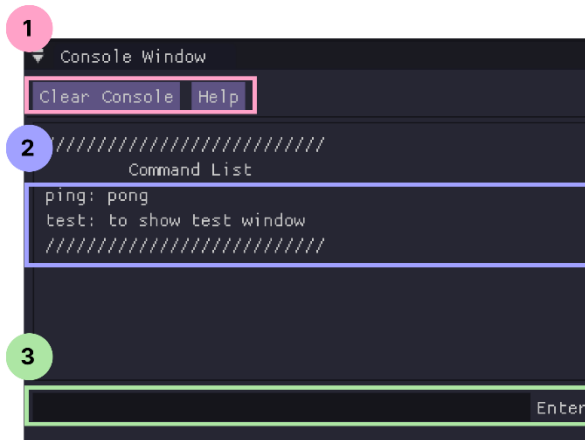
1. Clear Logs/Filters/Search

- The user can filter the types of debug logs they are able to see
- Different types are colored differently
- The user can clear all debug logs currently displayed
- The user can type in specific texts that they want to filter the debug logs by

2. Content

- All debug text printed is displayed here

Console Window:



1. Clear/Help

- The user can clear the console of all texts currently displayed
- The user can click the 'Help' button to view a list of all valid commands

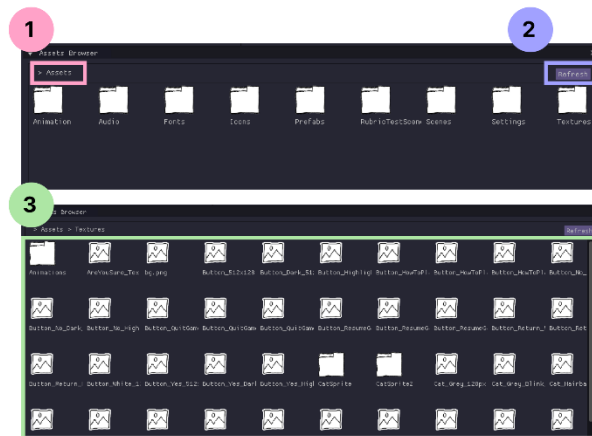
2. Command Window

- All commands sent and responded to are displayed here

3. Input

- The user can input commands in this text box and press ENTER when they are ready to send it

Assets Browser:



1. File Directory

- The user can view the file directory of the current folders shown in the window

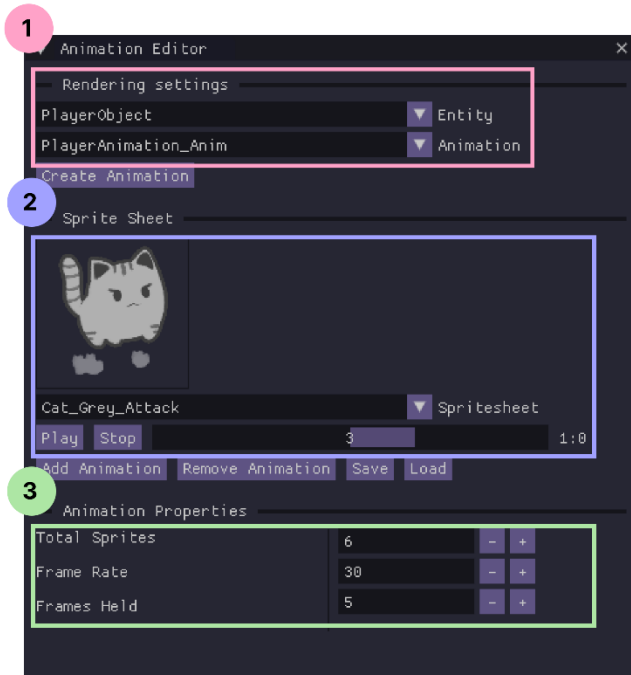
2. Refresh

- Upon importing or deleting assets, the user can click 'Refresh' to show the updated assets

3. View Assets

- The user can view all assets in this window
- Drag and drop assets from file explorer or within the editor is also supported

Animation Editor Window:



1. Entity/Animation

- The user can select which entity to assign the specified animation to
- New animations can be created, or existing animations can be used

2. Preview Window

- A preview of the animation is shown
- The user can play and stop the preview as needed
- The animation can be saved as a .json file at any time

3. Animation Properties

- The user can adjust the total sprites the sheet has, the animation frame rate, and the number of frames to hold for each sprite

About Entities

Render Order

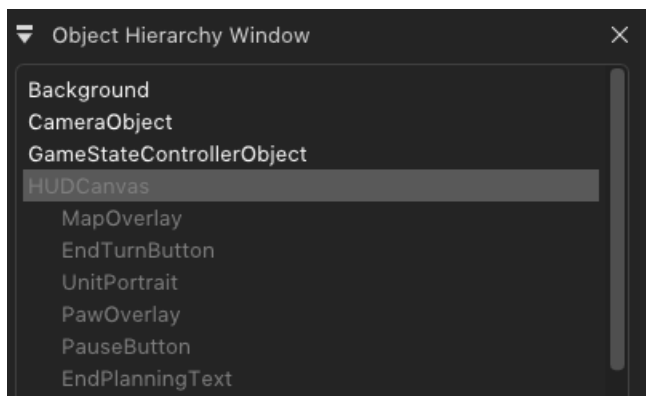
Entities' render order are affected in the following ways, in order of priority:

1. Render Layer order in the Entity Descriptor component
 - a. Adjust the slider to change the render layer the entity belongs to
 - b. A higher number will render the entity later
(ie. Render Layer 1 will be displayed below Render Layer 2)
2. Entity order in the Object Hierarchy/list Window
 - a. An entity lower down the list will render later
(ie. Obstacle 11 will be displayed below Obstacle 12)

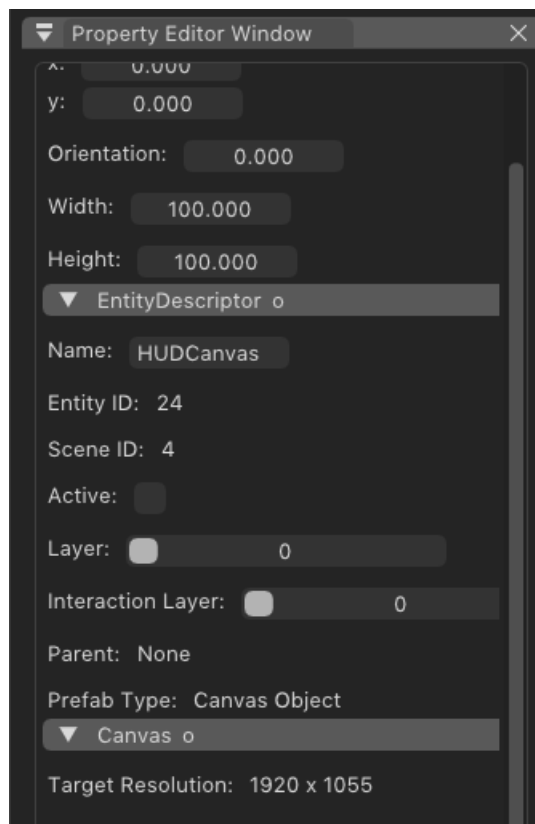
Note that children will always render above their parent, regardless of their render layer order.

Enabling/Disabling Entities

If an entity is currently not active in the scene, it will not appear in scene view at all. It will remain in the object hierarchy window but its displayed name will be greyed out.



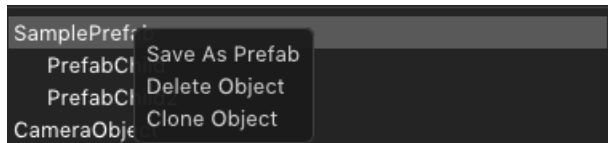
The user can enable or disable an entity by clicking on the box next to the 'Active' label in the Entity Descriptor component.



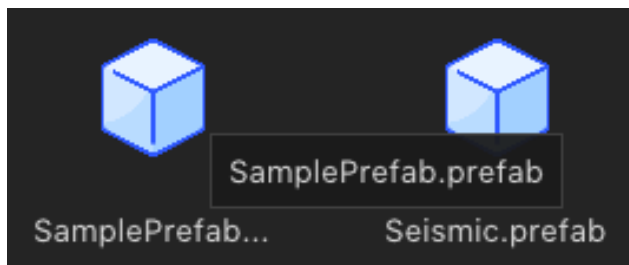
Prefab Editor

Any object can be made into a prefab, which saves the object and all its settings as a template that can be reused in any scene.

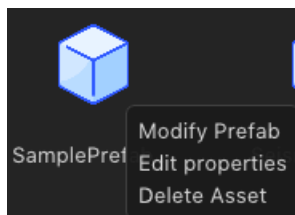
To make an object a prefab, the user has to right click on an object in the hierarchy and select 'Save As Prefab'.



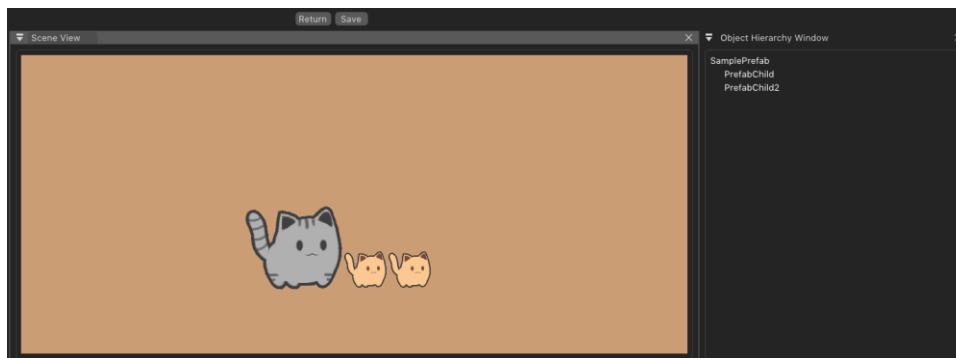
The saved prefab will appear under Assets > Prefabs in the assets browser after clicking 'Refresh'.



Existing prefabs can be edited. This can be done by right clicking on the prefab in the assets browser and selecting 'Modify Prefab'.



Changes can be made to the prefab and saved.



Prefabs can consist of 1 singular object, or a parent and 1 or more children.

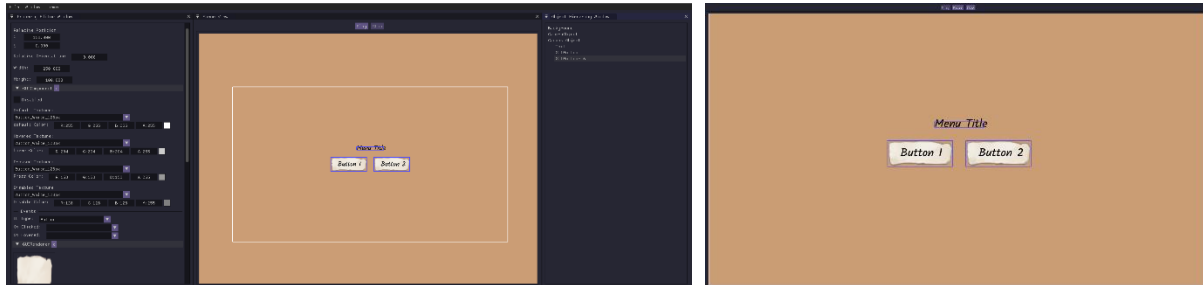
Deleting a prefab parent in a scene will orphan the children.

About Default Objects

Canvas Objects

[Last updated: 6 Jan 2023]

Canvas objects are world space objects in editor mode that represent the bounds and position of screen space (the game window) when in playmode, as pictured in the 2nd screenshot below. Because they are in world space, you can zoom the camera out to see the whole UI layout.

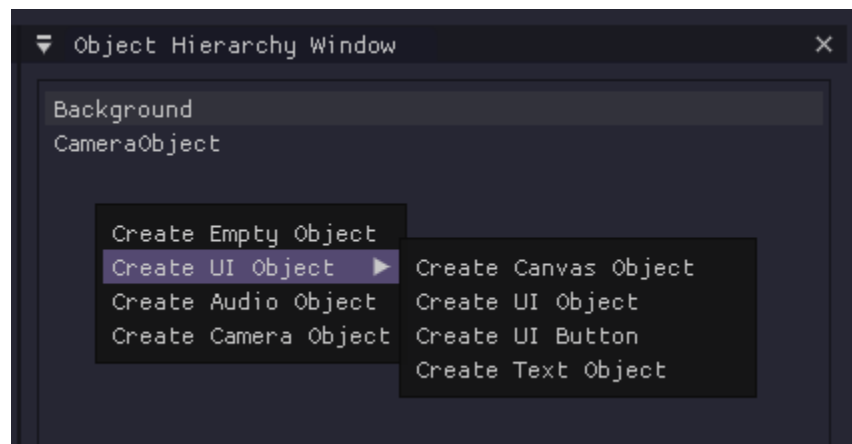


UI objects and buttons that are not childed to canvases will not get drawn and cannot be interacted with.

- [As of 6 Jan, not implemented yet] Note that text objects will still be drawn if not childed to a canvas object, just that they will be rendered in world space instead of screen space.

Creating Canvas Objects

Right-click an empty space in the Object Hierarchy window and select “Create UI Object” > “Create Canvas Object”



Alternatively, create an empty object and add a Canvas component to it.

Text Objects

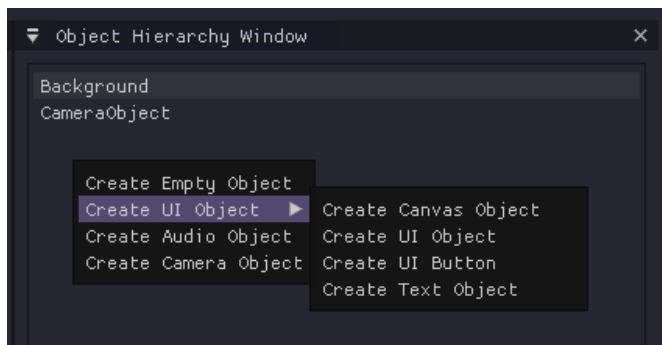
Text objects are world space objects that are bounded by a text box. The position, width and height of the text object is relative to its text box.



Text object will only be drawn if childed to a canvas object.

Creating Text Objects

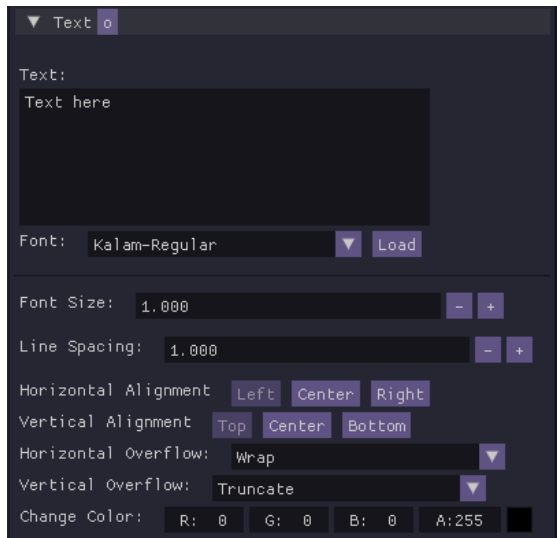
Right-click an empty space in the Object Hierarchy window and select “Create UI Object” > “Create Text Object”



Text object can only be created this way and cannot be added as a component to an empty object.

Editing Text Objects

Text objects have the following editable parameters



Text: Text here will be rendered to the screen

Font: Font to use

Font Size: Size of font to be rendered

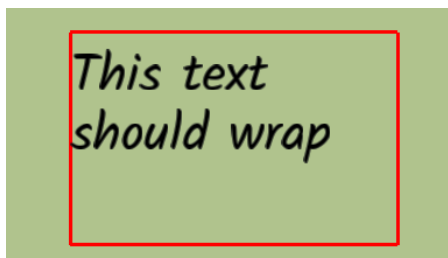
Line Spacing: Spacing ratio between subsequent lines

Horizontal Alignment: Left/Center/Right

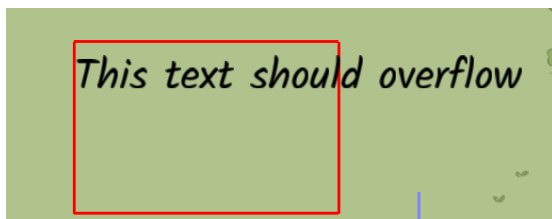
Vertical Alignment: Top/Center/Bottom

Horizontal Overflow:

- Wrap

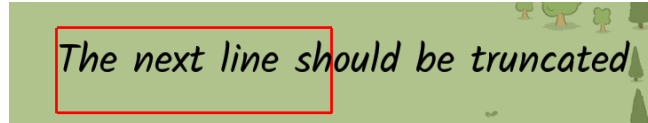


- Overflow

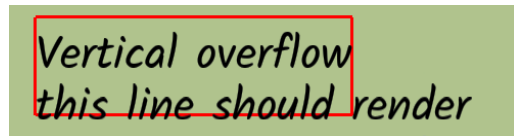


Vertical Overflow:

- Truncate

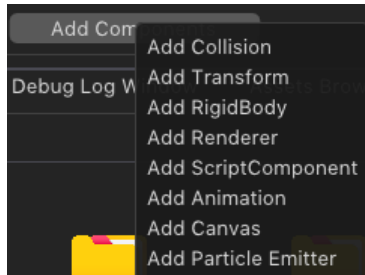


- Overflow



Particle Emitter

The particle emitter is a component that can be added to any default object. The component will turn the object into a particle system, and the user can then modify various parameters.

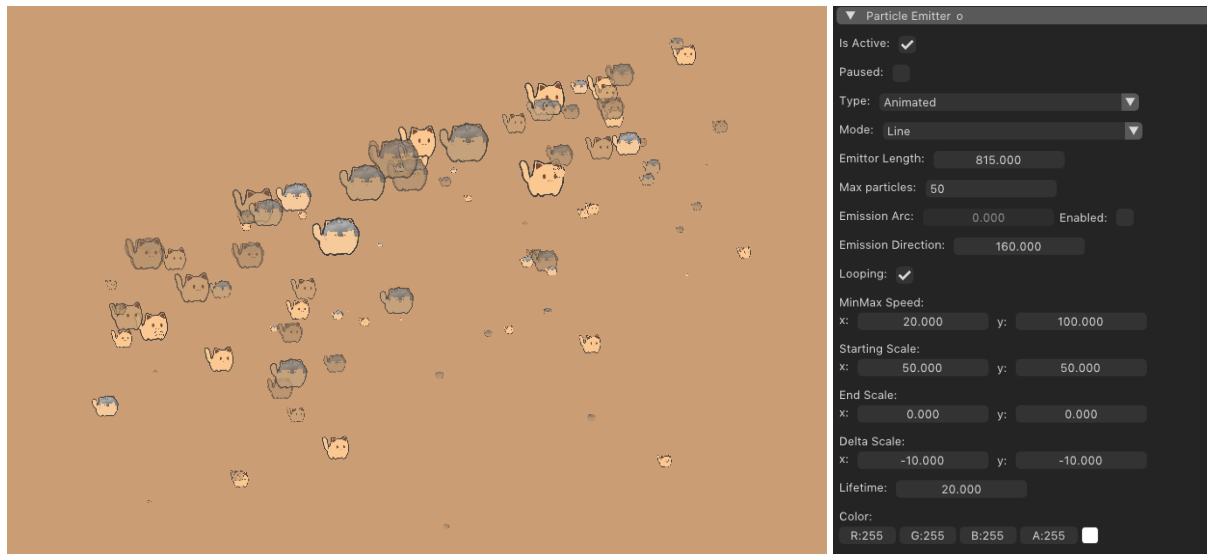


Like any other component, the particle emitter can be enabled or disabled using the 'Is Active' toggle. Particle effects always play by default even if the object is not selected, but individual objects can be paused using the 'Paused' toggle.

The rest of the parameters are as listed.

Variable	Description
Type	Animated: Uses the animation component on the object, overrides renderer component Textured: Uses the renderer component on the object
Mode	Line: Particles emit along a single line Point: Particles emit from a single point
Emittor Length	Only applicable for Line mode, controls the length of the line
Max particles	Maximum number of particles that can be emitted
Emission Arc	Disabled by default, can be enabled. Controls how much spread the particles emit in
Emission Direction	Controls which direction the particles emit from
Looping	Toggles whether particle emission should loop or stop once lifetime is over
MinMax Speed	Minimum and maximum speed particles can travel at
Starting Scale	Values for xy scale at the start of emission
End Scale	Values for xy scale at the end of emission
Delta Scale	Values for xy scale changes over delta time
Lifetime	Number of seconds before each particle is destroyed
Color	Allows for changing of RGBA values

This is an example of what the particle emission looks like, with the following settings.



This is the same particle emission, but with a Textured type instead, with the optional Animator component removed.

