Table of Contents

ImGui
ImGuiContext
ImLineStyle
ImPane
ImPaneFlags
ImPaneOffset
ImPaneStyle
ImPaneStyleExtensions
ImProgressBarStyle
ImProgressBarStyleExtensions
ImRect
ImScope
ImScrollArea
ImScrollAreaStyle
ImScrollAreaStyleExtensions
ImSkipLineStyle
ImSliderStyle
ImSliderStyleExtensions
ImTextFieldStyle
ImTextFieldStyleExtensions
ImTextStyle
ImTextStyleExtensions
ImWindow
IStyle
PruneFlag
StyleExtensions
UnmanagedImWindow
WindowBehaviorExtensions

InitialPrefabs.NimGui.Common

ImIdUtility

InitialPrefabs.NimGui.Inputs

InputHelper
InputText
InputTextExtensions
Mouse
Mouse.State
MouseExtensions

Installation

> It is important to follow **all relevant steps** on this page!

Access your asset via the `Package Manager -> My Assets`, find NimGui, download and import it into your project.

Dependencies

ImGui aims to be easy to setup, but requires the minimum dependencies such as

- Burst Compiler: 1.7.0
- Unity Collections: 1.3.1
- Unity Mathematics: 1.2.6

All packages above can be installed via the Package Manager. For experimental/preleased/preview packages such as Unity Collections, please ensure that your Package Manager can show them.

For more information, please visit the Unity docs here.

Installing Dependencies Manually.

Alternatively, you can install all these packages by editing `Packages/manifest.json` directly in your project and adding the following to your dependencies:

```
"com.unity.collections": "1.2.3",
"com.unity.burst": "1.6.4",
"com.unity.mathematics": "1.2.6"
```

Burst Compiler

Burst Compiler is used to optimize & vectorize the mesh generation each frame.

Unity Collections

Likewise, because we are using Unity's Burst Compiler, we also utilize Unity's Collection package which are designed to be thread friendly and compatible with Unity's Burst Compiler. Please make sure this is installed via the Package Manager.

Optional Dependencies

Some dependencies are optional because you may or may not be using the packages in your project.

- Universal Render Pipeline (URP)
- Builtin Render Pipeline
- Input System

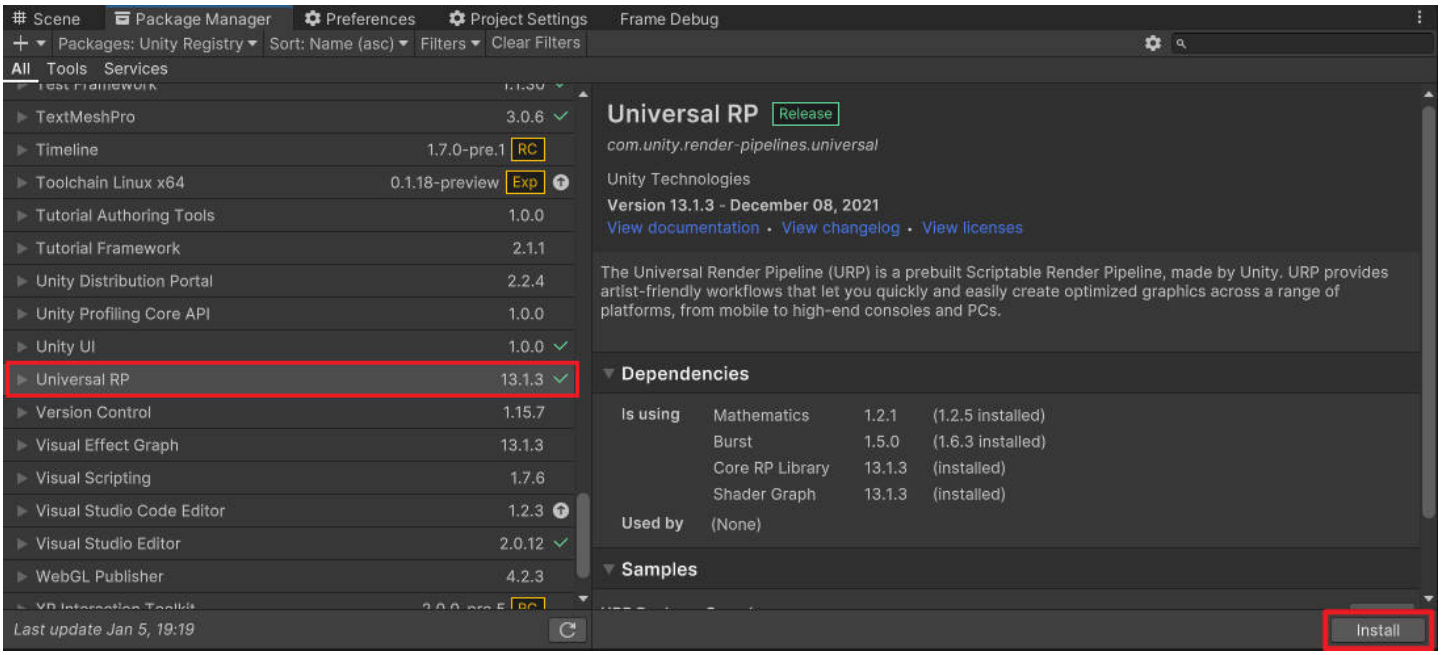
Universal Render Pipeline (URP)

In general any SRP that supports render features should work, however this framework was initially built with URP.

Grab URP from the Package Manager by navigating to:

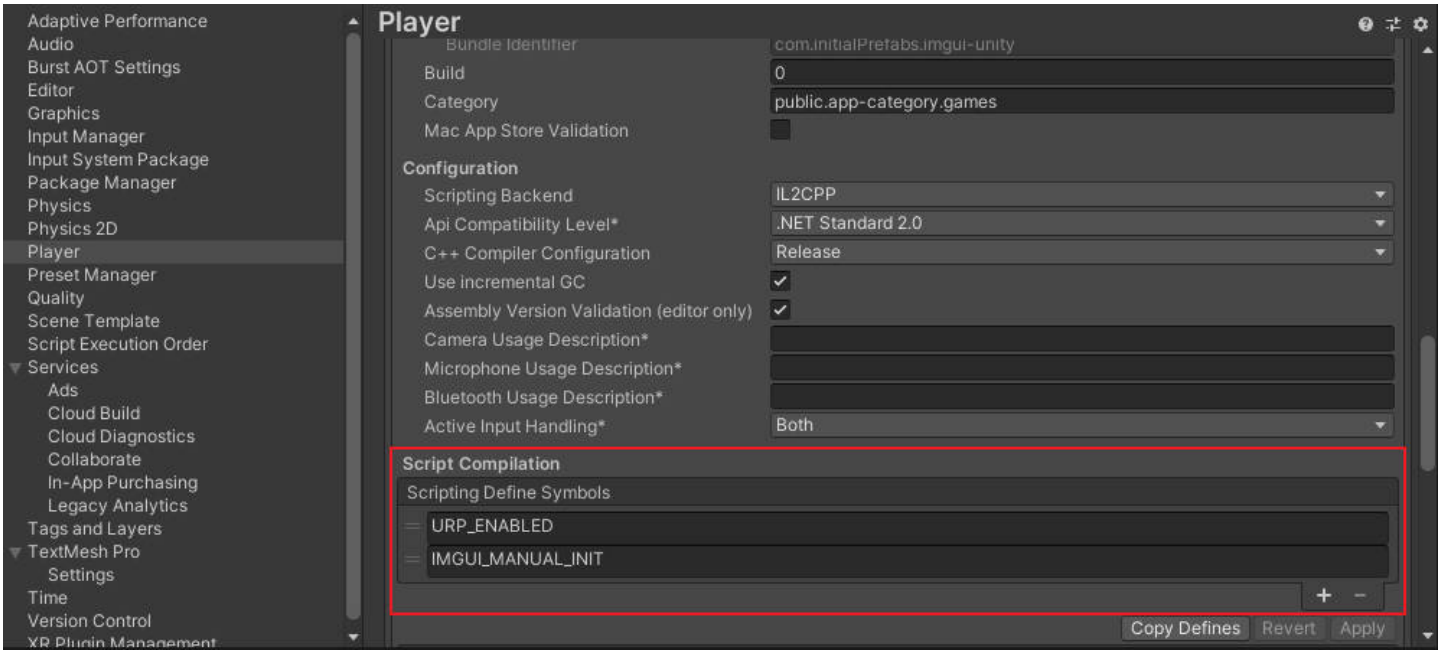
```
Window -> Package Manager -> Unity Registry
```

Find Universal RP and click the install button.



Setting Up the Render Feature

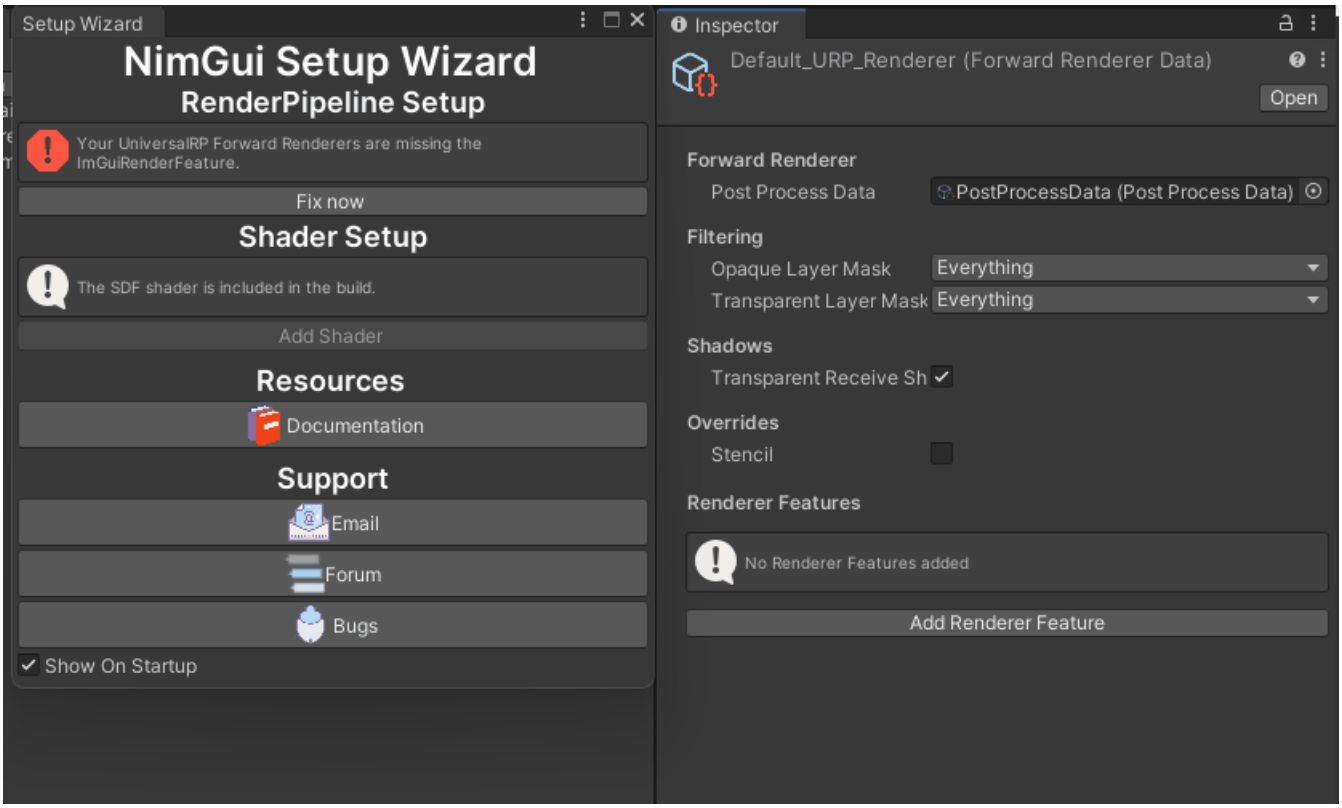
1. Access your Scripting Defines, by going to `Edit -> Project Settings -> Player`
2. Go to `Player -> Other Settings -> Scripting Define Symbols`.
3. Add `URP_ENABLED` as a scripting symbol.



Updating an existing URP Asset

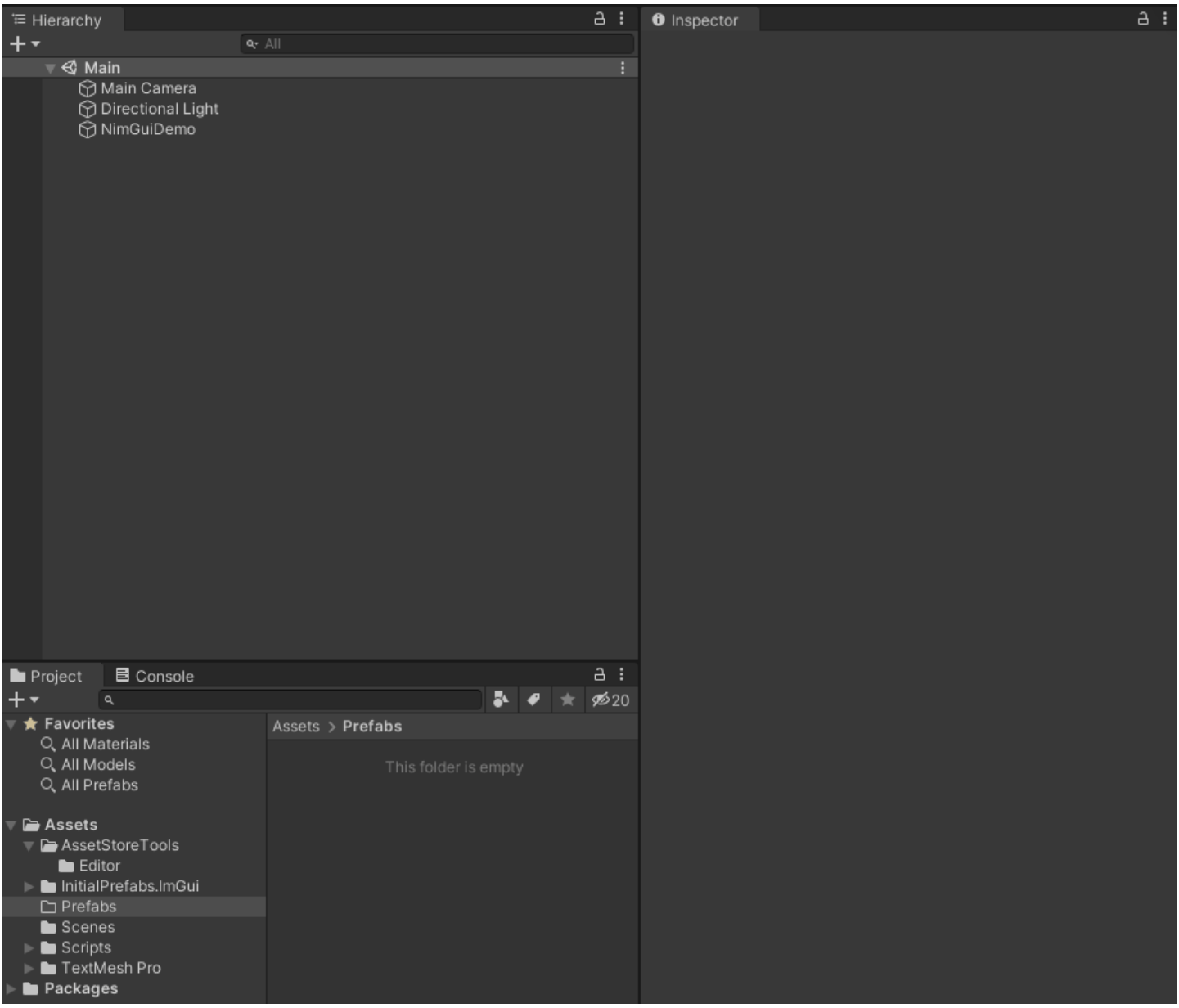
If you already have an existing URP Asset assigned to your Graphics Settings.

1. Go to `Tools -> NimGui -> Setup Wizard`.
2. Under `RenderPipeline Setup`, click the **Fix now** button. This will add the `ImGuiRenderFeature` to your existing URP Asset (don't worry you can undo this by pressing `Ctrl/Cmd + Z`).



Starting Fresh with URP

1. Create a Universal Render Pipeline asset, by right clicking in the project view.
2. Select `Rendering -> Universal Render Pipeline -> Pipeline Asset (Forward Renderer)`
3. Select the Forward Renderer asset which was created alongside your Pipeline Asset and click, `Add Render Feature`. Find `ImGuiRenderFeature` from the dropdown and add it to your Render Features. > If you are using URP + 2D, URP version 12.x.0 onwards, the 2D workflow natively supports Render Passes. For **version 11.x.0 and below**, you will need to force the render passes to load.



Builtin Render Pipeline

Because we will need a hard dependency on the Camera, you will need to setup the camera manually for the Builtin Render Pipeline compared to the SRP alternative. You will need to call the following functions for set up and clean up:

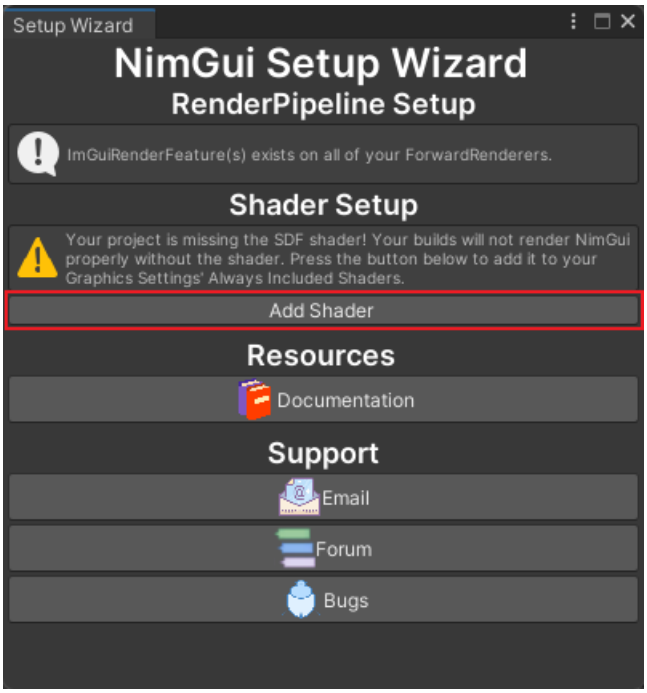
```
// Put this in OnEnable for a MonoBehaviour
DefaultImGuiInitialization.SetupCamera(camera, cameraEvent);

// Put this in OnDisable for a MonoBehaviour
DefaultImGuiInitialization.TearDownCamera(camera, cameraEvent);
```

Adding Shaders to Builds

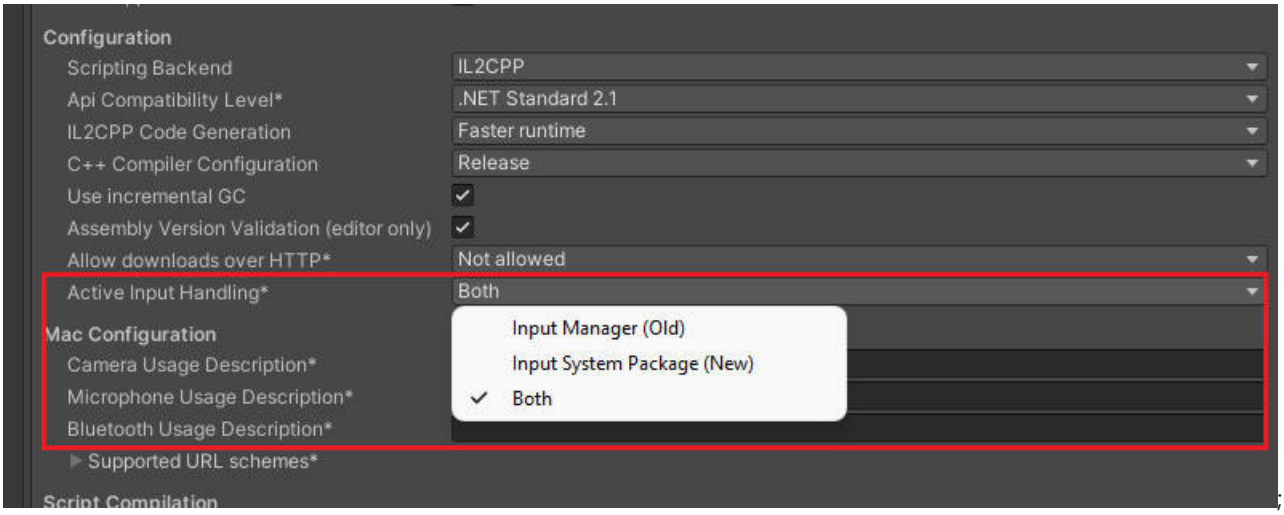
It is important to set up NimGui's SDF shader to always be included in your builds. If you create a build without including the shaders NimGui will not render and may cause your game to crash.

1. Go to `Tools -> NimGui -> Setup Wizard` and a new window will open.
2. Click the `Add Shader`
    - If the button is disabled, then that means the SDF shader is already added to the `Always Included Shaders` of your

graphics settings. 

Choosing your Input Backend

You are free to use Unity's legacy Input Manager or their new Input System.



If you choose `Input Manager (Old)`, then NimGui will use the `UnityEngine.Input` class. If you choose `Input System Package (New)`, then NimGui will use `UnityEngine.InputSystem` as its default input backend.

If you choose `Both`, then NimGui will use the legacy input system by default.

QuickStart

Displaying Your First Widget

Create a new MonoBehaviour called ImGuiExample.cs in your project. Unlike Unity's ImGui, there is no reliance on implementing a separate callback function like `OnGui()`, you can call all ImGui functions in `Update`.

> □□ While it is possible to call ImGui in `FixedUpdate`, it *will* only appear for a single frame until the next time FixedUpdate is called. This is the correct behaviour of how ImGui works, don't fight me on this. If you need to draw during `FixedUpdate`, ***cache*** the data you need and pass that data to the `Update` loop instead.

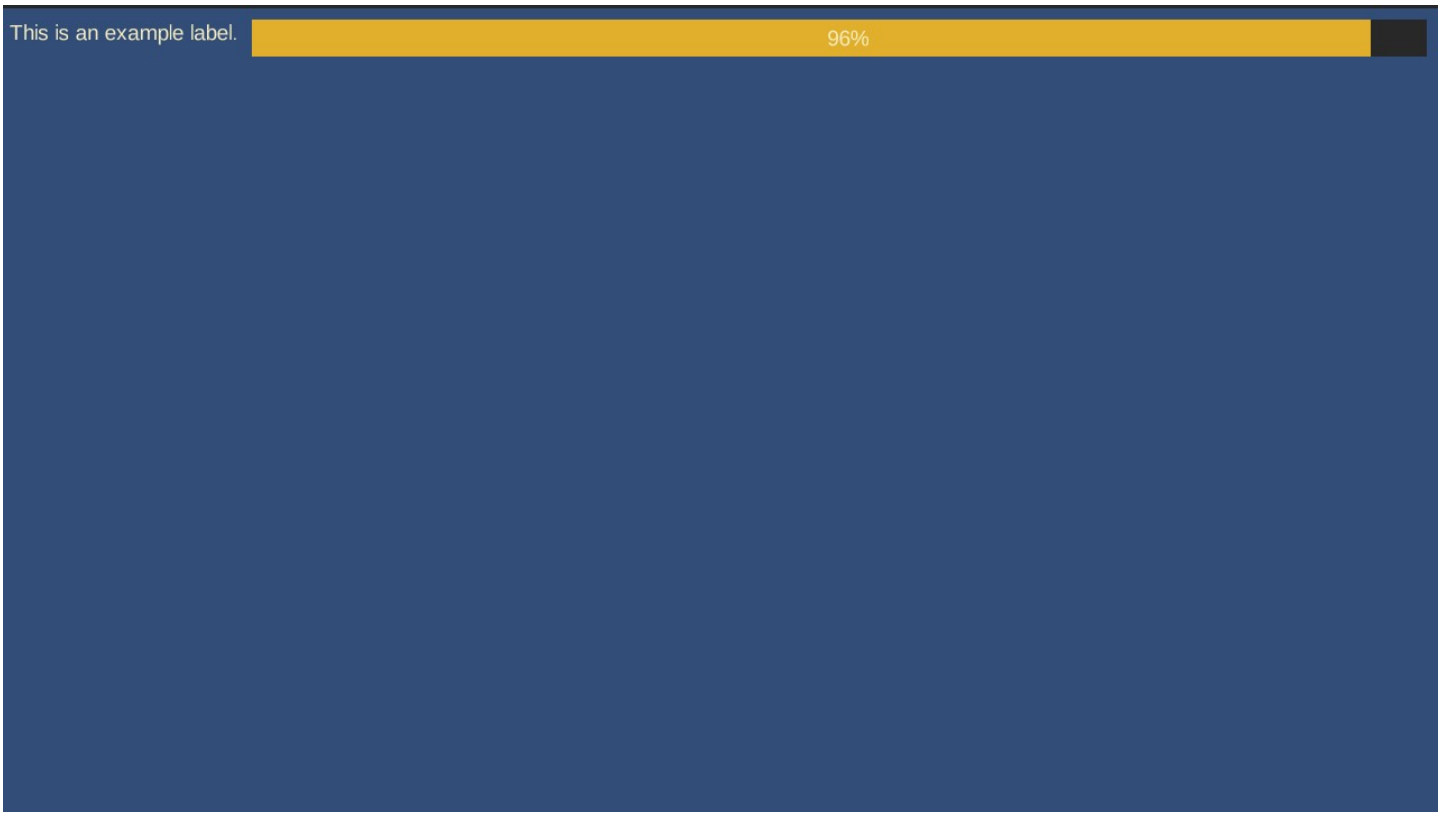
Add the following snippet:

```
using InitialPrefabs.NimGui;

public class ImGuiExample : MonoBehaviour {

    void Update() {
        ImGui.Label("This is an example label.");
        ImGui.SameLine();
        var t = 0.5f * Mathf.Cos(Time.time * 0.5f + Mathf.PI) + 0.5f;
        ImGui.ProgressBar(t);
    }
}
```

This will create a label displayed in the top left corner of the screen with a progress bar.



And that's it! Getting started with `NimGui` is pretty much calling a bunch of static APIs to draw your widgets. We strongly recommend looking at the `NimGuiDemo.cs` file as a point of reference.

Execution Order

NimGui, by default, injects custom Update functions into Unity's Player Loop. This makes it easy to allow developers to quickly get started in using ImGui instead of setting it up.

Initialization

NimGui does not do anything special here.

PreUpdate

In Unity's PreUpdate step, NimGui completes its own previously scheduled Job. This utilitizes Unity's C# Job System in order to construct the mesh which is used to display the UI.

After the mesh is constructed, NimGui submits the mesh for rendering and constructs the command buffer which will playback at the correct CameraEvent. See the Installation page for further information about the `CameraEvent`.

NimGui resets every window's internal states. This includes:

- Draw Commands
- Text storage (bump allocator is reset here)

This allows NimGui to start off at a clean slate before it receives its first draw command from a widget.

Update

This is where you draw your UI with widgets. NimGui doesn't do anything special here.

PreLateUpdate

NimGui doesn't do anything special here.

PostLateUpdate

This is where NimGui collects all draw commands and schedules a job on a separate thread to build the UI. NimGui is scheduled to run immediately after Unity's LateUpdate calls, so you can draww widgets in LateUpdate.

Rendering Step

The CommandBuffer that NimGui internally builds is played back so that the widgets are drawn.

Text

Text is rendered through a custom SDF shader. The UI is built to include the correct UV coordinates so that the SDF shader properly renders text with smooth lines.

Why not use TextMeshPro?

Implementing and configuring the mesh to work properly with TextMeshPro shader is a bit of a mystery. TextMeshPro's font generation seems to skip certain characters despite the font actually supporting it. This makes creating a 1 draw call UI difficult to create.

Using the shipped font

NimGui uses Ubuntu Mono Nerd Font to support unicode characters which are used to render the following glyphs:

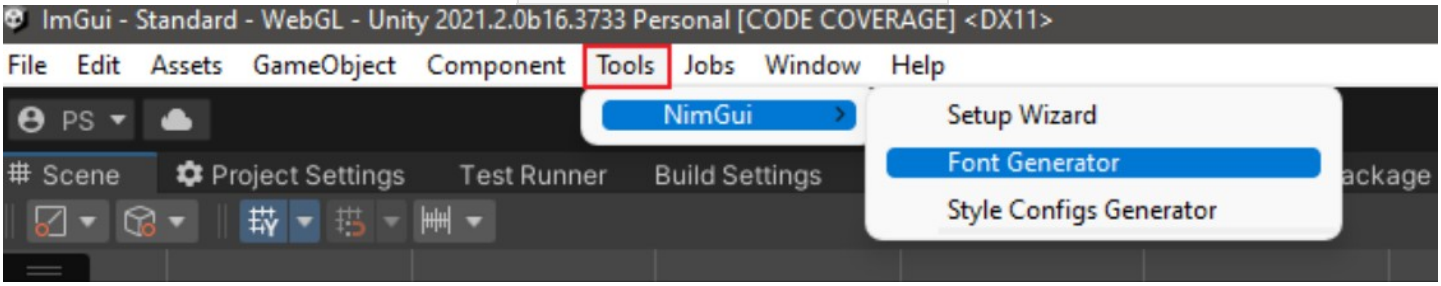
```
☺☻♥♦♣♠•◘○◙♂♀♪♫☼►◄↕‼¶§▬↨↑↓→←∟↔▲▼!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~⌂ÇüéâäàåçêëèïîìÄÅÉæÆôöòûùÿÖÜ¢£¥₧ƒáíóúñÑªº¿⌐¬½¼¡
«»░▒▓│┤╡╢╖╕╣║╗╝╜╛┐└┴┬├─┼╞╟╚╔╩╦╠═╬╧╨╤╥╙╘╒╓╫╪┘┌ ▄▌▐▀αßΓπΣσμτΦΘΩδ∞φε∩≡±≥≤⌠⌡÷≈°∙·√ⁿ²■✖×
```

Ubuntu Mono Nerd Font follows the Ubuntu Font License.

Using a custom font

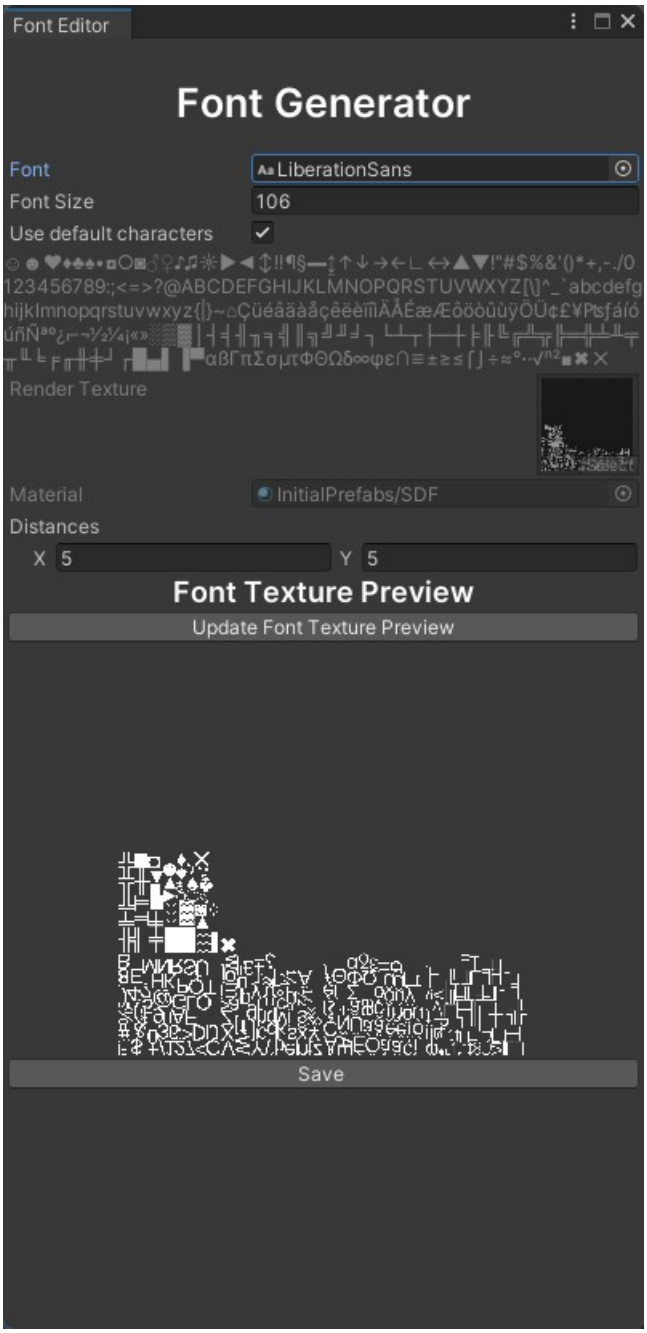
To load your own font, you will need to generate your own Font Texture and SerializedFontData. This is can be done through the **Font Generator** window.

Access the Font Generator window by clicking `Tools -> NimGui -> Font Generator`.



The window will look like the following where you can set your `Font Size` and the signed distance field `Distance`.

> Font Size: The font size should typically be kept large to provide enough space for each glyph's signed distance field.
>
> Distance: The x field describes the pixels that are fully within the contours, while the y distance describes the pixels that are fully outside of the contours.

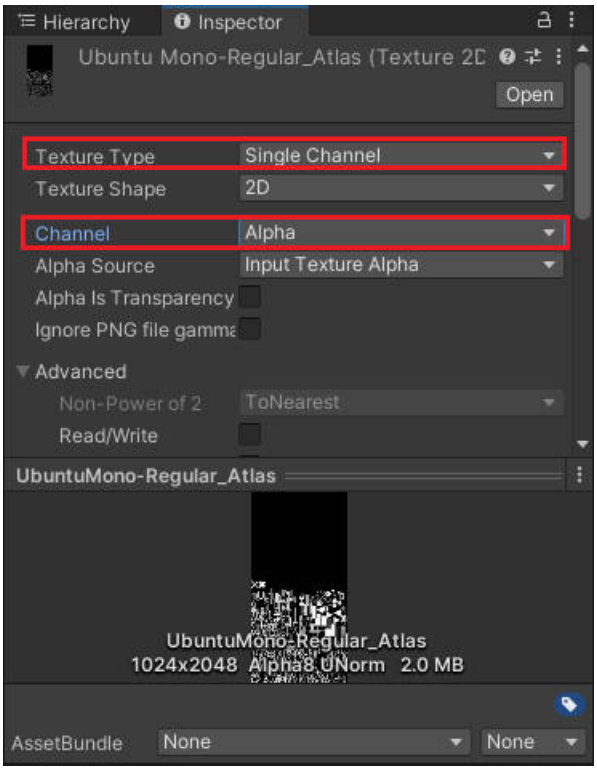


After setting the values, you can update the **Font Texture Preview** to see how the font's texture will look.

Lastly, you can **save** the Texture which will also generate a `SerializedFontAsset` which contains the glyph table that NimGui will reference when drawing fonts.

Font Texture

You must change your Font Texture's Texture type to `Single Channel` and Channel to `Alpha`.



Loading the custom font

To load the custom font call the following function in `OnEnable` in a MonoBehaviour:

```
// Assigned via inspector
[SerializeField] Texture2D tex;
[SerializeField] SerializedFontData fontDataAsset;

// Initialize the static ImGuiRenderUtils and allocate internal memory.
void OnEnable() {
    ImGuiRenderUtils.Initialize(tex, fontDataAsset);
}

// Release allocated memory.
void OnDisable() {
    ImGuiRenderUtils.Release();
}
```

> NOTE: You can only load in 1 custom font at any time. You cannot mix and match fonts together.

How is text stored?

Text is stored as a series of characters in a *bump allocator*. Bump allocators are initialized with a fixed size with a stored index. This index increments everytime we request a new `ImString` from a window.

This allows us to avoid having to rely strictly on `string` datatypes as the `ImString` contains a pointer and a length.

NimGui is initialized to store 8192 characters in the bump allocator.

Limitations

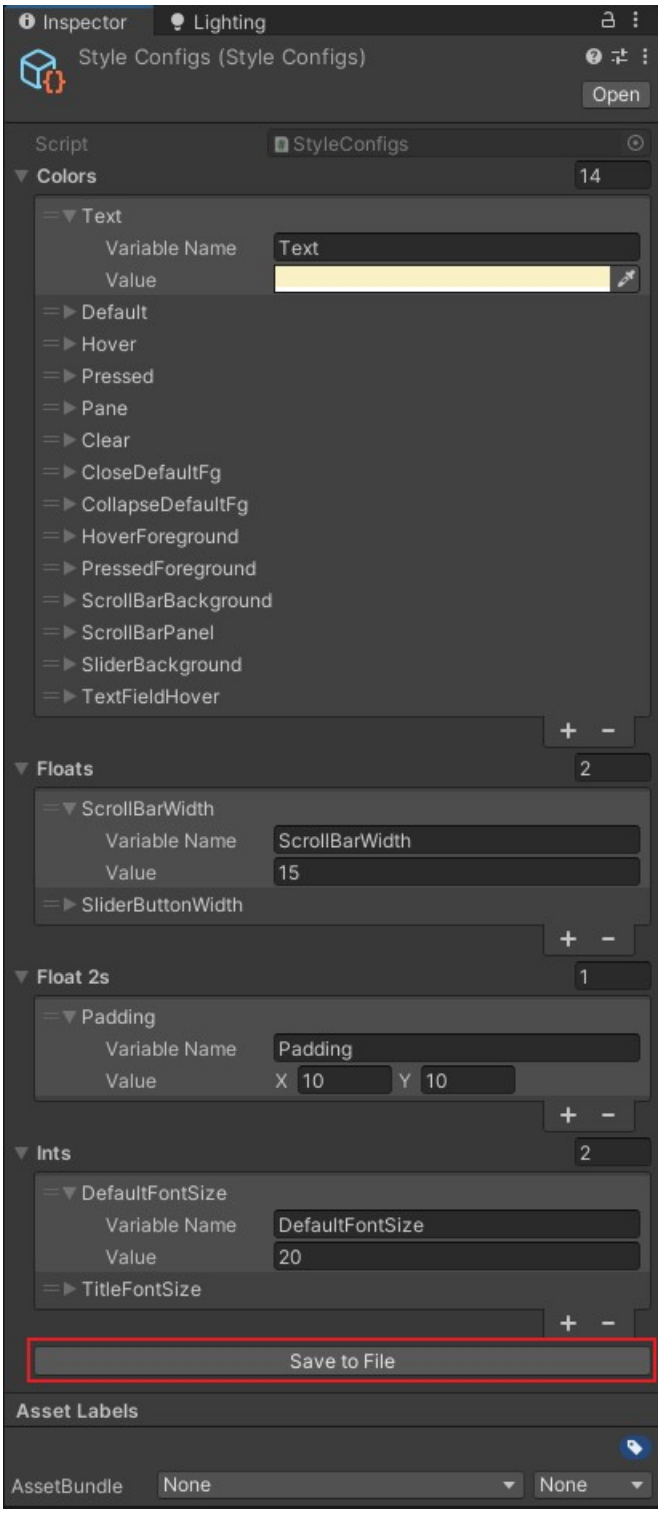
You cannot resize the bump allocator to request more than 8192 at the moment, unless you edit the source directly.

Styles

All default styles are found in the `DefaultStyles.cs`. This is a static file of predefined values that is generated from a `StylesConfig` scriptable object.

StylesConfig

You can create your own StyleConfig by right clicking in your Project and selecting, `Scriptable Objects -> StyleConfigs`.



You can add color, floats, integers, and float2s to define a colorscheme/properties. Once you are done configuring your style, you can generate a static C# script in your project by clicking on the `Save to File` button, outlined in the red box.

Using styles

Styles can be provided to each widget by passing in a custom struct with the defined styles. Styles follow a fluent API/builder pattern which allows you to build the struct using a chain of method calls.

NimGui comes with a default style which is defined via `DefaultStyles.cs`.

As an example, let's take a look at the default `ImTextStyle`.

```
// DocStrings are removed in this example.
public partial struct ImTextStyle {

    public int FontSize;
    public HorizontalAlignment Column;
    public VerticalAlignment Row;
    public Color32 TextColor;
    public float2 Padding;

    [MethodImpl(MethodImplOptions.AggressiveInlining)]
    public static ImTextStyle New() {
        return new ImTextStyle {
            FontSize  = DefaultStyles.DefaultFontSize,
            Column    = HorizontalAlignment.Center,
            Row       = VerticalAlignment.Center,
            TextColor = DefaultStyles.Text,
            Padding   = DefaultStyles.Padding
        };
    }
}
```

We can define:

- the size of the text
- the horizontal and vertical alignment
- the text color
- the amount of padding each text has

Because C# `struct`s do not have overridable constructors, we use a static method to define a new default `ImTextStyle`. To help you get started, you can use the static method.

```
ImTextStyle textStyle = ImTextStyle.New();
```

Instead of memorizing the struct's content, we can use the following extension methods to define all the properties.

```
ImTextStyle textStyle = ImTextStyle.New();
textStyle
    .WithColumn(HorizontalAlignment.Left)
    .WithColor(Color.red);
```

Widgets

A widget is a visual element which provides a specific functionality or displays specific information. So far, we introduced in our QuickStart a Label widget and a Progress bar widget, which respectively displays text and displays the current progress of an operation (in our case we just mapped cosine).

NimGui contains a library of widgets grouped into the following categories:

Non Interactable Widgets

- Box
- Label
- Line
- ProgressBar

Interactable Widgets

- Buttons
- Collapsible
- Dropdowns
- Pane
- Scroll Area
- Sliders
- Textfield
- Toggle

Box

Draws a rectangle with a size on the screen.

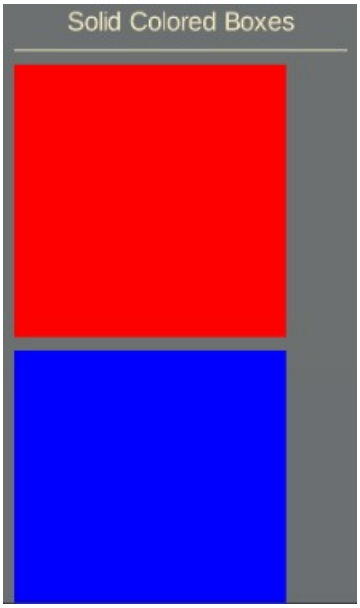
Remarks

When calling the `ImGui.Box(...)` function, the layout will not be updated. This is useful if you would like to overlay the box with some other content. For example, you may want to add text on top of the box.

If you need the layout to be automatic, please pass in a `true` as the last parameter of the function. Please see the page about layouts.

Example

```
// By passing in a true, we force the scope to update.
ImGui.Box(new float2(200), Color.red, true);
ImGui.Box(new float2(200), Color.blue, true);
```



Buttons

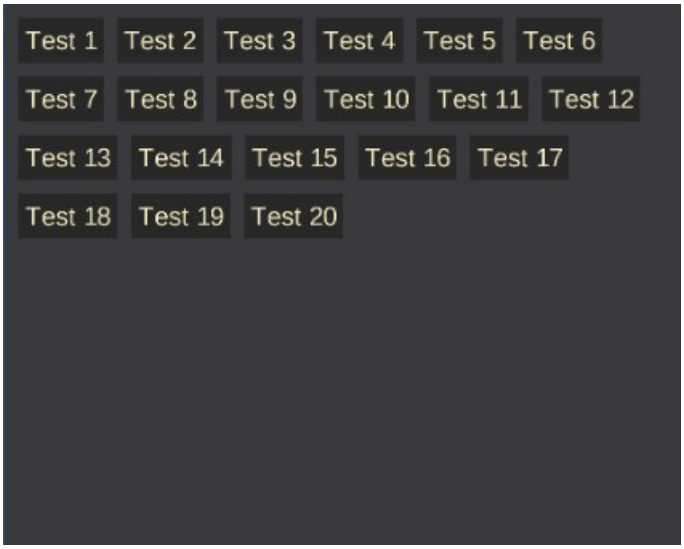
Buttons are interactable widgets which are only registered when the mouse is released after clicking the button.

Remarks

Buttons will respect the relative order they are drawn in. For example, if another button is on top of a button, then the button drawn on top is the one interacted with.

Example

```
for (int i = 0; i < 20; ++i) {
    if (ImGui.Button($"Test Button {i + 1}")) {
        Debug.Log($"Test Button: {i + 1} was pressed!"):
    }
    ImGui.SameLine();
}
```



Collapsible Area

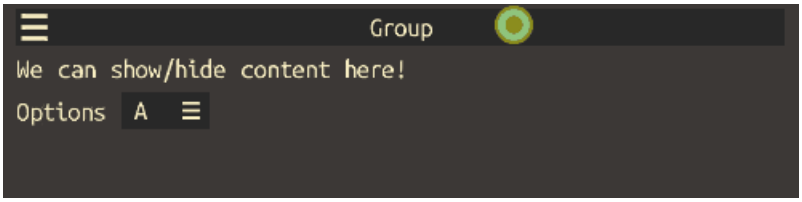
Use collapsible areas to define a region of space that content is drawn can be hidden. To interact with the collasible area, simply click the header button.

Remarks

- The collapsible button header respects the relative order it is drawn in.
- CollapsibleAreas internally cache a unique ID to keep track of whether or not the content should show.
    - Please see the Prune API for more information.

Example

```
using (var collapse = new ImCollapsibleArea("Group")) {
    if (collapse.IsVisible) {
        ImGui.Label("We can show/hide content here!");
        ImGui.Dropdown("Options", new string[] { "A", "B" });
    }
}
```



Dropdown

Dropdowns allow you to select an option based on the available options presented. The dropdown will return the index of the clicked option.

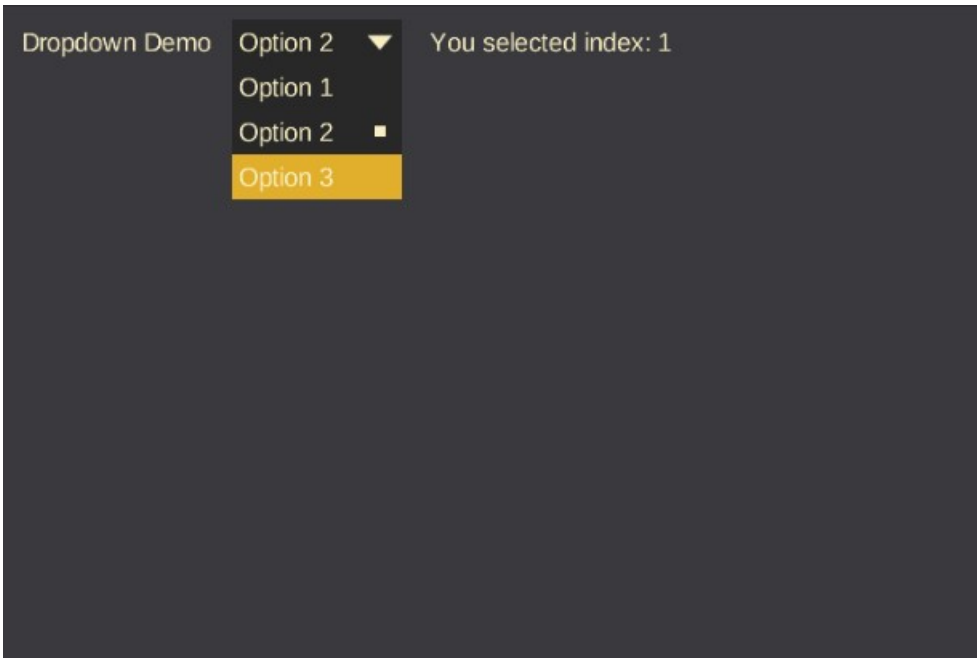
If a click is detected and it is not one of the available options, the dropdown menu will hide itself and keep the selected index the same.

Example

```
int index; // class variable

index = ImGui.Dropdown("Dropdown Demo", new string[] {"Option 1", "Option 2", "Option 3"});
ImGui.SameLine();

ImGui.Label($"You selected index: {index}");
```



Label

Displays text on the screen.

Limitations

1. Non Latin characters are not guarenteed to be supported.
2. Right to Left is currently not supported in Labels.
3. Special escape characters such as `\n` and `\t` are not accounted for.
4. No **bold** or *italicize* support.

Remarks

Internally, labels and any subsequent widget which displays text will calculate the total size of your text with respect to the current layout. Text that contains a lot of content will attempt to respect the widget's scope and automatically push text to the next line.

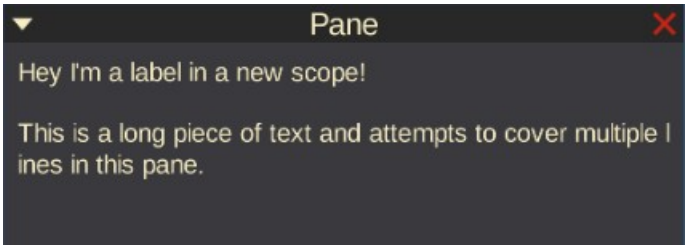
Example

Simple Label

```
for (int i = 0; i < 10; ++i) {
    ImGui.Label("Native ImGuiDemo");
}
```

Multiline Label

```
using (var pane = new ImPane("Pane", new float2(300), new float2(500, 500))) {
    if (pane.IsVisible) {
        ImGui.Label("Hey I'm a label in a new scope!");
        ImGui.Label("This is a long piece of text and attempts to cover multiple lines in this pane.");
    }
}
```



Layout

NimGui does not have a feature rich Layout Engine. If you are looking for a UI system with a complex layout engine, we recommend looking for a different solution.

That said, NimGui does contain a minimal layout engine to help determine where to render each widget. Internally, it keeps track of the last scope and any deltas.

What do we mean by "last scope?"

For example, let's say we did not create a Pane and wanted to draw a label.

```
void Update() {
    ImGui.Label("Hey I'm a label");
}
```

The label would appear in the top left corner like so.

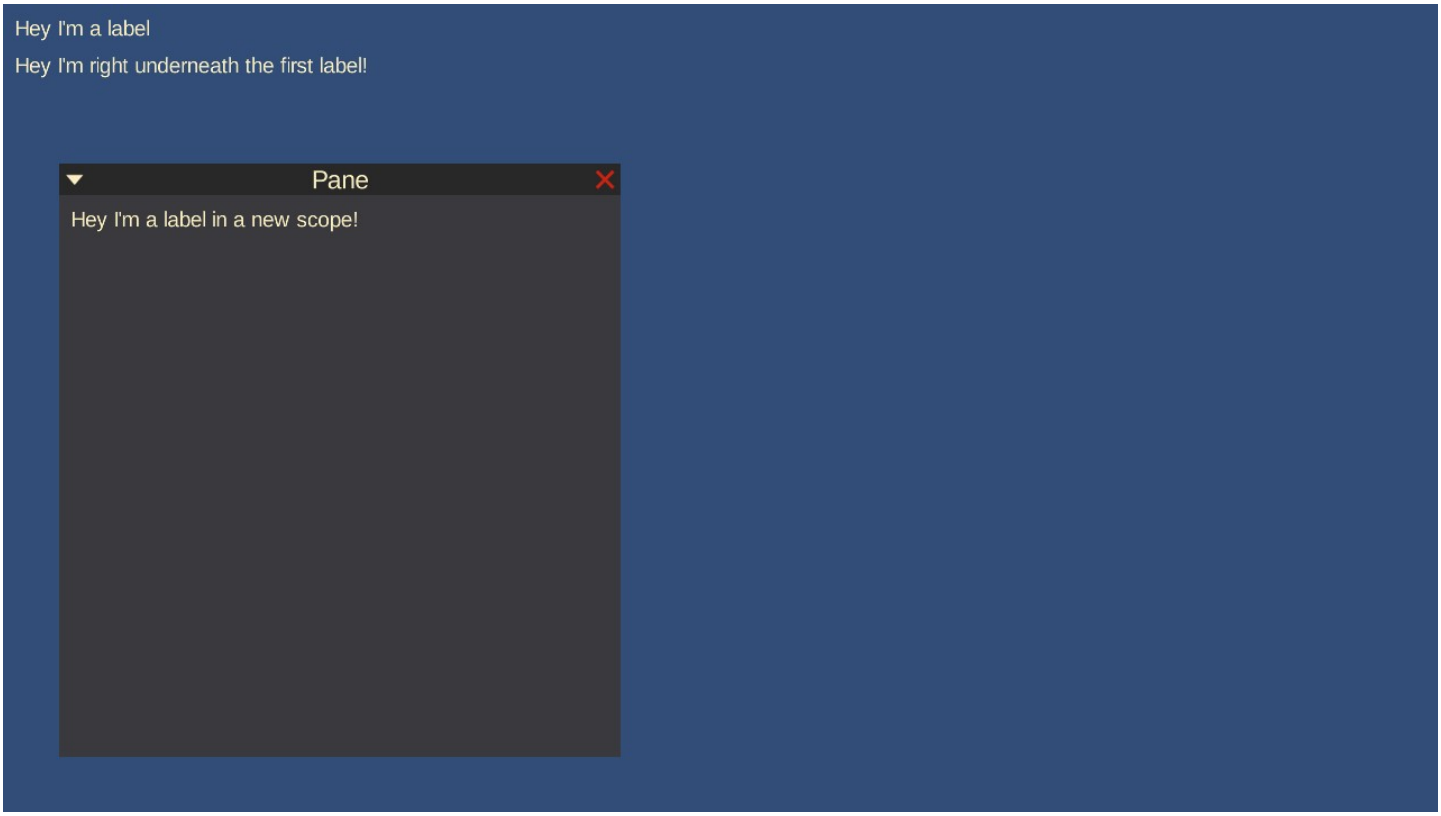
We know we are drawing relative to the screen size. In the image above, the screen resolution is 1280x720p. As we continue to draw more elements by declaring more widgets, we keep track of the previous' widgets dimensions so that we can figure out the next widget's position.

The next widget's position is determined by subtracting the previous widget's height, such that the widgets will be positioned down the screen.

How does this affect Panes?

Panes allow you to declare a scope of a specific size. If you declare a scope of size of 500x500, then internally NimGui, will keep track of this scope until we are finished with the Pane. Panes by default will not update the last known scope's positions as panes are draggable and can overlap on top of other elements.

```
void Update() {
    ImGui.Label("Hey I'm a label");
    using (var pane = new ImPane("Pane", new float2(300), new float2(500, 500))) {
        if (pane.IsVisible) {
            ImGui.Label("Hey I'm a label in a new scope!");
        }
    }
    ImGui.Label("Hey I'm right underneath the first label!");
}
```



In the image above, we can see that

> Hey I'm a label

is shown first. The label right underneath it,

> Hey I'm right underneath the first label!

is actually drawn last because we declared our Pane as the middle elemenet. Panes do not update the current scope as they create a new scope to allow elements to be relatively drawn to it.

This is why our second label

> "Hey I'm a label in a new scope!

in our example code, is drawn within the pane towards the bottom of the screen.

How does this affect `ImGui.SameLine()`?

`ImGui.SameLine()` reads the current scope and attempts to offset the the next widget position to be on the same line. This is done by reversing the offset internally and shifting the position by the previous widget's width.

Line

Draws a horizontal line on the screen.

Example

```
ImGui.Label("Native ImGui Demo");
ImGui.Line();
ImGui.Label("This line is full width");
ImGui.Line();
```

Pane

Panes are small windows which can be dragged, collapsed, and closed. They allow you to group widgets together in an enclosed area, providing contextual information.

Remarks

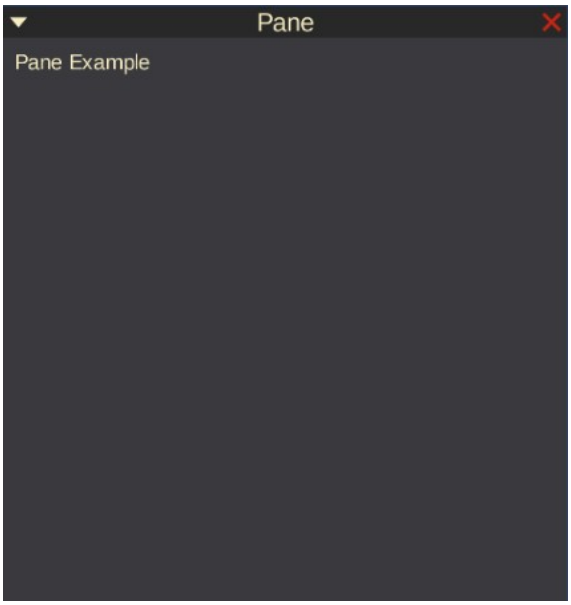
- Panes can be collapsed and closed.
  - The collapsed button is indicated by the arrow in the top left corner.
  - The close button is indicated by the red x in the top right corner.
- The pane struct contains an `IsVisible` property. If this is false, then the content in the pane will not show. It is best to wrap your content logic around the `IsVisible` property (see example below).
  - This value is only true if the Pane is expanded and if the Pane is not closed.
- Panes can be dragged and pinned.
  - Panes internally cache a unique ID so that the pane can be dragged to a new position, collapsed, or closed.
    - Please see the Prune API](/imgui.book/api/InitialPrefabs.NimGui.ImGui.html#InitialPrefabs_NimGui_ImGui_Prune_System_String_InitialPrefabs_NimGui_PruneFlag_) for more information for more information.
- If you add too much content and it does not fit the window, the content will not show (see limitations).

Limitations

- Panes are not resizable like in more modern ImGui implementations (this will be addressed much later).
- Panes cannot be children of each other. The child pane will be clipped.

Example

```
using (ImPane pane = new ImPane("Pane", new float2(600), new float2(500))) {
    // We should only render our label if the IsVisible property is true
    if (pane.IsVisible) {
        ImGui.Label("Pane Example");
    }
}
```



ProgressBar

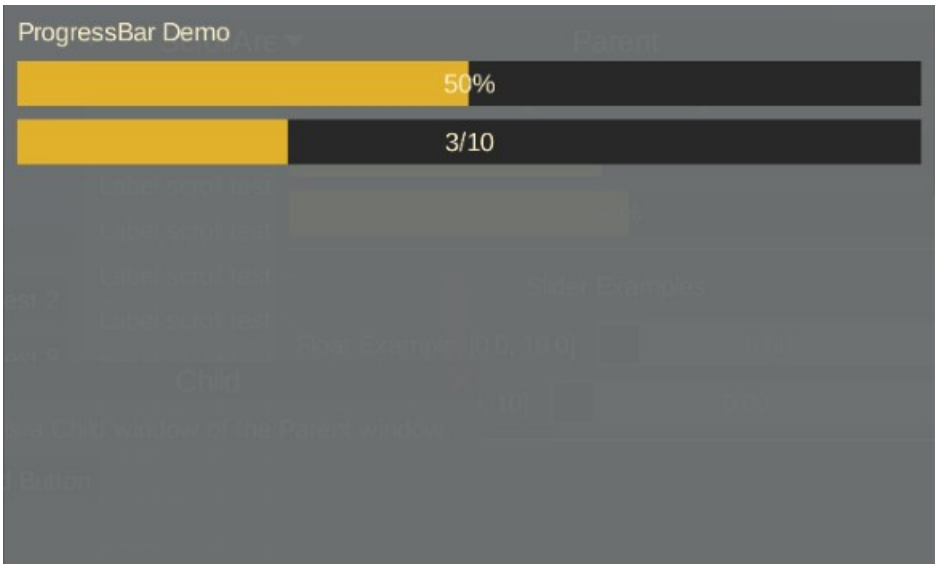
Draws a progress bar on the screen. The amount to fill is based on the **ratio** passed to the static function.

Limitations

- Currently right to left filling isn't supported yet
- Vertical ProgressBars are also not supported yet
- Radial fill is not supported yet

Example

```
ImGui.Label("ProgressBar Demo");
ImGui.ProgressBar(0.5f); // Will draw the label as a percent
ImGui.ProgressBar(3, 10); // Will draw the label as a fraction
```



Scroll Area

Scroll areas provide a dynamic viewport where the content can be scrolled through. Any content that does not fit into the scroll viewport will not be shown. Scroll areas can be scrolled up or down using the mouse wheel or dragging the scroll bar on the side.

Remarks

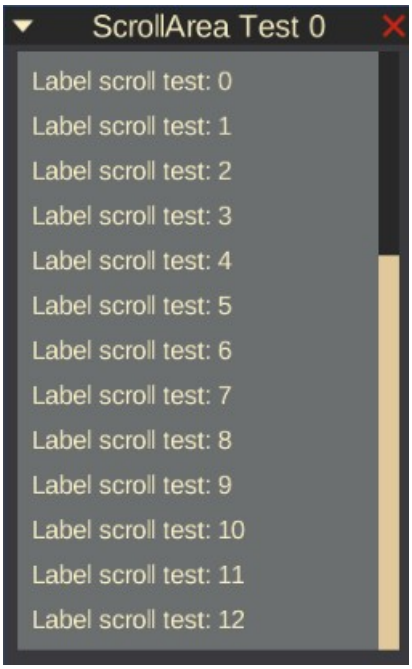
- Scroll areas internally store an ID to keep the offset of the scroll view consistent.
    - To clean up the cached ID, please see the Prune API for more information.

Limitations

- Scroll Areas are not dynamic and do not grow as more elements are drawn in the scroll area. Once the element exceeds the defined space, elements are clipped out of view.

Example

```
using (new ImScrollArea("ScrollArea 0", 450, 1350)) {
    for (int x = 0; x < 20; x++) {
        ImGui.Label($"Label scroll test: {i});
    }
}
```



Sliders (float and integer based)

Sliders allow you to interpolate between two values. You can drag the slider left or right to retrieve a value in between. Dragging the slider to the extremities returns the min/max (left is min, right is max).

Remarks

- Sliders internally cache an ID to keep the current offset for the slider.
  - Please see the Prune API for more information.

Limitations

Sliders currently only support `float` and `int` types. If there is enough demand to support types such as `short`, `ushort`, `uint`, etc it will likely be supported.

Example

```
int iValue;
float fValue;

ImGui.Label($"Int Slider Value: {iValue}");
iValue = ImGui.Slider(0, 10);

ImGui.Label($"Float Slider Value: {fValue}");
fValue = ImGui.Slider(-10f, 10f);
```

TextField

TextFields allow you to type and record characters from your keyboard. Currently, textfields only support Latin characters with simple text layout.

Remarks

- You must supply the `TextField` with a preallocated `StringBuilder`. This is to ensure that each frame, NimGui does not return any garbage collected `string`.
- You must define the string builder's max capacity.

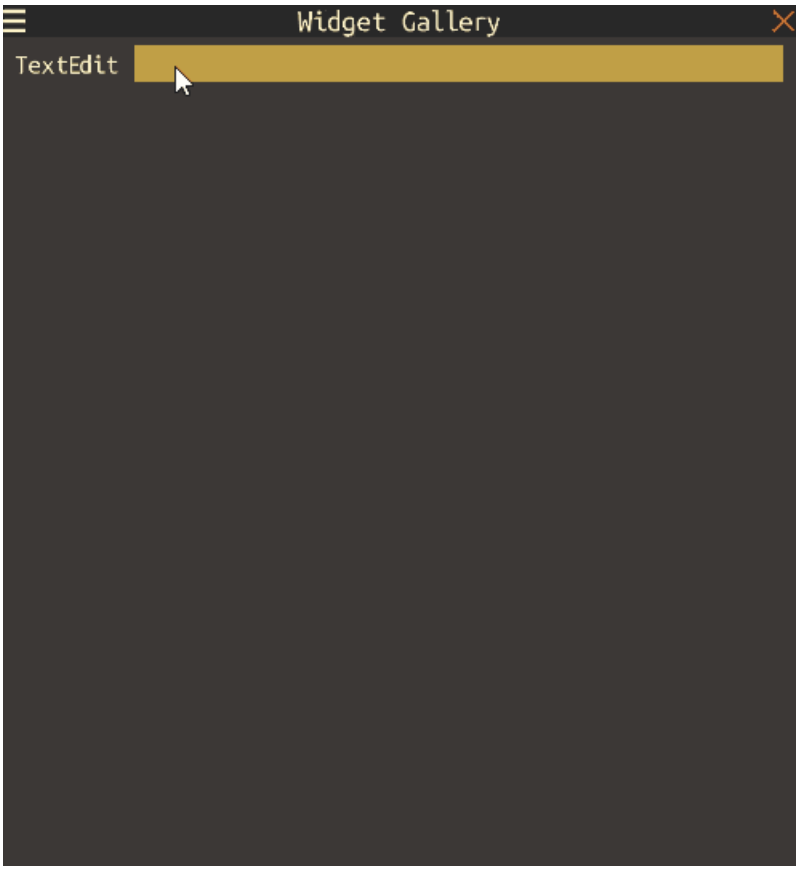
Limitations

- Clicking and highlighting the text is currently not supported.
- You cannot move the cursor left or right and insert a character in between.
- There is no complex text layout support yet.

Example

```
// Allocate a stringBuilder with an initial capacity to avoid having
// to reallocate and copy string data each frame.
StringBuilder stringBuilder = new StringBuilder(128, 256);

ImGui.TextField("TextEdit", stringBuilder);
```



Toggle Button

A toggle button returns true if the box is ticked. Otherwise, it returns false. Use this if you need to keep track of a state.

Remarks

- Toggle states internally cache a bool to keep the state.
    - Please see the Prune API for more information.

Example

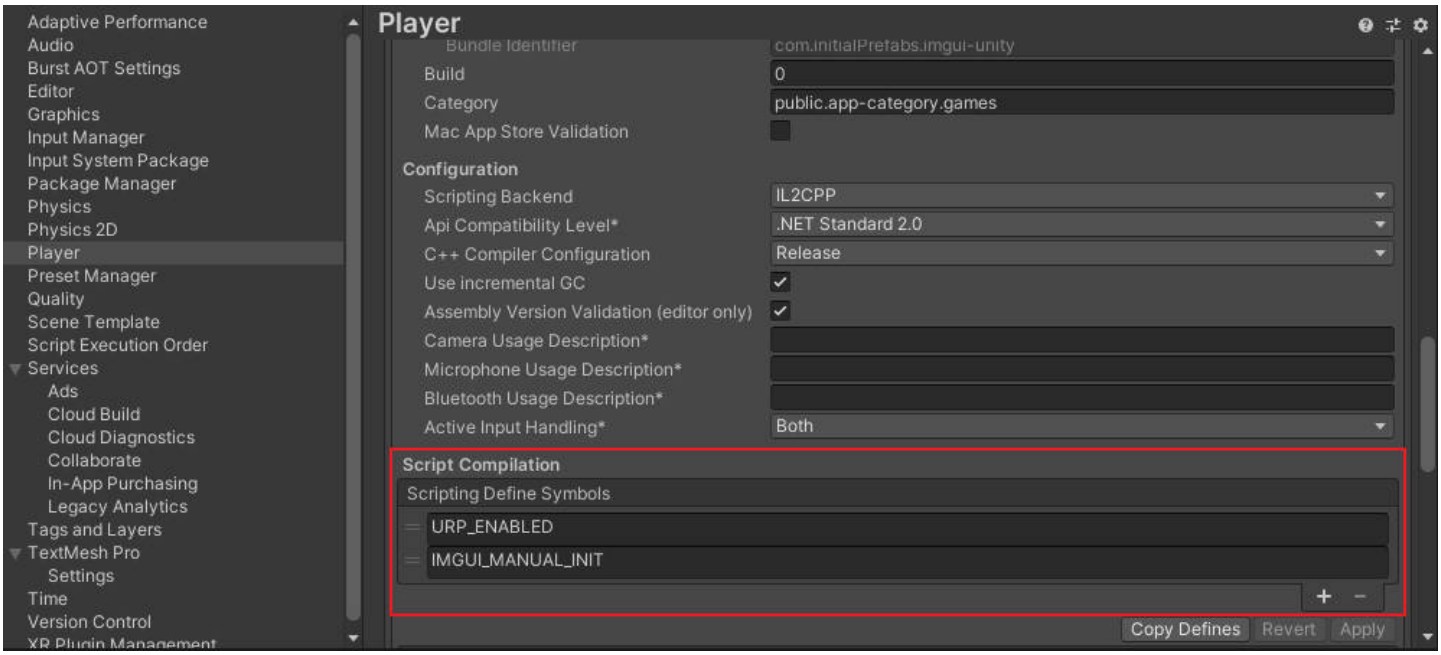
```
bool toggled; // class variable

ImGui.Label($"Toggle Value: {toggled}");
toggled = ImGui.Toggle("Toggle Button Demo");
```

Manual Initialization

If you would like to manually initialize NimGui, add the scripting define, `IMGUI_MANUAL_INIT` to your Player Settings.

This can be found under `Edit -> Player Settings -> Player -> Other Settings -> Scripting Define Symbols`.



Initializing NimGui

You will need to the **Font Texture** and **Serialized Font Data** assets generated via the NimGui Font Generator window.

Please see the Text page for more details on how to use the Font Generator window. Once they're generated, you will need to have a MonoBehavior reference them and initialize all the data that NimGui needs to run. See the example below.

```
public class Init : MonoBehaviour {

    [SerializeField]
    Texture2D fontTexture;

    [SerializeField]
    SerializedFontData fontData;

    void Start() {
        // The following are all required to be initialized
        ImGuiRenderUtils.Initialize(fontTexture, fontData);
        ImGuiContext.Initialize();
        ImGuiRunner.Initialize();
        LegacyInputHelper.Initialize();
    }

    void OnDestroy() {
        // Complete any inflight job before we teardown ImGui
        ImGuiRunner.Dependency.Complete();

        ImGuiContext.Release();
        ImGuiRunner.Release();
        ImGuiRenderUtils.Release();
        LegacyInputHelper.Release();
    }
}
```

Prune

What is pruning?

Pruning in NimGui's context is the act cleaning up global states that are no long in use. The reason behind this is because NimGui internally caches unmanaged data and does not do any kind of automatic cleanup.

Example

All panes intenall store their offset data so that when you drag a pane, the next frame knows that there is an offset so the rect position must be shifted.

This data is typically stored in an `UnsafeHashMap<uint, ImPaneOffset>`, where `uint` is our unique ID for the pane.

```
void Update() {
    // As you continue to use the pane each frame,
    // you continue to read the internally cached
    // offset.
    using (ImPane pane = new ImPane("Title", ...)) {
        ...
    }
}
```

The HashMap is filled for each new pane with a unique ID. if you happen to use the same ID for a different pane, then the cached data is read and applied.

```
void Update() {
    using (ImPane pane = new ImPane("Title", ...)) {
        ...
    }

    // Because our unique identifier is "A new title!".
    // This means that the unique ID is generated for
    // "A new title!" and the hashmap is filled with the
    // new ID.
    using (ImPane pane = new ImPane("A new title!", ...)) {
        ...
    }

    // We're going to draw another pane with the same
    // identifier, "Title." This pane will use the
    // exact same offset declared first.
    using (ImPane pane = new ImPane("Title", ...)) {
        ...
    }
}
```

It is good practice to clear out any internally cached IDs when you're no longer intending to draw the widget.

```
// Pruning example
void OnDestroy() {
    // This will hash the Title string into its unique ID
    // and remove the offset from the HashMap.
    ImGui.Prune("Title");
}

void Update() {
    using (ImPane pane = new ImPane("Title", ...)) {
        ...
    }
}
```

Please see the Prune API for more information!

Performance

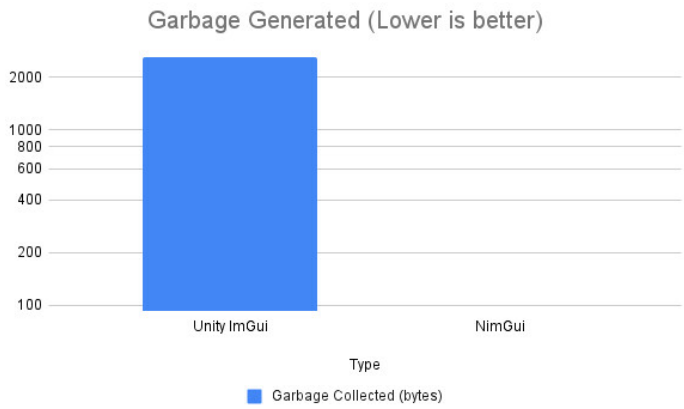
NimGui was designed to be minimalistic and optimized for Windows, macOS, and Linux.

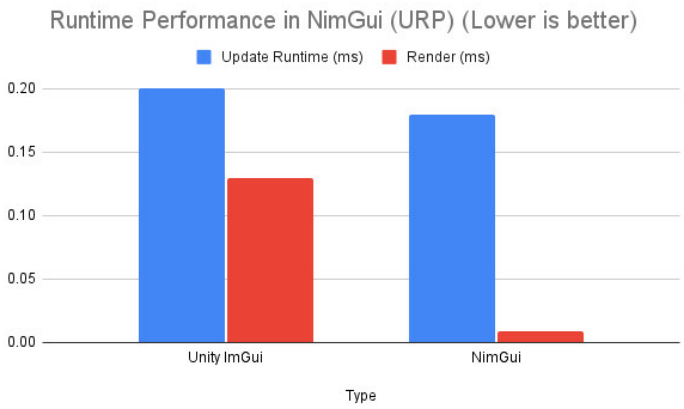
Here are some early stats for NimGui on Windows (Core i3-8100 @ 3.6GHz). All stats are taken with URP.

TYPE	GARBAGE COLLECTED (BYTES)
Unity ImGui	2600
NimGui	0

TYPE	UPDATE RUNTIME (MS)	RENDER (MS)
Unity ImGui	0.2	0.13
NimGui	0.18	0.009

Charts





Frequently Asked Questions

I don't see my UI in builds!

If you see your UI in the editor, but not your builds, make sure that the SDF shader is added to the `Always Included Shaders` of your Graphics Settings. Please see the installation instructions.

Hey, calling a widget in FixedUpdate only shows it for 1 frame!

This is the correct behaviour, please see the Quickstart page for more details.

I can't display widgets from a Job thread or a C# thread.

A command buffer API has not been introduced yet to allow you to draw widgets from a separate job thread.

Linear Colorspace looks more contrasted than Gamma Colorspace.

The SDF shader implements the formula:

```
color = pow(color, 2.2)
```

If there is a better way to convert linear colorspace into gamma colorspace, do let me know.

I can't use the static API in a Job! You said it supports DOTS!

The workflow is to typically call the static API in an `OnUpdate` loop. Native jobs support will come in **much** later and requires that all calls handle unmanaged memory. Currently all primary windows are managed objects, which is part of the reason why static APIs do not integrate nicely in Jobs specifically.

If you have a workflow in mind, do let me know so I can understand the workflow and so I can figure out a solution to cover various cases.

WebGL is pretty slow, what can I do?

If WebGL is slow and disabling NimGui causes your game to speed up, I would recommend taking a look at the memory settings in `Edit -> Project Settings -> Player Settings -> Publishing Settings`.

Here are the following settings I used to help reduce how much WebGL was allocating and to improve performance:

- Initial Memory Size: 32
- Memory Growth Mode: Linear
- Maximum Memory Size: 2048
- Linear Memory Growth Step: 16

Your mileage may vary and I am having difficulty tracking down what settings are causing web to specifically slow down.

Where do I submit bug reports?

You can submit bug reports here: https://github.com/InitialPrefabs/nimgui/issues. You can also contact **info at initialprefabs dot com** for additional support.

Namespace InitialPrefabs.NimGui

Classes

DefaultStyles

ImButtonStyleExtensions

ImDropDownStyleExtensions

ImGui

ImGuiContext

ImPaneStyleExtensions

ImProgressBarStyleExtensions

ImScrollAreaStyleExtensions

ImSliderStyleExtensions

ImTextFieldStyleExtensions

ImTextStyleExtensions

ImWindow

ImWindow stores rendering information that can't be used in jobs.

StyleExtensions

WindowBehaviorExtensions

Structs

ImButtonStyle

Stores the color states of the button.

ImCollapsibleArea

Convenience struct to conveniently create a collapsible scope.

ImDropDownStyle

Defines the style of the DropDown.

ImLineStyle

Defines the style of the Line.

ImPane

The ImPane is a stack allocated convenience struct to create draggable and collapsible panes.

ImPaneOffset

Stores the position and offset when dragging the pane.

ImPaneStyle

Stores the color states for the Pane.

ImProgressBarStyle

Defines the style of the progress bar.

ImRect

Stores the center and extents of the box.

ImScope

Stores the last known size and position of the scope we intend to layout.

ImScrollArea

ImScrollAreaStyle

Defines the style of the viewable content, the scrollbar, and the scrollbar button.

ImSkipLineStyle

Defines the style of the skipped line.

ImSliderStyle

Defines how the slider button looks and its background.

ImTextFieldStyle

Stores the style of the TextField.

ImTextStyle

Stores the font style.

UnmanagedImWindow

To support Unity Jobs and Burst, this will store only blittable information.

Interfaces

IStyle

A trait for all common style structs for code generation.

Enums

ImDrawCommandType

Describes the type of draw command enqueued.

ImPaneFlags

Describes the state of the Pane.

PruneFlag

The global state to clean up.

Class DefaultStyles

Inheritance

System.Object
DefaultStyles

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class DefaultStyles
```

Fields

Clear

Declaration

```
public static readonly Color32 Clear
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

CloseDefaultFg

Declaration

```
public static readonly Color32 CloseDefaultFg
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

CollapseDefaultFg

Declaration

```
public static readonly Color32 CollapseDefaultFg
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Default

Declaration

```
public static readonly Color32 Default
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

DefaultFontSize

Declaration

```
public const int DefaultFontSize = 18
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Hover

Declaration

```
public static readonly Color32 Hover
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

HoverForeground

Declaration

```
public static readonly Color32 HoverForeground
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Padding

Declaration

```
public static readonly float2 Padding
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Pane

Declaration

```
public static readonly Color32 Pane
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Pressed

Declaration

```
public static readonly Color32 Pressed
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

PressedForeground

Declaration

```
public static readonly Color32 PressedForeground
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

ScrollBarBackground

Declaration

```
public static readonly Color32 ScrollBarBackground
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

ScrollBarPanel

Declaration

```
public static readonly Color32 ScrollBarPanel
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

ScrollBarWidth

Declaration

```
public const float ScrollBarWidth = 15F
```

Field Value

TYPE	DESCRIPTION
System.Single	

SliderBackground

Declaration

```
public static readonly Color32 SliderBackground
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

SliderButtonWidth

Declaration

```
public const float SliderButtonWidth = 20F
```

Field Value

TYPE	DESCRIPTION
System.Single	

Text

Declaration

```
public static readonly Color32 Text
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

TextFieldHover

Declaration

```
public static readonly Color32 TextFieldHover
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

TitleFontSize

Declaration

```
public const int TitleFontSize = 20
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Struct ImButtonStyle

Stores the color states of the button.

Implements

IStyle

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImButtonStyle : IStyle
```

Fields

Background

The default color.

Declaration

```
public Color32 Background
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Column

Column-wise alignment.

Declaration

```
public HorizontalAlignment Column
```

Field Value

TYPE	DESCRIPTION
HorizontalAlignment	

FontSize

Size of the text.

Declaration

```
public int FontSize
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Hover

The color when the mouse is hovered over the button.

Declaration

```
public Color32 Hover
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Padding

The spacing between the button and the next widget.

Declaration

```
public float2 Padding
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Pressed

The color when the mouse is clicked on the button.

Declaration

```
public Color32 Pressed
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Row

Row-wise alignment.

Declaration

```
public VerticalAlignment Row
```

Field Value

TYPE	DESCRIPTION
VerticalAlignment	

Text

The color of the text.

Declaration

```
public Color32 Text
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Methods

New()

Constructs a new instance of the ButtonStyle with default settings.

Declaration

```
public static ImButtonStyle New()
```

Returns

TYPE	DESCRIPTION
ImButtonStyle	

Implements

IStyle

Extension Methods

StyleExtensions.WithBackground(ref ImButtonStyle, Color32)
StyleExtensions.WithHover(ref ImButtonStyle, Color32)
StyleExtensions.WithPressed(ref ImButtonStyle, Color32)
StyleExtensions.WithText(ref ImButtonStyle, Color32)
StyleExtensions.WithFontSize(ref ImButtonStyle, UInt16)
StyleExtensions.WithPadding(ref ImButtonStyle, float2)
StyleExtensions.WithColumn(ref ImButtonStyle, HorizontalAlignment)
StyleExtensions.WithRow(ref ImButtonStyle, VerticalAlignment)
ImButtonStyleExtensions.GetTextStyle(in ImButtonStyle)

Class ImButtonStyleExtensions

Inheritance

System.Object
ImButtonStyleExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class ImButtonStyleExtensions
```

Methods

GetTextStyle(in ImButtonStyle)

Gets the implicit ImTextStyle from the ImButtonStyle.

Declaration

```
public static ImTextStyle GetTextStyle(this in ImButtonStyle buttonStyle)
```

Parameters

TYPE	NAME	DESCRIPTION
ImButtonStyle	buttonStyle	The button style to reference.

Returns

TYPE	DESCRIPTION
ImTextStyle	An ImButtonStyle

Struct ImCollapsibleArea

Convenience struct to conveniently create a collapsible scope.

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public ref struct ImCollapsibleArea
```

Constructors

ImCollapsibleArea(String)

Stack only struct to call BeginCollapsible in a scope similar to ImPane.

Declaration

```
public ImCollapsibleArea(string label)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	The label for the clickable button.

Remarks

This should be used with the Dispose pattern.

Examples

How to use the Dispose pattern.

```
using (ImCollapsibleArea area = new ImCollapsible("Collapsible")) {
    if (area.IsVisible) {
        // Draw more logic here.
    }
}
```

ImCollapsibleArea(String, in ImButtonStyle)

Stack only struct to call BeginCollapsible in a scope similar to ImPane.

Declaration

```
public ImCollapsibleArea(string label, in ImButtonStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	The label for the clickable button.
ImButtonStyle	style	A custom button style for the collapsible area.

Remarks

This should be used with the Dispose pattern.

Examples

How to use the Dispose pattern.

```
using (ImCollapsibleArea area = new ImCollapsible("Collapsible")) {
    if (area.IsVisible) {
        // Draw more logic here.
    }
}
```

ImCollapsibleArea(String, Boolean)

Stack only struct to call BeginCollapsible in a scope similar to ImPane.

Declaration

```
public ImCollapsibleArea(string label, bool isInitiallyCollapsed)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	The StringBuilder containing the label for the clickable button.
System.Boolean	isInitiallyCollapsed	Do you want the area collapsed.

Remarks

This should be used with the Dispose pattern.

Examples

How to use the Dispose pattern.

```
using (ImCollapsibleArea area = new ImCollapsible("Collapsible", true)) {
    if (area.IsVisible) {
        // Draw more logic here.
    }
}
```

ImCollapsibleArea(String, Boolean, in ImButtonStyle)

Stack only struct to call BeginCollapsible in a scope similar to ImPane.

Declaration

```
public ImCollapsibleArea(string label, bool isInitiallyCollapsed, in ImButtonStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	The StringBuilder containing the label for the clickable button.
System.Boolean	isInitiallyCollapsed	Do you want the area collapsed.
ImButtonStyle	style	A custom button style.

Remarks

This should be used with the Dispose pattern.

Examples

How to use the Dispose pattern.

```
using (ImCollapsibleArea area = new ImCollapsible("Collapsible", true)) {
    if (area.IsVisible) {
        // Draw more logic here.
    }
}
```

ImCollapsibleArea(StringBuilder)

Stack only struct to call BeginCollapsible in a scope similar to ImPane.

Declaration

```
public ImCollapsibleArea(StringBuilder builder)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Text.StringBuilder	builder	The builder containing the label for the clickable button.

Remarks

This should be used with the Dispose pattern.

Examples

How to use the Dispose pattern.

```
using (ImCollapsibleArea area = new ImCollapsible("Collapsible")) {
    if (area.IsVisible) {
        // Draw more logic here.
    }
}
```

ImCollapsibleArea(StringBuilder, in ImButtonStyle)

Stack only struct to call BeginCollapsible in a scope similar to ImPane.

Declaration

```
public ImCollapsibleArea(StringBuilder builder, in ImButtonStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Text.StringBuilder	builder	The builder containing the label for the clickable button.
ImButtonStyle	style	A custom button style.

Remarks

This should be used with the Dispose pattern.

Examples

How to use the Dispose pattern.

```
using (ImCollapsibleArea area = new ImCollapsible("Collapsible")) {
    if (area.IsVisible) {
        // Draw more logic here.
    }
}
```

ImCollapsibleArea(StringBuilder, Boolean)

Stack only struct to call BeginCollapsible in a scope similar to ImPane.

Declaration

```
public ImCollapsibleArea(StringBuilder builder, bool isInitiallyCollapsed)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Text.StringBuilder	builder	The builder containing the label for the clickable button.
System.Boolean	isInitiallyCollapsed	Do you want the area collapsed.

Remarks

This should be used with the Dispose pattern.

Examples

How to use the Dispose pattern.

```
using (ImCollapsibleArea area = new ImCollapsible("Collapsible", true)) {
    if (area.IsVisible) {
        // Draw more logic here.
    }
}
```

ImCollapsibleArea(StringBuilder, Boolean, in ImButtonStyle)

Stack only struct to call BeginCollapsible in a scope similar to ImPane.

Declaration

```
public ImCollapsibleArea(StringBuilder builder, bool isInitiallyCollapsed, in ImButtonStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Text.StringBuilder	builder	The builder containing the label for the clickable button.
System.Boolean	isInitiallyCollapsed	Do you want the area collapsed.
ImButtonStyle	style	A custom button style.

Remarks

This should be used with the Dispose pattern.

Examples

How to use the Dispose pattern.

```
using (ImCollapsibleArea area = new ImCollapsible("Collapsible", true)) {
    if (area.IsVisible) {
        // Draw more logic here.
    }
}
```

Fields

IsVisible

Use this property to determine whether or not you should draw widgets.

Declaration

```
public readonly bool IsVisible
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Methods

Dispose()

Declaration

```
public void Dispose()
```

Enum ImDrawCommandType

Describes the type of draw command enqueued.

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public enum ImDrawCommandType
```

Fields

NAME	DESCRIPTION
Image	
Scissor	
Text	

Struct ImDropDownStyle

Defines the style of the DropDown.

Implements

IStyle

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImDropDownStyle : IStyle
```

Fields

Background

The default color.

Declaration

```
public Color32 Background
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Column

Column-wise alignment.

Declaration

```
public HorizontalAlignment Column
```

Field Value

TYPE	DESCRIPTION
HorizontalAlignment	

FontSize

Size of the text.

Declaration

```
public int FontSize
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Hover

The color when the mouse is over the dropdown.

Declaration

```
public Color32 Hover
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Padding

Spacing between the dropdown and the next widget.

Declaration

```
public float2 Padding
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Pressed

The color when the mouse clicks the dropdown.

Declaration

```
public Color32 Pressed
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Row

Row-wise alignment.

Declaration

```
public VerticalAlignment Row
```

Field Value

TYPE	DESCRIPTION
VerticalAlignment	

Text

The color of text.

Declaration

```
public Color32 Text
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Methods

New()

Constructs the default DropdownStyle.

Declaration

```
public static ImDropDownStyle New()
```

Returns

TYPE	DESCRIPTION
ImDropDownStyle	The dropdown style using values from DefaultStyles.

Implements

IStyle

Extension Methods

StyleExtensions.WithBackground(ref ImDropDownStyle, Color32)
StyleExtensions.WithHover(ref ImDropDownStyle, Color32)
StyleExtensions.WithPressed(ref ImDropDownStyle, Color32)
StyleExtensions.WithText(ref ImDropDownStyle, Color32)
StyleExtensions.WithColumn(ref ImDropDownStyle, HorizontalAlignment)
StyleExtensions.WithRow(ref ImDropDownStyle, VerticalAlignment)
StyleExtensions.WithFontSize(ref ImDropDownStyle, UInt16)
StyleExtensions.WithPadding(ref ImDropDownStyle, float2)
ImDropDownStyleExtensions.GetButtonStyle(in ImDropDownStyle)
ImDropDownStyleExtensions.GetTextStyle(in ImDropDownStyle)

Class ImDropDownStyleExtensions

Inheritance

System.Object
ImDropDownStyleExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class ImDropDownStyleExtensions
```

Methods

GetButtonStyle(in ImDropDownStyle)

Gets the associated ImButtonStyle.

Declaration

```
public static ImButtonStyle GetButtonStyle(this in ImDropDownStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
ImDropDownStyle	style	The dropdown style to reference.

Returns

TYPE	DESCRIPTION
ImButtonStyle	Returns the style of the ImButtonStyle.

GetTextStyle(in ImDropDownStyle)

Gets the associated ImTextStyle.

Declaration

```
public static ImTextStyle GetTextStyle(this in ImDropDownStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
ImDropDownStyle	style	The dropdown style to reference.

Returns

TYPE	DESCRIPTION
ImTextStyle	Returns the style of the ImTextStyle.

Class ImGui

Inheritance

System.Object
ImGui

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class ImGui
```

Methods

BeginCollapsible(ImWindow, in ImString, in ImButtonStyle, Boolean, out Boolean)

Creates a collapsible area.

Declaration

```
public static void BeginCollapsible(ImWindow window, in ImString label, in ImButtonStyle style, bool
isInitiallyCollapsed, out bool isCollapsed)
```

Parameters

TYPE	NAME	DESCRIPTION
ImWindow	window	The window to create the collapsible area in.
ImString	label	The content to show in the clickable button.
ImButtonStyle	style	The style of the collapsible area.
System.Boolean	isInitiallyCollapsed	Should the area be collapsed on start?
System.Boolean	isCollapsed	Is the area currently collapsed?

BeginPane(String, in float2, in float2, in ImButtonStyle, in ImPaneStyle, out Boolean, out Boolean, ImPaneFlags)

Begins a draggable pane that defines a scope that can be drawn into.

Declaration

```
public static bool BeginPane(string title, in float2 position, in float2 size, in ImButtonStyle buttonStyle,
in ImPaneStyle paneStyle, out bool collapse, out bool isBackbuffered, ImPaneFlags flags = (ImPaneFlags)0)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	title	Label for the window
Unity.Mathematics.float2	position	The position of where to initially draw
Unity.Mathematics.float2	size	How big the window is
ImButtonStyle	buttonStyle	Buttons colors
ImPaneStyle	paneStyle	Window colors
System.Boolean	collapse	Is the window collapsed?
System.Boolean	isBackbuffered	Is the window queued to a different buffer?
ImPaneFlags	flags	

Returns

TYPE	DESCRIPTION
System.Boolean	

BeginScope(ImRect)

Begins a new scope and ensures so that all widgets drawn are now relative to the scope.

Declaration

```
public static void BeginScope(ImRect rect)
```

Parameters

TYPE	NAME	DESCRIPTION
ImRect	rect	The size and position of the new scope.

BeginScrollArea(ImWindow, String, Single, Single, in ImScrollAreaStyle)

Starts a scroll area given the viewport height and the max height. The scroll area is hardscissored to the region drawn.

Declaration

```
public static void BeginScrollArea(ImWindow window, string title, float viewportHeight, float maxHeight, in
ImScrollAreaStyle scrollAreaStyle)
```

Parameters

TYPE	NAME	DESCRIPTION
ImWindow	window	
System.String	title	The unique identifier of the scroll area.
System.Single	viewportHeight	The height in which we can view. Typically smaller than the maxHeight.
System.Single	maxHeight	The maximum height that the area can store.
ImScrollAreaStyle	scrollAreaStyle	The style of the scroll area.

Box(float2, Color32, Boolean)

Draws a box.

Declaration

```
public static void Box(float2 size, Color32 color, bool updateScope = false)
```

Parameters

TYPE	NAME	DESCRIPTION
Unity.Mathematics.float2	size	How big is the box?
UnityEngine.Color32	color	What color is the box?
System.Boolean	updateScope	If the scope is updated, the next widget will be drawn below the box.

Button(String)

The default Button using a preconfigured Button Style. This registers a click when the Mouse is released.

Declaration

```
public static bool Button(string label)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	The text to display in the button.

Returns

TYPE	DESCRIPTION
System.Boolean	

Button(String, in ImButtonStyle)

Constructs a text button given a label and a button style.

Declaration

```
public static bool Button(string label, in ImButtonStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	The text to display in the button.
ImButtonStyle	style	The style of button.

Returns

TYPE	DESCRIPTION
System.Boolean	

Button(UInt32, String)

Constructs a Button given a control ID and label with the default ButtonStyle.

Declaration

```
public static bool Button(uint controlID, string label)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt32	controlID	A supplied unique ID.
System.String	label	The text to display

Returns

TYPE	DESCRIPTION
System.Boolean	

Button(UInt32, String, ImButtonStyle)

Constructs a Button given a control ID and label

Declaration

```
public static bool Button(uint controlID, string label, ImButtonStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt32	controlID	A supplied unique ID.
System.String	label	The text to display
ImButtonStyle	style	A custom button style

Returns

TYPE	DESCRIPTION
System.Boolean	

CalculateRemainingLineSize(ImWindow, Int32, in float2)

Calcaultes the size of the line based on the scope's next widget position.

Declaration

```
public static float2 CalculateRemainingLineSize(ImWindow window, int fontSize, in float2 padding)
```

Parameters

TYPE	NAME	DESCRIPTION
ImWindow	window	The window to calculate the remaining size.
System.Int32	fontSize	The size of the font.
Unity.Mathematics.float2	padding	The amount of spacing between the current and next widget.

Returns

TYPE	DESCRIPTION
Unity.Mathematics.float2	The size of the rect.

Dropdown(String, String[])

Shows a menu dropdown where you can select a single option. Elements behind the dropdown menu will not be selected.

Declaration

```
public static ushort Dropdown(string label, string[] options)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	A label to provide more context
System.String[]	options	A set number of options

Returns

TYPE	DESCRIPTION
System.UInt16	The index of the element selected

Dropdown(String, String[], in ImDropDownStyle)

Shows a menu dropdown where you can select a single option. Elements behind the dropdown menu will not be selected.

Declaration

```
public static ushort Dropdown(string label, string[] options, in ImDropDownStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	A label to provide more context
System.String[]	options	A set number of options
ImDropDownStyle	style	A style for the dropdown menu

Returns

TYPE	DESCRIPTION
System.UInt16	The index of the element selected

Dropdown(String, UInt16, String[])

Shows a menu dropdown where you can select a single option. Elements behind the dropdown menu will not be selected.

Declaration

```
public static ushort Dropdown(string label, ushort initialIndex, string[] options)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	A label to provide more context
System.UInt16	initialIndex	The initial index of the Dropdown
System.String[]	options	A set number of options

Returns

TYPE	DESCRIPTION
System.UInt16	The index of the element selected

Dropdown(String, UInt16, String[], in ImDropDownStyle)

Shows a menu dropdown where you can select a single option. Elements behind the dropdown menu will not be selected.

Declaration

```
public static ushort Dropdown(string label, ushort initialIndex, string[] options, in ImDropDownStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	A label to provide more context
System.UInt16	initialIndex	The initial index of the Dropdown
System.String[]	options	A set number of options
ImDropDownStyle	style	A style for the dropdown menu

Returns

TYPE	DESCRIPTION
System.UInt16	The index of the element selected

Dropdown(String[])

Shows a menu dropdown where you can select a single option. Elements behind the dropdown menu will not be selected.

Declaration

```
public static ushort Dropdown(string[] options)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String[]	options	A set number of options

Returns

TYPE	DESCRIPTION
System.UInt16	The index of the element selected

Dropdown(String[], in ImDropDownStyle)

Shows a menu dropdown where you can select a single option. Elements behind the dropdown menu will not be selected.

Declaration

```
public static ushort Dropdown(string[] options, in ImDropDownStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String[]	options	A set number of options
ImDropDownStyle	style	A style for the dropdown menu

Returns

TYPE	DESCRIPTION
System.UInt16	The index of the element selected

Dropdown(String[], UInt16)

Shows a menu dropdown where you can select a single option. Elements behind the dropdown menu will not be selected.

Declaration

```
public static ushort Dropdown(string[] options, ushort initialIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String[]	options	A set number of options
System.UInt16	initialIndex	

Returns

TYPE	DESCRIPTION
System.UInt16	The index of the element selected

Dropdown(String[], UInt16, in ImDropDownStyle)

Shows a menu dropdown where you can select a single option. Elements behind the dropdown menu will not be selected.

Declaration

```
public static ushort Dropdown(string[] options, ushort initialIndex, in ImDropDownStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String[]	options	A set number of options
System.UInt16	initialIndex	
ImDropDownStyle	style	A style for the dropdown menu

Returns

TYPE	DESCRIPTION
System.UInt16	The index of the element selected

EndPane(ImWindow, Boolean, Boolean)

Ends the Pane so the next widget can be correctly drawn.

Declaration

```
public static void EndPane(ImWindow window, bool pop, bool autoLayout = false)
```

Parameters

TYPE	NAME	DESCRIPTION
ImWindow	window	
System.Boolean	pop	Popping ensures that the next widget drawn is not backbuffered
System.Boolean	autoLayout	If you need to update the layout, for free floating panes, you don't need to

EndScope(Boolean)

Ends the previous scope and updates the layout engine.

Declaration

```
public static void EndScope(bool autoLayout = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	autoLayout	

EndScrollArea(Boolean)

Ends the scroll area and disables hardware scissoring.

Declaration

```
public static void EndScrollArea(bool updateLayout = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	updateLayout	Should the layout be updated? Typically, yes.

Label(Int32)

Creates a label from an integer.

Declaration

```
public static void Label(int value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	The integer to display.

Label(Int32, in ImTextStyle)

Creates a label from an integer.

Declaration

```
public static void Label(int value, in ImTextStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	The integer to display.
ImTextStyle	style	The style of the text.

Label(Single)

Creates a label from a floating point value.

Declaration

```
public static void Label(float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	The floating point to display.

Label(Single, in ImTextStyle)

Creates a label from a floating point value with a custom style.

Declaration

```
public static void Label(float value, in ImTextStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	The floating point to display.
ImTextStyle	style	The style of the text.

Label(String)

Creates a label to display some text with the default style.

Declaration

```
public static void Label(string label)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	Text to display

Label(String, in ImTextStyle)

Creates a label to display some text with the default style.

Declaration

```
public static void Label(string label, in ImTextStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.String	label	Text to display
ImTextStyle	style	The style of the text.

Label(String, in float2)

Creates a label to display text with a custom size, but using the default style.

Declaration

```
public static void Label(string label, in float2 size)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	Text to display
Unity.Mathematics.float2	size	The size of the label's area

Label(String, in float2, in ImTextStyle)

Creates a label to display text with a custom style.

Declaration

```
public static void Label(string label, in float2 size, in ImTextStyle textStyle)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	Text to display.
Unity.Mathematics.float2	size	The size of the label's area.
ImTextStyle	textStyle	The style of the text.

Label(StringBuilder)

Creates a label to display some text using a StringBuilder with the default style.

Declaration

```
public static void Label(StringBuilder builder)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Text.StringBuilder	builder	The StringBuilder containing the text.

Label(StringBuilder, in ImTextStyle)

Creates a label using a StringBuilder to display some text with the default style.

Declaration

```
public static void Label(StringBuilder builder, in ImTextStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Text.StringBuilder	builder	The StringBuilder containing the label.
ImTextStyle	style	The style of the text.

Label(StringBuilder, float2, in ImTextStyle)

Create a label to display text using a StringBuilder and a custom style.

Declaration

```
public static void Label(StringBuilder builder, float2 size, in ImTextStyle textStyle)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Text.StringBuilder	builder	The StringBuilder containing the label.
Unity.Mathematics.float2	size	The size of the label's area.
ImTextStyle	textStyle	The style of the text.

Line()

Draws a line using the default style.

Declaration

```
public static void Line()
```

Line(ImLineStyle)

Draws a line given a custom style.

Declaration

```
public static void Line(ImLineStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
ImLineStyle	style	The style of the line.

ProgressBar(Int32, Int32)

Creates a ProgressBar and displays the ratio as a fraction with a numerator and denominator using the default style. The width and height is based off of the last known scope and the size of the text.

Declaration

```
public static void ProgressBar(int num, int den)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	num	Numerator of the fraction
System.Int32	den	Denominator of the fraction

ProgressBar(Int32, Int32, in ImProgressBarStyle)

Creates a ProgressBar and displays the ratio as a fraction with a numerator and denominator using a custom style. The width and height is based off of the last known scope and the size of the text.

Declaration

```
public static void ProgressBar(int num, int den, in ImProgressBarStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	num	Numerator of the fraction
System.Int32	den	Denominator of the fraction
ImProgressBarStyle	style	A custom style

ProgressBar(Int32, Int32, float2)

Creates a ProgressBar and displays the ratio as a fraction with a numerator and denominator using the default style.

Declaration

```
public static void ProgressBar(int num, int den, float2 size)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	num	Numerator of the fraction
System.Int32	den	Denominator of the fraction
Unity.Mathematics.float2	size	A custom size to define how large the progress bar is.

ProgressBar(Int32, Int32, float2, in ImProgressBarStyle)

Creates a ProgressBar and displays the ratio as a fraction with a numerator and denominator using a custom style.

Declaration

```
public static void ProgressBar(int num, int den, float2 size, in ImProgressBarStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	num	Numerator of the fraction
System.Int32	den	Denominator of the fraction
Unity.Mathematics.float2	size	A custom size to define how large the progress bar is.
ImProgressBarStyle	style	A custom style

ProgressBar(Single)

Creates a ProgressBar given a size using default styling. The size is determined by the line height and the width is determine by the parent scope.

Declaration

```
public static void ProgressBar(float ratio)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	ratio	A value between 0 and 1

ProgressBar(Single, Single)

Creates a ProgressBar given a width using the default styling.

Declaration

```
public static void ProgressBar(float ratio, float width)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	ratio	A value between 0 and 1
System.Single	width	Max width of the rectangle containing the progress bar

ProgressBar(Single, Single, in ImProgressBarStyle)

Creates a ProgressBar given a width using custom styling.

Declaration

```
public static void ProgressBar(float ratio, float width, in ImProgressBarStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	ratio	A value between 0 and 1
System.Single	width	Max width of the rectangle containing the progress bar
ImProgressBarStyle	style	Custom style

ProgressBar(Single, float2)

Creates a ProgressBar given a size using the default styling.

Declaration

```
public static void ProgressBar(float ratio, float2 size)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	ratio	A value between 0 and 1
Unity.Mathematics.float2	size	Size along the x and y axis

ProgressBar(Single, float2, in ImProgressBarStyle)

Creates a ProgressBar given a size using custom styling.

Declaration

```
public static void ProgressBar(float ratio, float2 size, in ImProgressBarStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	ratio	A value between 0 and 1
Unity.Mathematics.float2	size	Size along the x and y axis
ImProgressBarStyle	style	A custom style

Prune(String, PruneFlag)

Removes any id that has been internally cached. This includes Collapsed elements, Dropdowns, Pane offsets, Scroll offsets, and Toggled states.

```
string title = "Title";

using (ImPane = new ImPane(title, ...)) {
    ...
}

ImGui.Prune(title, PruneFlag.Prune);
```

Declaration

```
public static void Prune(string title, PruneFlag flags = PruneFlag.All)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	title	A unique title.
PruneFlag	flags	The internal state to clean up, by default we clean everything.

Prune(String[], PruneFlag)

Removes any id that has been internally cached. This includes Collapsed elements, Dropdowns, Pane offsets, Scroll offsets, and Toggled states.

```
string title = "Title";

using (ImPane = new ImPane(title, ...)) {
    ...
}

ImGui.Prune(title, PruneFlag.Prune);
```

Declaration

```
public static void Prune(string[] titles, PruneFlag flags = PruneFlag.All)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String[]	titles	An array of unique labels.
PruneFlag	flags	The internal state to clean up, by default we clean everything.

Prune(UInt32, PruneFlag)

Removes any id that has been internally cached. This includes Collapsed elements, Dropdowns, Pane offsets, Scroll offsets, and Toggled states.

```
int controlID = 1;

using (ImPane pane = new ImPane(controlID, ...)) {
    ...
}

ImGui.Prune(controlID, PruneFlag.Pane);
```

Declaration

```
public static void Prune(uint id, PruneFlag flags = PruneFlag.All)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt32	id	The unique ID to remove.
PruneFlag	flags	The internal state to clean up, by default we clean everything.

Prune(UInt32[], PruneFlag)

Removes any id that has been internally cached. This includes Collapsed elements, Dropdowns, Pane offsets, Scroll offsets, and Toggled states.

```
string title = "Title";

using (ImPane = new ImPane(title, ...)) {
    ...
}

ImGui.Prune(title, PruneFlag.Prune);
```

Declaration

```
public static void Prune(uint[] ids, PruneFlag flags = PruneFlag.All)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt32[]	ids	An array of ids.
PruneFlag	flags	The internal state to clean up, by default we clean everything.

SameLine()

Ensures that the next widget is drawn on the same line instead of the next line.

```
------------   ------------
| Widget 1 |   | Widget 2 |
------------   ------------
The second widget will be drawn on the same line after Widget 1.
```

Declaration

```
public static void SameLine()
```

SkipLine()

Instead of drawing the widget on the next line, the next line is skipped and the next widget is drawn on that succeeding line.

```
------------
| Widget 1 |
------------
            ImGui.SkipLine() will create an empty line here
------------
| Widget 2 |
------------
```

Declaration

```
public static void SkipLine()
```

SkipLine(in ImSkipLineStyle)

Instead of drawing the widget on the next line, the next line is skipped and the next widget is drawn on that succeeding line.

```
------------
| Widget 1 |
------------
            ImGui.SkipLine() will create an empty line here
------------
| Widget 2 |
------------
```

Declaration

```
public static void SkipLine(in ImSkipLineStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
ImSkipLineStyle	style	A custom style for the skipped line.

Slider(Int32, Int32, in ImSliderStyle, Single)

Returns a value between two integers.

Declaration

```
public static int Slider(int min, int max, in ImSliderStyle style, float t = 0F)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	min	The minimum value, must be less than the min
System.Int32	max	The maximum value, must be greater than the min
ImSliderStyle	style	A custom slider style.
System.Single	t	The initial value between 0 and 1 that describes the value between the min and max.

Returns

TYPE	DESCRIPTION
System.Int32	

Slider(Int32, Int32, Single)

Returns a value between two integers. Uses the default style. ImSliderStyle

Declaration

```
public static int Slider(int min, int max, float t = 0F)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	min	The minimum value, must be less than the min
System.Int32	max	The maximum value, must be greater than the min
System.Single	t	The initial value between 0 and 1 that describes the value between the min and max.

Returns

TYPE	DESCRIPTION
System.Int32	

Slider(Single, Single, in ImSliderStyle, Single)

Returns a value between two floating points.

Declaration

```
public static float Slider(float min, float max, in ImSliderStyle style, float t = 0F)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	min	The minimum value, must be less than the min
System.Single	max	The maximum value, must be greater than the min
ImSliderStyle	style	A custom slider style.
System.Single	t	The initial value between 0 and 1 that describes the value between the min and max.

Returns

TYPE	DESCRIPTION
System.Single	

Slider(Single, Single, Single)

Returns a value between two floating points. Uses the default style. ImSliderStyle

Declaration

```
public static float Slider(float min, float max, float t = 0F)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	min	The minimum value, must be less than the min
System.Single	max	The maximum value, must be greater than the min
System.Single	t	The initial value between 0 and 1 that describes the value between the min and max.

Returns

TYPE	DESCRIPTION
System.Single	

Slider(String, Int32, Int32, in ImSliderStyle)

Returns a value between two integers.

Declaration

```
public static int Slider(string label, int min, int max, in ImSliderStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	
System.Int32	min	The minimum value, must be less than the min.
System.Int32	max	The maximum value, must be greater than the min.
ImSliderStyle	style	A custom slider style.

Returns

TYPE	DESCRIPTION
System.Int32	

Slider(String, Int32, Int32, Single)

Returns a value between two integers.

Declaration

```
public static int Slider(string label, int min, int max, float t = 0F)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	A label to provide more context
System.Int32	min	The minimum value, must be less than the min
System.Int32	max	The maximum value, must be greater than the min
System.Single	t	The initial value between 0 and 1 that describes the value between the min and max.

Returns

TYPE	DESCRIPTION
System.Int32	

Slider(String, Single, Single, in ImSliderStyle, Single)

Returns a value between two floating points.

Declaration

```
public static float Slider(string label, float min, float max, in ImSliderStyle style, float t = 0F)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	A label to provide more context of the slider.
System.Single	min	The minimum value, must be less than the min
System.Single	max	The maximum value, must be greater than the min
ImSliderStyle	style	A custom slider style
System.Single	t	The initial value between 0 and 1 that describes the value between the min and max.

Returns

TYPE	DESCRIPTION
System.Single	

Slider(String, Single, Single, Single)

Returns a value between two floating points.

Declaration

```
public static float Slider(string label, float min, float max, float t = 0F)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	A label to provide more context
System.Single	min	The minimum value, must be less than the min

TYPE	NAME	DESCRIPTION
System.Single	max	The maximum value, must be greater than the min
System.Single	t	The initial value between 0 and 1 that describes the value between the min and max.

Returns

TYPE	DESCRIPTION
System.Single	

TextField(String, StringBuilder)

Fills a StringBuilder with inputs given from the keyboard.

Declaration

```
public static void TextField(string label, StringBuilder builder)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	The label to describe the textfield's purpose
System.Text.StringBuilder	builder	The StringBuilder which will be filled by the TextField.

TextField(String, StringBuilder, in ImTextFieldStyle)

Fills a StringBuilder with inputs given from the keyboard.

Declaration

```
public static void TextField(string label, StringBuilder builder, in ImTextFieldStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	The label to describe the textfield's purpose
System.Text.StringBuilder	builder	The StringBuilder which will be filled by the TextField.
ImTextFieldStyle	style	The style of the TextField.

Toggle(String, in ImButtonStyle, Boolean)

Returns the current state of the Toggle box. If the box is checked, then the state is true, otherwise returns false.

Unlike Toggle(out UInt32, Boolean), the control ID is determined by using GetStringHash(String).

```
class UIBehaviour {
    // Gets called once per frame.
    void Update() {
        ImGui.Toggle("Label");
    }

    ~UIBehaviour() {
        ImGui.PruneToggle(TextUtils.GetStringHash("Label"), PruneFlags.Toggle);
    }
}
```

Declaration

```
public static bool Toggle(string label, in ImButtonStyle style, bool initial = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	A label to provide context for the toggle box
ImButtonStyle	style	A custom style for the button.
System.Boolean	initial	The initial value of the Toggle.

Returns

TYPE	DESCRIPTION
System.Boolean	True, if checked, false if not.

Toggle(String, Boolean)

Returns the current state of the Toggle box. If the box is checked, then the state is true, otherwise returns false.

Unlike Toggle(out UInt32, Boolean), the control ID is determined by using GetStringHash(String).

```
class UIBehaviour {
    // Gets called once per frame.
    void Update() {
        ImGui.Toggle("Label");
    }

    ~UIBehaviour() {
        ImGui.PruneToggle(TextUtils.GetStringHash("Label"), PruneFlags.Toggle);
    }
}
```

Declaration

```
public static bool Toggle(string label, bool initial = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	label	A label to provide context for the toggle box
System.Boolean	initial	The initial value of the Toggle.

Returns

TYPE	DESCRIPTION
System.Boolean	True, if checked, false if not.

Toggle(out UInt32, in ImButtonStyle, Boolean)

Returns the current state of the Toggle box. If the box is checked, then the state is true, otherwise returns false.

The control ID is useful for pruning cached global states. You only need to prune when you are no longer using the Toggle functionality.

```
class UIBehaviour {
    uint toggleID;
    // Gets called once per frame.
    void Update() {
        if (ImGui.Toggle(out toggleID)) {
            ...
        }
    }

    ~UIBehaviour() {
        // Internally this will remove the cached Toggle state from
        // it's internal hashmap, when this Object is destroyed.
        ImGui.PruneToggle(toggleID, PruneFlags.Toggle);
    }
}
```

Declaration

```
public static bool Toggle(out uint controlID, in ImButtonStyle style, bool initial = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt32	controlID	The value of the ID that NimGui generated for this Toggle
ImButtonStyle	style	A custom style for the toggle box.
System.Boolean	initial	The initial state of the Toggle.

Returns

TYPE	DESCRIPTION
System.Boolean	True, if checked, false if not.

Toggle(out UInt32, Boolean)

Returns the current state of the Toggle box. If the box is checked, then the state is true, otherwise returns false.

The control ID is useful for pruning cached global states. You only need to prune when you are no longer using the Toggle functionality.

```
class UIBehaviour {
    uint toggleID;
    // Gets called once per frame.
    void Update() {
        if (ImGui.Toggle(out toggleID)) {
            ...
        }
    }

    ~UIBehaviour() {
        // Internally this will remove the cached Toggle state from
        // it's internal hashmap, when this Object is destroyed.
        ImGui.Prune(toggleID, PruneFlag.Toggle);
    }
}
```

Declaration

```
public static bool Toggle(out uint controlID, bool initial = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt32	controlID	The value of the ID that NimGui generated for this Toggle
System.Boolean	initial	The initial state of the Toggle.

Returns

TYPE	DESCRIPTION
System.Boolean	True, if checked, false if not.

Class ImGuiContext

Inheritance

System.Object
ImGuiContext

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class ImGuiContext
```

Methods

All()

If for some reason we are tracking multiple windows, this allows you to reference and iterate through windows.

Declaration

```
public static ReadOnlyCollection<ImWindow> All()
```

Returns

TYPE	DESCRIPTION
InitialPrefabs.NimGui.Collections.ReadOnlyCollection<ImWindow>	

GetCurrentWindow()

Returns the first Window stored.

Declaration

```
public static ImWindow GetCurrentWindow()
```

Returns

TYPE	DESCRIPTION
ImWindow	

Struct ImLineStyle

Defines the style of the Line.

Implements

IStyle

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImLineStyle : IStyle
```

Fields

Color

The foreground color.

Declaration

```
public Color32 Color
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Padding

The amount of spacing along the x axis. Unlike most styles, this does not care about the the Y axis.

Declaration

```
public float Padding
```

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

New()

Constructs the default style of the Line using the DefaultStyles config.

Declaration

```
public static ImLineStyle New()
```

Returns

TYPE	DESCRIPTION
ImLineStyle	

Implements

IStyle

Extension Methods

StyleExtensions.WithColor(ref ImLineStyle, Color32)
StyleExtensions.WithPadding(ref ImLineStyle, Single)

Struct ImPane

The ImPane is a stack allocated convenience struct to create draggable and collapsible panes.

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public ref struct ImPane
```

Constructors

ImPane(String, float2, float2, ImPaneFlags, Boolean)

Disposable stack-only struct to conveniently call BeginPane and EndPane. This uses the default color schemes. How to use the dispose pattern

```
using (var pane = new ImPane(label, position, size)) {
    // Write your pane logic here
}
```

Declaration

```
public ImPane(string title, float2 position, float2 size, ImPaneFlags flags = (ImPaneFlags)0, bool autoLayout
= false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	title	The title of the pane
Unity.Mathematics.float2	position	The initial position of the pane
Unity.Mathematics.float2	size	The size of the pane
ImPaneFlags	flags	Any flags which define the initial behavior of the Pane
System.Boolean	autoLayout	Optionally, you can force the pane to automically layout the window

ImPane(String, float2, float2, in ImPaneStyle, in ImButtonStyle, ImPaneFlags, Boolean)

Disposable stack-only struct to conveniently call BeginPane and EndPane. How to use the dispose pattern

```
using (var pane = new ImPane(label, position, size, windowStyle, buttonStyle)) {
    // Write your pane logic here
}
```

Declaration

```
public ImPane(string title, float2 position, float2 size, in ImPaneStyle paneStyle, in ImButtonStyle
buttonStyle, ImPaneFlags flags = (ImPaneFlags)0, bool autoLayout = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	title	The title of the pane
Unity.Mathematics.float2	position	The initial position of the pane
Unity.Mathematics.float2	size	The size of the pane
ImPaneStyle	paneStyle	The style of the pane
ImButtonStyle	buttonStyle	The style of the button
ImPaneFlags	flags	Any flags which define the initial behavior of the Pane
System.Boolean	autoLayout	Optionally, you can force the pane to automically layout the window

ImPane(UInt32, String, float2, float2, ImPaneFlags, in ImPaneStyle, in ImButtonStyle, Boolean)

Disposable stack-only struct to conveniently call BeginPane and EndPane. You can supply a unique id instead of the BeginPane method hashing the label.

How to use the dispose pattern

```
using (var pane = new ImPane(label, position, size)) {
    // Write your pane logic here
}
```

Declaration

```
public ImPane(uint controlID, string title, float2 position, float2 size, ImPaneFlags flags, in ImPaneStyle
paneStyle, in ImButtonStyle buttonStyle, bool autoLayout = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt32	controlID	A unique ID for the pane

TYPE	NAME	DESCRIPTION
System.String	title	The title of the pane
Unity.Mathematics.float2	position	The initial position of the pane
Unity.Mathematics.float2	size	The size of the pane
ImPaneFlags	flags	
ImPaneStyle	paneStyle	The style of the pane
ImButtonStyle	buttonStyle	The style of the button
System.Boolean	autoLayout	Optionally, you can force the pane to automically layout the window

Fields

IsVisible

Internally checks if the pane should show. This property is set using the out parameter of ImGui.BeginPane(...) method.

Declaration

```
public readonly bool IsVisible
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Methods

Dispose()

Declaration

```
public void Dispose()
```

Enum ImPaneFlags

Describes the state of the Pane.

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public enum ImPaneFlags
```

Fields

NAME	DESCRIPTION
Closed	
Collapsed	
Pinned	

Struct ImPaneOffset

Stores the position and offset when dragging the pane.

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImPaneOffset
```

Fields

Offset

Declaration

```
public float2 Offset
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Position

Declaration

```
public float2 Position
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Struct ImPaneStyle

Stores the color states for the Pane.

Implements

IStyle

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImPaneStyle : IStyle
```

Fields

CloseDefaultFg

Default close button color.

Declaration

```
public Color32 CloseDefaultFg
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

CloseHoverFg

Close button color when the mouse is over it.

Declaration

```
public Color32 CloseHoverFg
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

ClosePressedFg

Close pressed color when the mouse clicks it.

Declaration

```
public Color32 ClosePressedFg
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

CollapseDefaultFg

Default collapse button color.

Declaration

```
public Color32 CollapseDefaultFg
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

CollapseHoverFg

Collapse button color when the mouse is over it.

Declaration

```
public Color32 CollapseHoverFg
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

CollapsePressedFg

Collapse button color when the mouse clicks it.

Declaration

```
public Color32 CollapsePressedFg
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Column

Column wise alignment.

Declaration

```
public HorizontalAlignment Column
```

Field Value

TYPE	DESCRIPTION
HorizontalAlignment	

DefaultButtonBackground

Default color of the button.

Declaration

```
public Color32 DefaultButtonBackground
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

DefaultButtonHover

Color of the button when the mouse is over it.

Declaration

```
public Color32 DefaultButtonHover
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

DefaultButtonPress

Color of the button when the mouse clicks it.

Declaration

```
public Color32 DefaultButtonPress
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

DefaultFontSize

The default size of the font.

Declaration

```
public int DefaultFontSize
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Padding

Amount of spacing between the current and next widget.

Declaration

```
public float2 Padding
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Pane

Background color.

Declaration

```
public Color32 Pane
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Row

Row wise alignment.

Declaration

```
public VerticalAlignment Row
```

Field Value

TYPE	DESCRIPTION
VerticalAlignment	

Text

Text color.

Declaration

```
public Color32 Text
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

TitleBar

Color of the top bar.

Declaration

```
public Color32 TitleBar
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

TitleFontSize

The size of the font for the title (top bar).

Declaration

```
public int TitleFontSize
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

New()

Constructs a new instance of the PaneStyle with default settings.

Declaration

```
public static ImPaneStyle New()
```

Returns

TYPE	DESCRIPTION
ImPaneStyle	

Implements

IStyle

Extension Methods

ImPaneStyleExtensions.GetButtonStyle(in ImPaneStyle)
ImPaneStyleExtensions.GetTextStyle(in ImPaneStyle)
ImPaneStyleExtensions.WithButtonStyle(ref ImPaneStyle, in ImButtonStyle)
StyleExtensions.WithTitleBar(ref ImPaneStyle, Color32)
StyleExtensions.WithText(ref ImPaneStyle, Color32)
StyleExtensions.WithPane(ref ImPaneStyle, Color32)
StyleExtensions.WithDefaultButtonBackground(ref ImPaneStyle, Color32)
StyleExtensions.WithDefaultButtonHover(ref ImPaneStyle, Color32)
StyleExtensions.WithDefaultButtonPress(ref ImPaneStyle, Color32)
StyleExtensions.WithCollapseDefaultFg(ref ImPaneStyle, Color32)
StyleExtensions.WithCollapseHoverFg(ref ImPaneStyle, Color32)
StyleExtensions.WithCollapsePressedFg(ref ImPaneStyle, Color32)
StyleExtensions.WithCloseDefaultFg(ref ImPaneStyle, Color32)
StyleExtensions.WithCloseHoverFg(ref ImPaneStyle, Color32)
StyleExtensions.WithClosePressedFg(ref ImPaneStyle, Color32)
StyleExtensions.WithPadding(ref ImPaneStyle, float2)
StyleExtensions.WithDefaultFontSize(ref ImPaneStyle, UInt16)
StyleExtensions.WithTitleFontSize(ref ImPaneStyle, UInt16)

StyleExtensions.WithColumn(ref ImPaneStyle, HorizontalAlignment)
StyleExtensions.WithRow(ref ImPaneStyle, VerticalAlignment)

Class ImPaneStyleExtensions

Inheritance

System.Object
ImPaneStyleExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class ImPaneStyleExtensions
```

Methods

GetButtonStyle(in ImPaneStyle)

Gets the implicit ImButtonStyle from the ImPaneStyle.

Declaration

```
public static ImButtonStyle GetButtonStyle(this in ImPaneStyle pane)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	pane	The pane style to reference.

Returns

TYPE	DESCRIPTION
ImButtonStyle	An instance of the ImButtonStyle

GetTextStyle(in ImPaneStyle)

Gets the implicit ImTextStyle from the ImPaneStyle.

Declaration

```
public static ImTextStyle GetTextStyle(this in ImPaneStyle pane)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
ImPaneStyle	pane	The pane style to reference.

Returns

TYPE	DESCRIPTION
ImTextStyle	An instance of the ImTextStyle.

WithButtonStyle(ref ImPaneStyle, in ImButtonStyle)

Sets the implicit ImButtonStyle in the ImPaneStyle.

Declaration

```
public static ref ImPaneStyle WithButtonStyle(this ref ImPaneStyle pane, in ImButtonStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	pane	The pane style to reference.
ImButtonStyle	style	The desired ImButtonStyle.

Returns

TYPE	DESCRIPTION
ImPaneStyle	An instance of the ImPaneStyle.

Struct ImProgressBarStyle

Defines the style of the progress bar.

Implements

IStyle

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImProgressBarStyle : IStyle
```

Fields

Background

Declaration

```
public Color32 Background
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Column

Declaration

```
public HorizontalAlignment Column
```

Field Value

TYPE	DESCRIPTION
HorizontalAlignment	

FontSize

Declaration

```
public int FontSize
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Foreground

Declaration

```
public Color32 Foreground
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Row

Declaration

```
public VerticalAlignment Row
```

Field Value

TYPE	DESCRIPTION
VerticalAlignment	

TextColor

Declaration

```
public Color32 TextColor
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Methods

New()

Constructs the default style of the ProgressBar using the DefaultStyles config.

Declaration

```
public static ImProgressBarStyle New()
```

Returns

TYPE	DESCRIPTION
ImProgressBarStyle	

Implements

IStyle

Extension Methods

ImProgressBarStyleExtensions.GetTextStyle(in ImProgressBarStyle)
StyleExtensions.WithBackground(ref ImProgressBarStyle, Color32)
StyleExtensions.WithForeground(ref ImProgressBarStyle, Color32)
StyleExtensions.WithTextColor(ref ImProgressBarStyle, Color32)
StyleExtensions.WithFontSize(ref ImProgressBarStyle, UInt16)
StyleExtensions.WithColumn(ref ImProgressBarStyle, HorizontalAlignment)

StyleExtensions.WithRow(ref ImProgressBarStyle, VerticalAlignment)

Class ImProgressBarStyleExtensions

Inheritance

System.Object
ImProgressBarStyleExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class ImProgressBarStyleExtensions
```

Methods

GetTextStyle(in ImProgressBarStyle)

Gets the implicit text style from the ImProgressBarStyle.

Declaration

```
public static ImTextStyle GetTextStyle(this in ImProgressBarStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
ImProgressBarStyle	style	The progress bar's style.

Returns

TYPE	DESCRIPTION
ImTextStyle	ImTextStyle

Struct ImRect

Stores the center and extents of the box.

Implements

System.IEquatable<ImRect>

Inherited Members

System.ValueType.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImRect : IEquatable<ImRect>
```

Constructors

ImRect(float2, float2)

Creates a rectangle given the center and extents.

Declaration

```
public ImRect(float2 center, float2 extents)
```

Parameters

TYPE	NAME	DESCRIPTION
Unity.Mathematics.float2	center	The position of the rectangle.
Unity.Mathematics.float2	extents	The extents is the width and height of the rectangle from the center.

Fields

Extents

The extents is the width and height of the rectangle from the center.

Declaration

```
public float2 Extents
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Position

Typically, the center of the rectangle.

Declaration

```
public float2 Position
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Properties

Max

The top right corner of the rectangle.

Declaration

```
public readonly float2 Max { get; }
```

Property Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	This is the position added by the extents.

Min

The bottom left corner of the rectangle.

Declaration

```
public readonly float2 Min { get; }
```

Property Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	This is the position subtracted by the extents.

Size

The size of the rectangle.

Declaration

```
public readonly float2 Size { get; }
```

Property Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	A multiplication of the Extents by 2

Methods

Clamp(in float4)

Clamps the current bounds of a rectangle to this rectangle's Min and Max points.

Declaration

```
public float4 Clamp(in float4 bounds)
```

Parameters

TYPE	NAME	DESCRIPTION
Unity.Mathematics.float4	bounds	A float4 where xy is the min and zw is the max.

Returns

TYPE	DESCRIPTION
Unity.Mathematics.float4	The clamped values in relation to this rectangle.

Contains(float2)

Checks if the point is wihtin the rectangle.

Declaration

```
public bool Contains(float2 point)
```

Parameters

TYPE	NAME	DESCRIPTION
Unity.Mathematics.float2	point	The point to check.

Returns

TYPE	DESCRIPTION
System.Boolean	True, if inside the rectangle.

Equals(ImRect)

Declaration

```
public bool Equals(ImRect other)
```

Parameters

TYPE	NAME	DESCRIPTION
ImRect	other	

Returns

TYPE	DESCRIPTION
System.Boolean	

GetHashCode()

Declaration

```
public override int GetHashCode()
```

Returns

TYPE	DESCRIPTION
System.Int32	

Overrides

System.ValueType.GetHashCode()

ToString()

Declaration

```
public override string ToString()
```

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.ValueType.ToString()

Operators

Implicit(ImRect to Rect)

Declaration

```
public static implicit operator Rect(ImRect rect)
```

Parameters

TYPE	NAME	DESCRIPTION
ImRect	rect	

Returns

TYPE	DESCRIPTION
UnityEngine.Rect	

Implements

System.IEquatable<T>

Struct ImScope

Stores the last known size and position of the scope we intend to layout.

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImScope
```

Fields

Delta

Stores the added size.

Declaration

```
public float2 Delta
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Next

Stores the next position for the next element. The position is stored in screen space.

Declaration

```
public float2 Next
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Previous

Stores the last known position.

Declaration

```
public float2 Previous
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Rect

Stores the bounding box.

Declaration

```
public ImRect Rect
```

Field Value

TYPE	DESCRIPTION
ImRect	

Methods

Create(in ImRect)

Creates a scope where the Next position is upper left.

Declaration

```
public static ImScope Create(in ImRect rect)
```

Parameters

TYPE	NAME	DESCRIPTION
ImRect	rect	The bounds of the scope

Returns

TYPE	DESCRIPTION
ImScope	A scope with the metadata generated

Struct ImScrollArea

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public ref struct ImScrollArea
```

Constructors

ImScrollArea(String, Boolean)

Declaration

```
public ImScrollArea(string name, bool autoLayout = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	
System.Boolean	autoLayout	

ImScrollArea(String, Single, Single, Boolean)

Convenience struct to call ImGui.BeginScrollArea. Using the dispose pattern also calls ImGui.EndScrollArea.

```
using (var scrollArea = new ImScrollArea("Title", viewportHeight, maxHeight)) {
    // Implement logic you would like to draw within the scroll area.
}
```

Declaration

```
public ImScrollArea(string name, float viewportHeight, float maxHeight, bool autoLayout = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	The name of the scroll area.
System.Single	viewportHeight	How much space should be added to he scroll area?
System.Single	maxHeight	What is the maximum height of the scroll area?

TYPE	NAME	DESCRIPTION
System.Boolean	autoLayout	Should the scroll area be laid out automatically? Generally, yes.

Methods

Dispose()

Declaration

```
public void Dispose()
```

Struct ImScrollAreaStyle

Defines the style of the viewable content, the scrollbar, and the scrollbar button.

Implements

IStyle

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImScrollAreaStyle : IStyle
```

Fields

ButtonDefault

Default scroll button color.

Declaration

```
public Color32 ButtonDefault
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

ButtonHover

Scroll button color when the mouse is over the scroll button.

Declaration

```
public Color32 ButtonHover
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

ButtonPressed

Scroll button color when the mouse presses the button.

Declaration

```
public Color32 ButtonPressed
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

DeltaTime

Time between frames.

Declaration

```
public float DeltaTime
```

Field Value

TYPE	DESCRIPTION
System.Single	

Padding

Amount of spacing between the current and next widget.

Declaration

```
public float2 Padding
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

ScrollBarBackground

Background color of the scrollbar.

Declaration

```
public Color32 ScrollBarBackground
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

ScrollBarPanel

Panel color of the scroll area.

Declaration

```
public Color32 ScrollBarPanel
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

ScrollButtonWidth

Size of the scroll button.

Declaration

```
public float ScrollButtonWidth
```

Field Value

TYPE	DESCRIPTION
System.Single	

ScrollSpeed

Speed of scrolling with a mousewheel.

Declaration

```
public float ScrollSpeed
```

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

New()

Constructs a new instance of the ScrollAreaStyle with default settings.

Declaration

```
public static ImScrollAreaStyle New()
```

Returns

TYPE	DESCRIPTION
ImScrollAreaStyle	

Implements

IStyle

Extension Methods

ImScrollAreaStyleExtensions.GetButtonStyle(in ImScrollAreaStyle)
ImScrollAreaStyleExtensions.WithButtonStyle(ref ImScrollAreaStyle, in ImButtonStyle)
StyleExtensions.WithScrollButtonWidth(ref ImScrollAreaStyle, Single)
StyleExtensions.WithPadding(ref ImScrollAreaStyle, float2)
StyleExtensions.WithButtonDefault(ref ImScrollAreaStyle, Color32)
StyleExtensions.WithButtonHover(ref ImScrollAreaStyle, Color32)
StyleExtensions.WithButtonPressed(ref ImScrollAreaStyle, Color32)
StyleExtensions.WithScrollBarBackground(ref ImScrollAreaStyle, Color32)
StyleExtensions.WithScrollBarPanel(ref ImScrollAreaStyle, Color32)
StyleExtensions.WithDeltaTime(ref ImScrollAreaStyle, Single)
StyleExtensions.WithScrollSpeed(ref ImScrollAreaStyle, Single)

Class ImScrollAreaStyleExtensions

Inheritance

System.Object
ImScrollAreaStyleExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class ImScrollAreaStyleExtensions
```

Methods

GetButtonStyle(in ImScrollAreaStyle)

Sets the implicit button style in the slider.

Declaration

```
public static ImButtonStyle GetButtonStyle(this in ImScrollAreaStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
ImScrollAreaStyle	style	The reference to the ImSliderStlye.

Returns

TYPE	DESCRIPTION
ImButtonStyle	The same reference to the ImScrollAreaStyle.

WithButtonStyle(ref ImScrollAreaStyle, in ImButtonStyle)

Sets the implicit button style in the slider.

Declaration

```
public static ref ImScrollAreaStyle WithButtonStyle(this ref ImScrollAreaStyle scroll, in ImButtonStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
ImScrollAreaStyle	scroll	The reference to the ImScrollAreaStyle.
ImButtonStyle	style	The reference to the ImSliderStlye.

Returns

TYPE	DESCRIPTION
ImScrollAreaStyle	The same reference to the ImScrollAreaStyle.

Struct ImSkipLineStyle

Defines the style of the skipped line.

Implements

IStyle

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImSkipLineStyle : IStyle
```

Fields

FontSize

The font sized used to define the amount of spacing to skip.

Declaration

```
public int FontSize
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Padding

The amount of spacing between the current the widget and next widget along both the X and Y axis.

Declaration

```
public float2 Padding
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Methods

New()

Constructs the default style of the skipped line using the DefaultStyles config.

Declaration

```
public static ImSkipLineStyle New()
```

Returns

TYPE	DESCRIPTION
ImSkipLineStyle	

Implements

IStyle

Extension Methods

StyleExtensions.WithPadding(ref ImSkipLineStyle, float2)
StyleExtensions.WithFontSize(ref ImSkipLineStyle, UInt16)

Struct ImSliderStyle

Defines how the slider button looks and its background.

Implements

IStyle

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImSliderStyle : IStyle
```

Fields

Background

Default color of the slider's background.

Declaration

```
public Color32 Background
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

ButtonDefault

Default color of the slider button.

Declaration

```
public Color32 ButtonDefault
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

ButtonHover

Declaration

```
public Color32 ButtonHover
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

ButtonPressed

Declaration

```
public Color32 ButtonPressed
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

FontSize

Declaration

```
public int FontSize
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Padding

Declaration

```
public float2 Padding
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

TextColor

Declaration

```
public Color32 TextColor
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Methods

New()

Constructs the default style.

Declaration

```
public static ImSliderStyle New()
```

Returns

TYPE	DESCRIPTION
ImSliderStyle	

Implements

IStyle

Extension Methods

StyleExtensions.WithBackground(ref ImSliderStyle, Color32)
StyleExtensions.WithButtonDefault(ref ImSliderStyle, Color32)
StyleExtensions.WithButtonHover(ref ImSliderStyle, Color32)
StyleExtensions.WithButtonPressed(ref ImSliderStyle, Color32)
StyleExtensions.WithTextColor(ref ImSliderStyle, Color32)
StyleExtensions.WithFontSize(ref ImSliderStyle, UInt16)
StyleExtensions.WithPadding(ref ImSliderStyle, float2)
ImSliderStyleExtensions.AsRef(ImSliderStyle)
ImSliderStyleExtensions.GetButtonStyle(in ImSliderStyle)
ImSliderStyleExtensions.GetTextStyle(in ImSliderStyle)
ImSliderStyleExtensions.WithButtonStyle(ref ImSliderStyle, in ImButtonStyle)

Class ImSliderStyleExtensions

Inheritance

System.Object
ImSliderStyleExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class ImSliderStyleExtensions
```

Methods

AsRef(ImSliderStyle)

Returns a reference to a copy of the style. This API is considered experimental.

Declaration

```
public static ref ImSliderStyle AsRef(this ImSliderStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
ImSliderStyle	style	The style to copy.

Returns

TYPE	DESCRIPTION
ImSliderStyle	The reference to the copy of the style.

GetButtonStyle(in ImSliderStyle)

Gets the implicit ImButtonStyle from the ImSliderStyle.

Declaration

```
public static ImButtonStyle GetButtonStyle(this in ImSliderStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
ImSliderStyle	style	The reference to the ImSliderStyle.

Returns

TYPE	DESCRIPTION
ImButtonStyle	An instance of the ImButtomStyle.

GetTextStyle(in ImSliderStyle)

Gets the implicit ImTextStyle from the ImSliderStyle.

Declaration

```
public static ImTextStyle GetTextStyle(this in ImSliderStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
ImSliderStyle	style	The reference to the ImSliderStyle.

Returns

TYPE	DESCRIPTION
ImTextStyle	An instance of the ImTextStyle.

WithButtonStyle(ref ImSliderStyle, in ImButtonStyle)

Sets the implicit ImButtonStyle to the ImSliderStyle.

Declaration

```
public static ref ImSliderStyle WithButtonStyle(this ref ImSliderStyle slider, in ImButtonStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
ImSliderStyle	slider	A reference to ImSliderStyle.
ImButtonStyle	style	A reference to the button style.

Returns

TYPE	DESCRIPTION
ImSliderStyle	

Struct ImTextFieldStyle

Stores the style of the TextField.

Implements

IStyle

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImTextFieldStyle : IStyle
```

Fields

Background

Background color for the textfield.

Declaration

```
public Color32 Background
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Column

Column wise alignment.

Declaration

```
public HorizontalAlignment Column
```

Field Value

TYPE	DESCRIPTION
HorizontalAlignment	

FontSize

Default size of the text.

Declaration

```
public int FontSize
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Hover

Color of the textfield when the mouse is over it.

Declaration

```
public Color32 Hover
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Padding

Spacing between the this and the next widget.

Declaration

```
public float2 Padding
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Row

Row wise alignment.

Declaration

```
public VerticalAlignment Row
```

Field Value

TYPE	DESCRIPTION
VerticalAlignment	

Text

Default color of the text.

Declaration

```
public Color32 Text
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Methods

New()

Constructs a new ImTextFieldStyle using values from the DefaultStyles. DefaultStyles

Declaration

```
public static ImTextFieldStyle New()
```

Returns

TYPE	DESCRIPTION
ImTextFieldStyle	An instance of the ImTextFieldStyle.

Implements

IStyle

Extension Methods

ImTextFieldStyleExtensions.GetTextStyle(in ImTextFieldStyle)
ImTextFieldStyleExtensions.GetButtonStyle(in ImTextFieldStyle)
StyleExtensions.WithFontSize(ref ImTextFieldStyle, UInt16)
StyleExtensions.WithText(ref ImTextFieldStyle, Color32)
StyleExtensions.WithBackground(ref ImTextFieldStyle, Color32)
StyleExtensions.WithHover(ref ImTextFieldStyle, Color32)
StyleExtensions.WithColumn(ref ImTextFieldStyle, HorizontalAlignment)
StyleExtensions.WithRow(ref ImTextFieldStyle, VerticalAlignment)
StyleExtensions.WithPadding(ref ImTextFieldStyle, float2)

Class ImTextFieldStyleExtensions

Inheritance

System.Object
ImTextFieldStyleExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class ImTextFieldStyleExtensions
```

Methods

GetButtonStyle(in ImTextFieldStyle)

Gets the implicit ImButtonStyle from the TextField.

Declaration

```
public static ImButtonStyle GetButtonStyle(this in ImTextFieldStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION
ImTextFieldStyle	style	The ImTextFieldStyle reference.

Returns

TYPE	DESCRIPTION
ImButtonStyle	An instance of ImButtonStyle.

GetTextStyle(in ImTextFieldStyle)

Gets the implicit ImTextFieldStyle from the TextField.

Declaration

```
public static ImTextStyle GetTextStyle(this in ImTextFieldStyle style)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
ImTextFieldStyle	style	The ImTextFieldStyle reference.

Returns

TYPE	DESCRIPTION
ImTextStyle	An instance of ImTextFieldStyle.

Struct ImTextStyle

Stores the font style.

Implements

IStyle

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImTextStyle : IStyle
```

Fields

Column

Column wise alignment.

Declaration

```
public HorizontalAlignment Column
```

Field Value

TYPE	DESCRIPTION
HorizontalAlignment	

FontSize

Default font size.

Declaration

```
public int FontSize
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Padding

The amount of space between the current and next widget.

Declaration

```
public float2 Padding
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Row

Row wise alignment.

Declaration

```
public VerticalAlignment Row
```

Field Value

TYPE	DESCRIPTION
VerticalAlignment	

TextColor

Default text color.

Declaration

```
public Color32 TextColor
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Methods

New()

Constructs a new instance of the ImTextStyle using default values from DefaultStyles. DefaultStyles

Declaration

```
public static ImTextStyle New()
```

Returns

TYPE	DESCRIPTION
ImTextStyle	An instance of ImTextStyle

Implements

IStyle

Extension Methods

StyleExtensions.WithFontSize(ref ImTextStyle, UInt16)
StyleExtensions.WithColumn(ref ImTextStyle, HorizontalAlignment)
StyleExtensions.WithRow(ref ImTextStyle, VerticalAlignment)
StyleExtensions.WithTextColor(ref ImTextStyle, Color32)
StyleExtensions.WithPadding(ref ImTextStyle, float2)

ImTextStyleExtensions.WithColor(ref ImTextStyle, Color32)

Class ImTextStyleExtensions

Inheritance

System.Object
ImTextStyleExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class ImTextStyleExtensions
```

Methods

WithColor(ref ImTextStyle, Color32)

Fluent API to set the color of the TextStyle.

Declaration

```
public static ref ImTextStyle WithColor(this ref ImTextStyle style, Color32 color)
```

Parameters

TYPE	NAME	DESCRIPTION
ImTextStyle	style	ImTextStyle to reference.
UnityEngine.Color32	color	Color of the text.

Returns

TYPE	DESCRIPTION
ImTextStyle	An instance of the ImTextStyle.

Class ImWindow

ImWindow stores rendering information that can't be used in jobs.

Inheritance

System.Object
ImWindow

Implements

System.IDisposable

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public class ImWindow : IDisposable
```

Constructors

ImWindow()

Declaration

```
public ImWindow()
```

ImWindow(Int32, int2, float2, String)

Declaration

```
public ImWindow(int capacity, int2 size, float2 position, string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	capacity	
Unity.Mathematics.int2	size	
Unity.Mathematics.float2	position	
System.String	name	

Methods

Dispose()

Declaration

```
public void Dispose()
```

Finalize()

Declaration

```
protected void Finalize()
```

PushTxt(ImString, ImRect, in ImTextStyle, Single)

Declaration

```
public void PushTxt(ImString content, ImRect r, in ImTextStyle style, float cutOff = 0.5F)
```

Parameters

TYPE	NAME	DESCRIPTION
ImString	content	
ImRect	r	
ImTextStyle	style	
System.Single	cutOff	

Implements

System.IDisposable

Extension Methods

WindowBehaviorExtensions.OpenPane(ImWindow, String)
WindowBehaviorExtensions.ClosePane(ImWindow, String)
WindowBehaviorExtensions.IsClosed(ImWindow, String)
ImGuiRenderUtils.PushSolidBox(ImWindow, in ImRect, in Color32, Single)
ImGuiRenderUtils.PushCheckmark(ImWindow, in ImRect, in Color32, Single)
ImGuiRenderUtils.PushX(ImWindow, in ImRect, in Color32, Single)
ImGuiRenderUtils.PushHamburgerMenu(ImWindow, in ImRect, in Color32, Single)

Interface IStyle

A trait for all common style structs for code generation.

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public interface IStyle
```

Enum PruneFlag

The global state to clean up.

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public enum PruneFlag : short
```

Fields

NAME	DESCRIPTION
All	
Collapsed	
Dropdown	
Pane	
Scroll	
Toggle	

Class StyleExtensions

Inheritance

System.Object
StyleExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class StyleExtensions
```

Methods

WithBackground(ref ImButtonStyle, Color32)

Declaration

```
public static ref ImButtonStyle WithBackground(this ref ImButtonStyle style, Color32 background)
```

Parameters

TYPE	NAME	DESCRIPTION
ImButtonStyle	style	
UnityEngine.Color32	background	

Returns

TYPE	DESCRIPTION
ImButtonStyle	

WithBackground(ref ImDropDownStyle, Color32)

Declaration

```
public static ref ImDropDownStyle WithBackground(this ref ImDropDownStyle style, Color32 background)
```

Parameters

TYPE	NAME	DESCRIPTION
ImDropDownStyle	style	
UnityEngine.Color32	background	

Returns

TYPE	DESCRIPTION
ImDropDownStyle	

WithBackground(ref ImProgressBarStyle, Color32)

Declaration

```
public static ref ImProgressBarStyle WithBackground(this ref ImProgressBarStyle style, Color32 background)
```

Parameters

TYPE	NAME	DESCRIPTION
ImProgressBarStyle	style	
UnityEngine.Color32	background	

Returns

TYPE	DESCRIPTION
ImProgressBarStyle	

WithBackground(ref ImSliderStyle, Color32)

Declaration

```
public static ref ImSliderStyle WithBackground(this ref ImSliderStyle style, Color32 background)
```

Parameters

TYPE	NAME	DESCRIPTION
ImSliderStyle	style	
UnityEngine.Color32	background	

Returns

TYPE	DESCRIPTION
ImSliderStyle	

WithBackground(ref ImTextFieldStyle, Color32)

Declaration

```
public static ref ImTextFieldStyle WithBackground(this ref ImTextFieldStyle style, Color32 background)
```

Parameters

TYPE	NAME	DESCRIPTION
ImTextFieldStyle	style	
UnityEngine.Color32	background	

Returns

TYPE	DESCRIPTION
ImTextFieldStyle	

WithButtonDefault(ref ImScrollAreaStyle, Color32)

Declaration

```
public static ref ImScrollAreaStyle WithButtonDefault(this ref ImScrollAreaStyle style, Color32 buttondefault)
```

Parameters

TYPE	NAME	DESCRIPTION
ImScrollAreaStyle	style	
UnityEngine.Color32	buttondefault	

Returns

TYPE	DESCRIPTION
ImScrollAreaStyle	

WithButtonDefault(ref ImSliderStyle, Color32)

Declaration

```
public static ref ImSliderStyle WithButtonDefault(this ref ImSliderStyle style, Color32 buttondefault)
```

Parameters

TYPE	NAME	DESCRIPTION
ImSliderStyle	style	
UnityEngine.Color32	buttondefault	

Returns

TYPE	DESCRIPTION
ImSliderStyle	

WithButtonHover(ref ImScrollAreaStyle, Color32)

Declaration

```
public static ref ImScrollAreaStyle WithButtonHover(this ref ImScrollAreaStyle style, Color32 buttonhover)
```

Parameters

TYPE	NAME	DESCRIPTION
ImScrollAreaStyle	style	
UnityEngine.Color32	buttonhover	

Returns

TYPE	DESCRIPTION
ImScrollAreaStyle	

WithButtonHover(ref ImSliderStyle, Color32)

Declaration

```
public static ref ImSliderStyle WithButtonHover(this ref ImSliderStyle style, Color32 buttonhover)
```

Parameters

TYPE	NAME	DESCRIPTION
ImSliderStyle	style	
UnityEngine.Color32	buttonhover	

Returns

TYPE	DESCRIPTION
ImSliderStyle	

WithButtonPressed(ref ImScrollAreaStyle, Color32)

Declaration

```
public static ref ImScrollAreaStyle WithButtonPressed(this ref ImScrollAreaStyle style, Color32 buttonpressed)
```

Parameters

TYPE	NAME	DESCRIPTION
ImScrollAreaStyle	style	
UnityEngine.Color32	buttonpressed	

Returns

TYPE	DESCRIPTION
ImScrollAreaStyle	

WithButtonPressed(ref ImSliderStyle, Color32)

Declaration

```
public static ref ImSliderStyle WithButtonPressed(this ref ImSliderStyle style, Color32 buttonpressed)
```

Parameters

TYPE	NAME	DESCRIPTION
ImSliderStyle	style	
UnityEngine.Color32	buttonpressed	

Returns

TYPE	DESCRIPTION
ImSliderStyle	

WithCloseDefaultFg(ref ImPaneStyle, Color32)

Declaration

```
public static ref ImPaneStyle WithCloseDefaultFg(this ref ImPaneStyle style, Color32 closedefaultfg)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
UnityEngine.Color32	closedefaultfg	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithCloseHoverFg(ref ImPaneStyle, Color32)

Declaration

```
public static ref ImPaneStyle WithCloseHoverFg(this ref ImPaneStyle style, Color32 closehoverfg)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
UnityEngine.Color32	closehoverfg	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithClosePressedFg(ref ImPaneStyle, Color32)

Declaration

```
public static ref ImPaneStyle WithClosePressedFg(this ref ImPaneStyle style, Color32 closepressedfg)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
UnityEngine.Color32	closepressedfg	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithCollapseDefaultFg(ref ImPaneStyle, Color32)

Declaration

```
public static ref ImPaneStyle WithCollapseDefaultFg(this ref ImPaneStyle style, Color32 collapsedefaultfg)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
UnityEngine.Color32	collapsedefaultfg	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithCollapseHoverFg(ref ImPaneStyle, Color32)

Declaration

```
public static ref ImPaneStyle WithCollapseHoverFg(this ref ImPaneStyle style, Color32 collapsehoverfg)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
UnityEngine.Color32	collapsehoverfg	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithCollapsePressedFg(ref ImPaneStyle, Color32)

Declaration

```
public static ref ImPaneStyle WithCollapsePressedFg(this ref ImPaneStyle style, Color32 collapsepressedfg)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
UnityEngine.Color32	collapsepressedfg	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithColor(ref ImLineStyle, Color32)

Declaration

```
public static ref ImLineStyle WithColor(this ref ImLineStyle style, Color32 color)
```

Parameters

TYPE	NAME	DESCRIPTION
ImLineStyle	style	
UnityEngine.Color32	color	

Returns

TYPE	DESCRIPTION
ImLineStyle	

WithColumn(ref ImButtonStyle, HorizontalAlignment)

Declaration

```
public static ref ImButtonStyle WithColumn(this ref ImButtonStyle style, HorizontalAlignment column)
```

Parameters

TYPE	NAME	DESCRIPTION
ImButtonStyle	style	
HorizontalAlignment	column	

Returns

TYPE	DESCRIPTION
ImButtonStyle	

WithColumn(ref ImDropDownStyle, HorizontalAlignment)

Declaration

```
public static ref ImDropDownStyle WithColumn(this ref ImDropDownStyle style, HorizontalAlignment column)
```

Parameters

TYPE	NAME	DESCRIPTION
ImDropDownStyle	style	
HorizontalAlignment	column	

Returns

TYPE	DESCRIPTION
ImDropDownStyle	

WithColumn(ref ImPaneStyle, HorizontalAlignment)

Declaration

```
public static ref ImPaneStyle WithColumn(this ref ImPaneStyle style, HorizontalAlignment column)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
HorizontalAlignment	column	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithColumn(ref ImProgressBarStyle, HorizontalAlignment)

Declaration

```
public static ref ImProgressBarStyle WithColumn(this ref ImProgressBarStyle style, HorizontalAlignment column)
```

Parameters

TYPE	NAME	DESCRIPTION
ImProgressBarStyle	style	
HorizontalAlignment	column	

Returns

TYPE	DESCRIPTION
ImProgressBarStyle	

WithColumn(ref ImTextFieldStyle, HorizontalAlignment)

Declaration

```
public static ref ImTextFieldStyle WithColumn(this ref ImTextFieldStyle style, HorizontalAlignment column)
```

Parameters

TYPE	NAME	DESCRIPTION
ImTextFieldStyle	style	
HorizontalAlignment	column	

Returns

TYPE	DESCRIPTION
ImTextFieldStyle	

WithColumn(ref ImTextStyle, HorizontalAlignment)

Declaration

```
public static ref ImTextStyle WithColumn(this ref ImTextStyle style, HorizontalAlignment column)
```

Parameters

TYPE	NAME	DESCRIPTION
ImTextStyle	style	
HorizontalAlignment	column	

Returns

TYPE	DESCRIPTION
ImTextStyle	

WithDefaultButtonBackground(ref ImPaneStyle, Color32)

Declaration

```
public static ref ImPaneStyle WithDefaultButtonBackground(this ref ImPaneStyle style, Color32
defaultbuttonbackground)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
UnityEngine.Color32	defaultbuttonbackground	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithDefaultButtonHover(ref ImPaneStyle, Color32)

Declaration

```
public static ref ImPaneStyle WithDefaultButtonHover(this ref ImPaneStyle style, Color32 defaultbuttonhover)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
UnityEngine.Color32	defaultbuttonhover	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithDefaultButtonPress(ref ImPaneStyle, Color32)

Declaration

```
public static ref ImPaneStyle WithDefaultButtonPress(this ref ImPaneStyle style, Color32 defaultbuttonpress)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
UnityEngine.Color32	defaultbuttonpress	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithDefaultFontSize(ref ImPaneStyle, UInt16)

Declaration

```
public static ref ImPaneStyle WithDefaultFontSize(this ref ImPaneStyle style, ushort defaultfontsize)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
System.UInt16	defaultfontsize	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithDeltaTime(ref ImScrollAreaStyle, Single)

Declaration

```
public static ref ImScrollAreaStyle WithDeltaTime(this ref ImScrollAreaStyle style, float deltatime)
```

Parameters

TYPE	NAME	DESCRIPTION
ImScrollAreaStyle	style	
System.Single	deltatime	

Returns

TYPE	DESCRIPTION
ImScrollAreaStyle	

WithFontSize(ref ImButtonStyle, UInt16)

Declaration

```
public static ref ImButtonStyle WithFontSize(this ref ImButtonStyle style, ushort fontsize)
```

Parameters

TYPE	NAME	DESCRIPTION
ImButtonStyle	style	
System.UInt16	fontsize	

Returns

TYPE	DESCRIPTION
ImButtonStyle	

WithFontSize(ref ImDropDownStyle, UInt16)

Declaration

```
public static ref ImDropDownStyle WithFontSize(this ref ImDropDownStyle style, ushort fontsize)
```

Parameters

TYPE	NAME	DESCRIPTION
ImDropDownStyle	style	
System.UInt16	fontsize	

Returns

TYPE	DESCRIPTION
ImDropDownStyle	

WithFontSize(ref ImProgressBarStyle, UInt16)

Declaration

```
public static ref ImProgressBarStyle WithFontSize(this ref ImProgressBarStyle style, ushort fontsize)
```

Parameters

TYPE	NAME	DESCRIPTION
ImProgressBarStyle	style	
System.UInt16	fontsize	

Returns

TYPE	DESCRIPTION
ImProgressBarStyle	

WithFontSize(ref ImSkipLineStyle, UInt16)

Declaration

```
public static ref ImSkipLineStyle WithFontSize(this ref ImSkipLineStyle style, ushort fontsize)
```

Parameters

TYPE	NAME	DESCRIPTION
ImSkipLineStyle	style	
System.UInt16	fontsize	

Returns

TYPE	DESCRIPTION
ImSkipLineStyle	

WithFontSize(ref ImSliderStyle, UInt16)

Declaration

```
public static ref ImSliderStyle WithFontSize(this ref ImSliderStyle style, ushort fontsize)
```

Parameters

TYPE	NAME	DESCRIPTION
ImSliderStyle	style	
System.UInt16	fontsize	

Returns

TYPE	DESCRIPTION
ImSliderStyle	

WithFontSize(ref ImTextFieldStyle, UInt16)

Declaration

```
public static ref ImTextFieldStyle WithFontSize(this ref ImTextFieldStyle style, ushort fontsize)
```

Parameters

TYPE	NAME	DESCRIPTION
ImTextFieldStyle	style	
System.UInt16	fontsize	

Returns

TYPE	DESCRIPTION
ImTextFieldStyle	

WithFontSize(ref ImTextStyle, UInt16)

Declaration

```
public static ref ImTextStyle WithFontSize(this ref ImTextStyle style, ushort fontsize)
```

Parameters

TYPE	NAME	DESCRIPTION
ImTextStyle	style	
System.UInt16	fontsize	

Returns

TYPE	DESCRIPTION
ImTextStyle	

WithForeground(ref ImProgressBarStyle, Color32)

Declaration

```
public static ref ImProgressBarStyle WithForeground(this ref ImProgressBarStyle style, Color32 foreground)
```

Parameters

TYPE	NAME	DESCRIPTION
ImProgressBarStyle	style	
UnityEngine.Color32	foreground	

Returns

TYPE	DESCRIPTION
ImProgressBarStyle	

WithHover(ref ImButtonStyle, Color32)

Declaration

```
public static ref ImButtonStyle WithHover(this ref ImButtonStyle style, Color32 hover)
```

Parameters

TYPE	NAME	DESCRIPTION
ImButtonStyle	style	
UnityEngine.Color32	hover	

Returns

TYPE	DESCRIPTION
ImButtonStyle	

WithHover(ref ImDropDownStyle, Color32)

Declaration

```
public static ref ImDropDownStyle WithHover(this ref ImDropDownStyle style, Color32 hover)
```

Parameters

TYPE	NAME	DESCRIPTION
ImDropDownStyle	style	
UnityEngine.Color32	hover	

Returns

TYPE	DESCRIPTION
ImDropDownStyle	

WithHover(ref ImTextFieldStyle, Color32)

Declaration

```
public static ref ImTextFieldStyle WithHover(this ref ImTextFieldStyle style, Color32 hover)
```

Parameters

TYPE	NAME	DESCRIPTION
ImTextFieldStyle	style	
UnityEngine.Color32	hover	

Returns

TYPE	DESCRIPTION
ImTextFieldStyle	

WithPadding(ref ImButtonStyle, float2)

Declaration

```
public static ref ImButtonStyle WithPadding(this ref ImButtonStyle style, float2 padding)
```

Parameters

TYPE	NAME	DESCRIPTION
ImButtonStyle	style	
Unity.Mathematics.float2	padding	

Returns

TYPE	DESCRIPTION
ImButtonStyle	

WithPadding(ref ImDropDownStyle, float2)

Declaration

```
public static ref ImDropDownStyle WithPadding(this ref ImDropDownStyle style, float2 padding)
```

Parameters

TYPE	NAME	DESCRIPTION
ImDropDownStyle	style	
Unity.Mathematics.float2	padding	

Returns

TYPE	DESCRIPTION
ImDropDownStyle	

WithPadding(ref ImLineStyle, Single)

Declaration

```
public static ref ImLineStyle WithPadding(this ref ImLineStyle style, float padding)
```

Parameters

TYPE	NAME	DESCRIPTION
ImLineStyle	style	
System.Single	padding	

Returns

TYPE	DESCRIPTION
ImLineStyle	

WithPadding(ref ImPaneStyle, float2)

Declaration

```
public static ref ImPaneStyle WithPadding(this ref ImPaneStyle style, float2 padding)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
Unity.Mathematics.float2	padding	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithPadding(ref ImScrollAreaStyle, float2)

Declaration

```
public static ref ImScrollAreaStyle WithPadding(this ref ImScrollAreaStyle style, float2 padding)
```

Parameters

TYPE	NAME	DESCRIPTION
ImScrollAreaStyle	style	
Unity.Mathematics.float2	padding	

Returns

TYPE	DESCRIPTION
ImScrollAreaStyle	

WithPadding(ref ImSkipLineStyle, float2)

Declaration

```
public static ref ImSkipLineStyle WithPadding(this ref ImSkipLineStyle style, float2 padding)
```

Parameters

TYPE	NAME	DESCRIPTION
ImSkipLineStyle	style	
Unity.Mathematics.float2	padding	

Returns

TYPE	DESCRIPTION
ImSkipLineStyle	

WithPadding(ref ImSliderStyle, float2)

Declaration

```
public static ref ImSliderStyle WithPadding(this ref ImSliderStyle style, float2 padding)
```

Parameters

TYPE	NAME	DESCRIPTION
ImSliderStyle	style	
Unity.Mathematics.float2	padding	

Returns

TYPE	DESCRIPTION
ImSliderStyle	

WithPadding(ref ImTextFieldStyle, float2)

Declaration

```
public static ref ImTextFieldStyle WithPadding(this ref ImTextFieldStyle style, float2 padding)
```

Parameters

TYPE	NAME	DESCRIPTION
ImTextFieldStyle	style	
Unity.Mathematics.float2	padding	

Returns

TYPE	DESCRIPTION
ImTextFieldStyle	

WithPadding(ref ImTextStyle, float2)

Declaration

```
public static ref ImTextStyle WithPadding(this ref ImTextStyle style, float2 padding)
```

Parameters

TYPE	NAME	DESCRIPTION
ImTextStyle	style	
Unity.Mathematics.float2	padding	

Returns

TYPE	DESCRIPTION
ImTextStyle	

WithPane(ref ImPaneStyle, Color32)

Declaration

```
public static ref ImPaneStyle WithPane(this ref ImPaneStyle style, Color32 pane)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
UnityEngine.Color32	pane	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithPressed(ref ImButtonStyle, Color32)

Declaration

```
public static ref ImButtonStyle WithPressed(this ref ImButtonStyle style, Color32 pressed)
```

Parameters

TYPE	NAME	DESCRIPTION
ImButtonStyle	style	
UnityEngine.Color32	pressed	

Returns

TYPE	DESCRIPTION
ImButtonStyle	

WithPressed(ref ImDropDownStyle, Color32)

Declaration

```
public static ref ImDropDownStyle WithPressed(this ref ImDropDownStyle style, Color32 pressed)
```

Parameters

TYPE	NAME	DESCRIPTION
ImDropDownStyle	style	
UnityEngine.Color32	pressed	

Returns

TYPE	DESCRIPTION
ImDropDownStyle	

WithRow(ref ImButtonStyle, VerticalAlignment)

Declaration

```
public static ref ImButtonStyle WithRow(this ref ImButtonStyle style, VerticalAlignment row)
```

Parameters

TYPE	NAME	DESCRIPTION
ImButtonStyle	style	
VerticalAlignment	row	

Returns

TYPE	DESCRIPTION
ImButtonStyle	

WithRow(ref ImDropDownStyle, VerticalAlignment)

Declaration

```
public static ref ImDropDownStyle WithRow(this ref ImDropDownStyle style, VerticalAlignment row)
```

Parameters

TYPE	NAME	DESCRIPTION
ImDropDownStyle	style	
VerticalAlignment	row	

Returns

TYPE	DESCRIPTION
ImDropDownStyle	

WithRow(ref ImPaneStyle, VerticalAlignment)

Declaration

```
public static ref ImPaneStyle WithRow(this ref ImPaneStyle style, VerticalAlignment row)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
VerticalAlignment	row	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithRow(ref ImProgressBarStyle, VerticalAlignment)

Declaration

```
public static ref ImProgressBarStyle WithRow(this ref ImProgressBarStyle style, VerticalAlignment row)
```

Parameters

TYPE	NAME	DESCRIPTION
ImProgressBarStyle	style	
VerticalAlignment	row	

Returns

TYPE	DESCRIPTION
ImProgressBarStyle	

WithRow(ref ImTextFieldStyle, VerticalAlignment)

Declaration

```
public static ref ImTextFieldStyle WithRow(this ref ImTextFieldStyle style, VerticalAlignment row)
```

Parameters

TYPE	NAME	DESCRIPTION
ImTextFieldStyle	style	
VerticalAlignment	row	

Returns

TYPE	DESCRIPTION
ImTextFieldStyle	

WithRow(ref ImTextStyle, VerticalAlignment)

Declaration

```
public static ref ImTextStyle WithRow(this ref ImTextStyle style, VerticalAlignment row)
```

Parameters

TYPE	NAME	DESCRIPTION
ImTextStyle	style	
VerticalAlignment	row	

Returns

TYPE	DESCRIPTION
ImTextStyle	

WithScrollBarBackground(ref ImScrollAreaStyle, Color32)

Declaration

```
public static ref ImScrollAreaStyle WithScrollBarBackground(this ref ImScrollAreaStyle style, Color32
scrollbarbackground)
```

Parameters

TYPE	NAME	DESCRIPTION
ImScrollAreaStyle	style	
UnityEngine.Color32	scrollbarbackground	

TYPE	NAME	DESCRIPTION

Returns

TYPE	DESCRIPTION
ImScrollAreaStyle	

WithScrollBarPanel(ref ImScrollAreaStyle, Color32)

Declaration

```
public static ref ImScrollAreaStyle WithScrollBarPanel(this ref ImScrollAreaStyle style, Color32
scrollbarpanel)
```

Parameters

TYPE	NAME	DESCRIPTION
ImScrollAreaStyle	style	
UnityEngine.Color32	scrollbarpanel	

Returns

TYPE	DESCRIPTION
ImScrollAreaStyle	

WithScrollButtonWidth(ref ImScrollAreaStyle, Single)

Declaration

```
public static ref ImScrollAreaStyle WithScrollButtonWidth(this ref ImScrollAreaStyle style, float
scrollbuttonwidth)
```

Parameters

TYPE	NAME	DESCRIPTION
ImScrollAreaStyle	style	
System.Single	scrollbuttonwidth	

Returns

TYPE	DESCRIPTION
ImScrollAreaStyle	

WithScrollSpeed(ref ImScrollAreaStyle, Single)

Declaration

```
public static ref ImScrollAreaStyle WithScrollSpeed(this ref ImScrollAreaStyle style, float scrollspeed)
```

Parameters

TYPE	NAME	DESCRIPTION
ImScrollAreaStyle	style	
System.Single	scrollspeed	

Returns

TYPE	DESCRIPTION
ImScrollAreaStyle	

WithText(ref ImButtonStyle, Color32)

Declaration

```
public static ref ImButtonStyle WithText(this ref ImButtonStyle style, Color32 text)
```

Parameters

TYPE	NAME	DESCRIPTION
ImButtonStyle	style	
UnityEngine.Color32	text	

Returns

TYPE	DESCRIPTION
ImButtonStyle	

WithText(ref ImDropDownStyle, Color32)

Declaration

```
public static ref ImDropDownStyle WithText(this ref ImDropDownStyle style, Color32 text)
```

Parameters

TYPE	NAME	DESCRIPTION
ImDropDownStyle	style	
UnityEngine.Color32	text	

Returns

TYPE	DESCRIPTION
ImDropDownStyle	

WithText(ref ImPaneStyle, Color32)

Declaration

```
public static ref ImPaneStyle WithText(this ref ImPaneStyle style, Color32 text)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
UnityEngine.Color32	text	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithText(ref ImTextFieldStyle, Color32)

Declaration

```
public static ref ImTextFieldStyle WithText(this ref ImTextFieldStyle style, Color32 text)
```

Parameters

TYPE	NAME	DESCRIPTION
ImTextFieldStyle	style	
UnityEngine.Color32	text	

Returns

TYPE	DESCRIPTION
ImTextFieldStyle	

WithTextColor(ref ImProgressBarStyle, Color32)

Declaration

```
public static ref ImProgressBarStyle WithTextColor(this ref ImProgressBarStyle style, Color32 textcolor)
```

Parameters

TYPE	NAME	DESCRIPTION
ImProgressBarStyle	style	
UnityEngine.Color32	textcolor	

Returns

TYPE	DESCRIPTION
ImProgressBarStyle	

WithTextColor(ref ImSliderStyle, Color32)

Declaration

```
public static ref ImSliderStyle WithTextColor(this ref ImSliderStyle style, Color32 textcolor)
```

Parameters

TYPE	NAME	DESCRIPTION
ImSliderStyle	style	
UnityEngine.Color32	textcolor	

Returns

TYPE	DESCRIPTION
ImSliderStyle	

WithTextColor(ref ImTextStyle, Color32)

Declaration

```
public static ref ImTextStyle WithTextColor(this ref ImTextStyle style, Color32 textcolor)
```

Parameters

TYPE	NAME	DESCRIPTION
ImTextStyle	style	
UnityEngine.Color32	textcolor	

Returns

TYPE	DESCRIPTION
ImTextStyle	

WithTitleBar(ref ImPaneStyle, Color32)

Declaration

```
public static ref ImPaneStyle WithTitleBar(this ref ImPaneStyle style, Color32 titlebar)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
UnityEngine.Color32	titlebar	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

WithTitleFontSize(ref ImPaneStyle, UInt16)

Declaration

```
public static ref ImPaneStyle WithTitleFontSize(this ref ImPaneStyle style, ushort titlefontsize)
```

Parameters

TYPE	NAME	DESCRIPTION
ImPaneStyle	style	
System.UInt16	titlefontsize	

Returns

TYPE	DESCRIPTION
ImPaneStyle	

Struct UnmanagedImWindow

To support Unity Jobs and Burst, this will store only blittable information.

Implements

System.IDisposable

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct UnmanagedImWindow : IDisposable
```

Constructors

UnmanagedImWindow(Int32, int2, float2)

Declaration

```
public UnmanagedImWindow(int capacity, int2 size, float2 position)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	capacity	
Unity.Mathematics.int2	size	
Unity.Mathematics.float2	position	

Methods

Dispose()

Declaration

```
public void Dispose()
```

Implements

System.IDisposable

Class WindowBehaviorExtensions

Inheritance

System.Object
WindowBehaviorExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class WindowBehaviorExtensions
```

Methods

ClosePane(ImWindow, String)

If the pane is available, closes the window.

Declaration

```
public static void ClosePane(this ImWindow window, string title)
```

Parameters

TYPE	NAME	DESCRIPTION
ImWindow	window	The window containing the pane.
System.String	title	The title of the pane

IsClosed(ImWindow, String)

Is the window closed?

Declaration

```
public static bool IsClosed(this ImWindow window, string title)
```

Parameters

TYPE	NAME	DESCRIPTION
ImWindow	window	The window containing the pane.

TYPE	NAME	DESCRIPTION
System.String	title	The title of the pane to look for.

Returns

TYPE	DESCRIPTION
System.Boolean	

OpenPane(ImWindow, String)

If the pane was previously closed, shows the window.

Declaration

```
public static void OpenPane(this ImWindow window, string title)
```

Parameters

TYPE	NAME	DESCRIPTION
ImWindow	window	The window containing the pane.
System.String	title	The title of the pane

Namespace InitialPrefabs.NimGui.Common

Classes

ImIdUtility

The ImId class is a utility to get an unsigned integer as a control ID.

Class ImIdUtility

The ImId class is a utility to get an unsigned integer as a control ID.

Inheritance

System.Object
ImIdUtility

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui.Common

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class ImIdUtility
```

Methods

RequestId()

Returns the current integer and increments so the next time RequestId() is called, the next positive integer is returned.

Declaration

```
public static uint RequestId()
```

Returns

TYPE	DESCRIPTION
System.UInt32	The current unsigned integer cached

Namespace InitialPrefabs.NimGui.Inputs

Classes

InputHelper

The LegacyInputHelper interfaces with Unity's old input system and tracks mouse clicks and scroll wheel.

InputTextExtensions

MouseExtensions

Structs

InputText

Generic struct which stores keyboard inputs used for a textfield.

Mouse

Stores information on what the mouse inputs are.

Enums

Mouse.State

Describes the current state of the mouse.

Class InputHelper

The LegacyInputHelper interfaces with Unity's old input system and tracks mouse clicks and scroll wheel.

Inheritance

System.Object
InputHelper

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui.Inputs

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class InputHelper
```

Methods

GetInputTextHelper()

Allows access to the InputTextHelper.

Declaration

```
public static ref InputText GetInputTextHelper()
```

Returns

TYPE	DESCRIPTION
InputText	A reference to the static InputTextHelper.

GetMouseState()

Returns a copy of the mouse struct. Any manipulations on the copy will not be reflected.

Declaration

```
public static Mouse GetMouseState()
```

Returns

TYPE	DESCRIPTION
Mouse	Copy of the mouse struct.

Struct InputText

Generic struct which stores keyboard inputs used for a textfield.

Implements

System.IDisposable

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui.Inputs

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct InputText : IDisposable
```

Constructors

InputText(Int32)

Construct a LegacyInputText given a max capacity.

Declaration

```
public InputText(int maxCapacity)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	maxCapacity	The size of the buffer

Properties

IsBackspaced

Did the user press backspace? This is a read only public property and is internally set.

Declaration

```
public bool IsBackspaced { readonly get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

IsEntered

Did the user press enter? This is a read only public property and is internally set.

Declaration

```
public bool IsEntered { readonly get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

Dispose()

Release all internal memory allocated.

Declaration

```
public void Dispose()
```

GetReadOnlyInput()

Returns a readonly array pointing to the internal memory allocated.

Declaration

```
public UnsafeArray<char>.ReadOnly GetReadOnlyInput()
```

Returns

TYPE	DESCRIPTION
InitialPrefabs.NimGui.Collections.UnsafeArray.ReadOnly<>	

IsCreated()

Checks if the LegacyInputText is initialized.

Declaration

```
public bool IsCreated()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

Reset()

Resets the bump allocator's index, allowing reuse of the internal buffer.

Declaration

```
public void Reset()
```

Implements

System.IDisposable

Extension Methods

InputTextExtensions.AppendTo(ref InputText, StringBuilder)

Class InputTextExtensions

Inheritance

System.Object
InputTextExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui.Inputs

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class InputTextExtensions
```

Methods

AppendTo(ref InputText, StringBuilder)

Copies all the characters in the internal buffer to a StringBuilder. The StringBuilder must be allocated by the developer with a MaxCapacity. If the inputText exceeds the max capacity of the StringBuilder, then the inputText is not appended to the StringBuilder.

```
StringBuilder builder = new StringBuilder(32, 64);
ImGui.TextField("TextEdit", builder);
```

Declaration

```
public static void AppendTo(this ref InputText inputText, StringBuilder builder)
```

Parameters

TYPE	NAME	DESCRIPTION
InputText	inputText	A reference to the LegacyInputText
System.Text.StringBuilder	builder	The allocated StringBuilder to add to.

Struct Mouse

Stores information on what the mouse inputs are.

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui.Inputs

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct Mouse
```

Fields

Click

What is the current primary mouse button state?

Declaration

```
public Mouse.State Click
```

Field Value

TYPE	DESCRIPTION
Mouse.State	

IsScrolling

Is the scroll wheel actively used?

Declaration

```
public bool IsScrolling
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Position

What position is the mouse on?

Declaration

```
public int2 Position
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.int2	

ScrollDelta

What was the change in the scroll wheel?

Declaration

```
public float2 ScrollDelta
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Methods

ToString()

Provide a human readable format of the text.

Declaration

```
public override string ToString()
```

Returns

TYPE	DESCRIPTION
System.String	An easily text readable statement of the mouse state.

Overrides

System.ValueType.ToString()

Extension Methods

MouseExtensions.Is(in Mouse, Mouse.State)
MouseExtensions.IsAny(in Mouse, Mouse.State)

Enum Mouse.State

Describes the current state of the mouse.

Namespace: InitialPrefabs.NimGui.Inputs

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public enum State
```

Fields

NAME	DESCRIPTION
Down	
Held	
None	
Released	

Class MouseExtensions

Inheritance

System.Object
MouseExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui.Inputs

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class MouseExtensions
```

Methods

Is(in Mouse, Mouse.State)

Extension function to check if the mouse is currently a state.

Declaration

```
public static bool Is(this in Mouse mouse, Mouse.State state)
```

Parameters

TYPE	NAME	DESCRIPTION
Mouse	mouse	Reference to the mouse struct
Mouse.State	state	The current state we want to compare to

Returns

TYPE	DESCRIPTION
System.Boolean	True, if the mouse state is the state we are looking for

IsAny(in Mouse, Mouse.State)

Extension function to check if the mouse is any of the following states. Mouse.State

Declaration

```
public static bool IsAny(this in Mouse mouse, Mouse.State state)
```

Parameters

TYPE	NAME	DESCRIPTION
Mouse	mouse	Reference to the mouse struct
Mouse.State	state	The current state we want to compare to

Returns

TYPE	DESCRIPTION
System.Boolean	True, if the mouse state is at least one of the state we are looking for

Namespace InitialPrefabs.NimGui.Loop

Classes

DefaultImGuiInitialization

Enums

ResultFlag

State flags to determine how NimGui initialized.

Class DefaultImGuiInitialization

Inheritance

System.Object
DefaultImGuiInitialization

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui.Loop

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class DefaultImGuiInitialization
```

Methods

SetupCamera(Camera, CameraEvent)

Declaration

```
public static void SetupCamera(Camera camera, CameraEvent evt)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Camera	camera	
UnityEngine.Rendering.CameraEvent	evt	

TearDownCamera(Camera, CameraEvent)

Cleans up the target camera of the command buffer responsible for drawing UI.

Declaration

```
public static void TearDownCamera(Camera camera, CameraEvent evt)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Camera	camera	The camera that handles rendering the UI.
UnityEngine.Rendering.CameraEvent	evt	The event at which UI is currently drawn at.

Enum ResultFlag

State flags to determine how NimGui initialized.

Namespace: InitialPrefabs.NimGui.Loop

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public enum ResultFlag
```

Fields

NAME	DESCRIPTION
MissingFontAsset	The default glyph asset was not imported when importing the NimGui package.
MissingFontTexture	The default font texture was not imported when importing the NimGui package.
MissingPipeline	The URP_ENABLED define is added into your project, but the Graphics Settings pipeline asset is unassigned.
MissingRenderPass	You are using URP, but the ImGuiRenderFeaturePass is not added to your Renderer.
MissingShader	InitialPrefabs/SDF shader is missing.
Success	There were no errors when initializing NimGui.

Namespace InitialPrefabs.NimGui.Plot

Classes

ImGui

Class ImGui

Inheritance

System.Object

ImGui

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: InitialPrefabs.NimGui.Plot

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class ImGui
```

Namespace InitialPrefabs.NimGui.Render

Classes

ImGuiRenderUtils

Structs

ImDrawData

Stores universal data for all elements rendered.

ImSpriteData

Convenient struct to store sprite data from Unity.

ImVertex

Stores rendering data such as position, color, texcoord0, texcoord1.

Struct ImDrawData

Stores universal data for all elements rendered.

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui.Render

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImDrawData
```

Fields

Arguments

Extraneous arguments to help decipher the draw data. This is arbitrary data.

Declaration

```
public sbyte Arguments
```

Field Value

TYPE	DESCRIPTION
System.SByte	

Color

Stores the vertex color of the mesh.

Declaration

```
public Color32 Color
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Color32	

Cutoff

Since we are using SDFs, it is important that each element in a "draw call" determine the cutoff for the SDF.

For example, with text, you may want a cutoff of 0.5. But for solid blocks, you may want a cutoff of 0 so that the rect is full.

Declaration

```
public float Cutoff
```

Field Value

TYPE	DESCRIPTION
System.Single	

Rect

Stores the general bounds of the UI element.

Declaration

```
public ImRect Rect
```

Field Value

TYPE	DESCRIPTION
ImRect	

Type

Stores the type of element we intend to render

Declaration

```
public ImDrawCommandType Type
```

Field Value

TYPE	DESCRIPTION
ImDrawCommandType	

Class ImGuiRenderUtils

Inheritance

System.Object
ImGuiRenderUtils

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui.Render

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class ImGuiRenderUtils
```

Methods

GetFontFace()

The primary font face used for rendering the entire UI.

Declaration

```
public static ref ImFontFace GetFontFace()
```

Returns

TYPE	DESCRIPTION
ImFontFace	A reference to the FontFace.

GetGlyphs()

The primary glyphs associated with the FontFace. All glyphs are sorted with their unicode values.

Declaration

```
public static UnsafeArray<ImGlyph> GetGlyphs()
```

Returns

TYPE	DESCRIPTION
InitialPrefabs.NimGui.Collections.UnsafeArray<ImGlyph>	The sorted glyphs.

GetMaterial()

The primary material used for rendering the entire UI.

Declaration

```
public static Material GetMaterial()
```

Returns

TYPE	DESCRIPTION
UnityEngine.Material	A reference to the material.

PushCheckmark(ImWindow, in ImRect, in Color32, Single)

Queues a draw command to draw a checkmark.

Declaration

```
public static void PushCheckmark(this ImWindow window, in ImRect rect, in Color32 color, float cutoff = 0.5F)
```

Parameters

TYPE	NAME	DESCRIPTION
ImWindow	window	The window to enqueue the draw command to.
ImRect	rect	The size of the checkmark.
UnityEngine.Color32	color	The color of the checkmark.
System.Single	cutoff	Optional cutoff, typically this should be set to 0.5 for minimal cutoff.

PushHamburgerMenu(ImWindow, in ImRect, in Color32, Single)

Queues a draw command to draw an X.

Declaration

```
public static void PushHamburgerMenu(this ImWindow window, in ImRect rect, in Color32 color, float cutoff =
0.5F)
```

Parameters

TYPE	NAME	DESCRIPTION
ImWindow	window	The window to enqueue to draw command to.
ImRect	rect	The size of the x.
UnityEngine.Color32	color	The color of the x.
System.Single	cutoff	Optional cutoff, typically this should be set to 0.5 for minimal cutoff.

PushSolidBox(ImWindow, in ImRect, in Color32, Single)

Queues a draw command to draw a solid colored box.

Declaration

```
public static void PushSolidBox(this ImWindow window, in ImRect rect, in Color32 color, float cutoff = 0F)
```

Parameters

TYPE	NAME	DESCRIPTION
ImWindow	window	The window to enqueue the draw command to.
ImRect	rect	The size of the box.
UnityEngine.Color32	color	The color of the box.
System.Single	cutoff	Optional cutoff, typically this should be set to 0 for no cutoff.

PushX(ImWindow, in ImRect, in Color32, Single)

Queues a draw command to draw an X.

Declaration

```
public static void PushX(this ImWindow window, in ImRect rect, in Color32 color, float cutoff = 0.5F)
```

Parameters

TYPE	NAME	DESCRIPTION
ImWindow	window	The window to enqueue to draw command to.
ImRect	rect	The size of the x.
UnityEngine.Color32	color	The color of the x.
System.Single	cutoff	Optional cutoff, typically this should be set to 0.5 for minimal cutoff.

Struct ImSpriteData

Convenient struct to store sprite data from Unity.

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui.Render

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImSpriteData
```

Fields

InnerUV

The UVs of the image where xy is the min and zw is the max.

Declaration

```
public float4 InnerUV
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float4	

Struct ImVertex

Stores rendering data such as position, color, texcoord0, texcoord1.

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui.Render

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImVertex
```

Fields

Color

Declaration

```
public float4 Color
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float4	

Position

Declaration

```
public float3 Position
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float3	

UV0

Declaration

```
public float2 UV0
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

UV1

Declaration

```
public float2 UV1
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Namespace InitialPrefabs.NimGui.Text

Classes

FontFaceExtensions

ImStringExtensions

SerializedFontData

A project wide asset which stores the FontFace and potential glyphs that the font can render.

This should generally only be constructed in the Editor.

TextUtils

Structs

GlyphComparer

Convenience struct to easily compare two glyphs' relative order.

HeightInfo

ImFontFace

Stores description of how the font is laid out.

ImGlyph

A glyph stores metrics of each character in the font. This describes how each character is laid out and how much space exists between each character.

ImString

Unsafe representation of a string. The struct does not implement an IDisposable interface because the purpose of the struct is to "borrow" a reference to a string's pointer. This is typically used in conjunection with ImWords or a fixed string. ImWords

ImWords

Create a persistent buffer which stores all characters requested. This is a bump allocator and the data must be reset each frame.

TextUtils.LineInfo

Enums

HorizontalAlignment

VerticalAlignment

Class FontFaceExtensions

Inheritance

System.Object
FontFaceExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui.Text

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class FontFaceExtensions
```

Methods

CalculateLineHeight(in ImFontFace, Int32, Single)

Declaration

```
public static float CalculateLineHeight(this in ImFontFace fontFace, int fontSize, float yPadding)
```

Parameters

TYPE	NAME	DESCRIPTION
ImFontFace	fontFace	
System.Int32	fontSize	
System.Single	yPadding	

Returns

TYPE	DESCRIPTION
System.Single	

Struct GlyphComparer

Convenience struct to easily compare two glyphs' relative order.

Implements

System.Collections.Generic.IComparer<ImGlyph>

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui.Text

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct GlyphComparer : IComparer<ImGlyph>
```

Methods

Compare(ImGlyph, ImGlyph)

Declaration

```
public int Compare(ImGlyph x, ImGlyph y)
```

Parameters

TYPE	NAME	DESCRIPTION
ImGlyph	x	
ImGlyph	y	

Returns

TYPE	DESCRIPTION
System.Int32	

Implements

System.Collections.Generic.IComparer<T>

Struct HeightInfo

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui.Text

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public ref struct HeightInfo
```

Constructors

HeightInfo(Int32, Single, in ImFontFace)

Declaration

```
public HeightInfo(int lineCount, float fontScale, in ImFontFace faceInfo)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	lineCount	
System.Single	fontScale	
ImFontFace	faceInfo	

Fields

AscentLine

Declaration

```
public readonly float AscentLine
```

Field Value

TYPE	DESCRIPTION
System.Single	

DescentLine

Declaration

```
public readonly float DescentLine
```

Field Value

TYPE	DESCRIPTION
System.Single	

LineHeight

Declaration

```
public readonly float LineHeight
```

Field Value

TYPE	DESCRIPTION
System.Single	

TextBlockHeight

Declaration

```
public readonly float TextBlockHeight
```

Field Value

TYPE	DESCRIPTION
System.Single	

Enum HorizontalAlignment

Namespace: InitialPrefabs.NimGui.Text

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public enum HorizontalAlignment : byte
```

Fields

NAME	DESCRIPTION
Center	
Left	
Right	

Struct ImFontFace

Stores description of how the font is laid out.

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui.Text

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
[Serializable]
public struct ImFontFace
```

Fields

AscentLine

Declaration

```
public float AscentLine
```

Field Value

TYPE	DESCRIPTION
System.Single	

BaseLine

Declaration

```
public float BaseLine
```

Field Value

TYPE	DESCRIPTION
System.Single	

CapLine

Declaration

```
public float CapLine
```

Field Value

TYPE	DESCRIPTION
System.Single	

DescentLine

Declaration

```
public float DescentLine
```

Field Value

TYPE	DESCRIPTION
System.Single	

LineHeight

Declaration

```
public float LineHeight
```

Field Value

TYPE	DESCRIPTION
System.Single	

MeanLine

Declaration

```
public float MeanLine
```

Field Value

TYPE	DESCRIPTION
System.Single	

PointSize

Declaration

```
public float PointSize
```

Field Value

TYPE	DESCRIPTION
System.Single	

Scale

Declaration

```
public float Scale
```

Field Value

TYPE	DESCRIPTION
System.Single	

StrikeThroughOffset

Declaration

```
public float StrikeThroughOffset
```

Field Value

TYPE	DESCRIPTION
System.Single	

StrikeThroughThickness

Declaration

```
public float StrikeThroughThickness
```

Field Value

TYPE	DESCRIPTION
System.Single	

SubscriptOffset

Declaration

```
public float SubscriptOffset
```

Field Value

TYPE	DESCRIPTION
System.Single	

SubscriptSize

Declaration

```
public float SubscriptSize
```

Field Value

TYPE	DESCRIPTION
System.Single	

SuperscriptOffset

Declaration

```
public float SuperscriptOffset
```

Field Value

TYPE	DESCRIPTION
System.Single	

SuperscriptSize

Declaration

```
public float SuperscriptSize
```

Field Value

TYPE	DESCRIPTION
System.Single	

TabWidth

Declaration

```
public float TabWidth
```

Field Value

TYPE	DESCRIPTION
System.Single	

UnderlineOffset

Declaration

```
public float UnderlineOffset
```

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

Create(FaceInfo)

Constructs a FontFace from UnityEngine.TextCore's FaceInfo.

Declaration

```
public static ImFontFace Create(FaceInfo info)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.TextCore.FaceInfo	info	The FaceInfo to construct from.

Returns

TYPE	DESCRIPTION
ImFontFace	A copy of the FaceInfo into a FontFace struct.

Extension Methods

FontFaceExtensions.CalculateLineHeight(in ImFontFace, Int32, Single)

Struct ImGlyph

A glyph stores metrics of each character in the font. This describes how each character is laid out and how much space exists between each character.

Implements

System.IComparable<ImGlyph>

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui.Text

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
[Serializable]
public struct ImGlyph : IComparable<ImGlyph>
```

Fields

Advance

The spacing between the left edge of the character to the next character.

Declaration

```
public float Advance
```

Field Value

TYPE	DESCRIPTION
System.Single	

Bearings

X bearing store the spacing between the previous rectangle and the character. While y bearing store the offset from the baseline to the top of the rectangle.

Declaration

```
public float2 Bearings
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

MetricsSize

How big is the rectangle for the font?

Declaration

```
public float2 MetricsSize
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float2	

Unicode

Declaration

```
public uint Unicode
```

Field Value

TYPE	DESCRIPTION
System.UInt32	

Uvs

Stores the texture coordinates to render the font. XY stores the min, while zw stores the max.

Declaration

```
public float4 Uvs
```

Field Value

TYPE	DESCRIPTION
Unity.Mathematics.float4	

Methods

CompareTo(ImGlyph)

Declaration

```
public int CompareTo(ImGlyph other)
```

Parameters

TYPE	NAME	DESCRIPTION
ImGlyph	other	

Returns

TYPE	DESCRIPTION
System.Int32	

Operators

Implicit(Char to ImGlyph)

Declaration

```
public static implicit operator ImGlyph(char c)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Char	c	

Returns

TYPE	DESCRIPTION
ImGlyph	

Implements

System.IComparable<T>

Struct ImString

Unsafe representation of a string. The struct does not implement an IDisposable interface because the purpose of the struct is to "borrow" a reference to a string's pointer. This is typically used in conjunection with ImWords or a fixed string. ImWords

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui.Text

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImString
```

Constructors

ImString(Char*, Int32)

Allows passing a pointer and length. This is generally used in conjunction with the TextBuffer.

Declaration

```
public ImString(char *ptr, int length)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Char*	ptr	
System.Int32	length	

ImString(String)

Pins a string using a fixed statement and stores the pointer to the head and stores the string's length.

Declaration

```
public ImString(string contents)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	contents	

Fields

Length

Declaration

```
public readonly ushort Length
```

Field Value

TYPE	DESCRIPTION
System.UInt16	

Properties

Item[Int32]

Declaration

```
public readonly ref char this[int index] { get; }
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	

Property Value

TYPE	DESCRIPTION
System.Char	

Methods

ToString()

Declaration

```
public override string ToString()
```

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.ValueType.ToString()

Class ImStringExtensions

Inheritance

System.Object

ImStringExtensions

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: InitialPrefabs.NimGui.Text

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class ImStringExtensions
```

Struct ImWords

Create a persistent buffer which stores all characters requested. This is a bump allocator and the data must be reset each frame.

Implements

System.IDisposable

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui.Text

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct ImWords : IDisposable
```

Constructors

ImWords(Int32)

Create a persistent buffer which stores all characters requested.

Declaration

```
public ImWords(int maxChars)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	maxChars	The maximum # of characters we can store.

Fields

Capacity

What is the maximum number of characters that the bump allocator can store?

Declaration

```
public readonly int Capacity
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

Dispose()

Frees the allocated fixed buffer.

Declaration

```
public void Dispose()
```

Request(Char)

Returns a string with a single character.

Declaration

```
public ImString Request(char c)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Char	c	The character, to request into the TextBuffer

Returns

TYPE	DESCRIPTION
ImString	A string with the character.

Request(Int32)

Returns an "empty" string with the requested size.

Declaration

```
public ImString Request(int size)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	size	The # of characters to request

Returns

TYPE	DESCRIPTION
ImString	A readonly "empty" string from the Words buffer.

Request(String)

Copies the contents of the string into the internal buffer and returns a ReadOnlyString that points to the memory's contents.

Declaration

```
public ImString Request(string text)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.String	text	The string to copy over

Returns

TYPE	DESCRIPTION
ImString	A readonly string from the Words buffer.

Request(StringBuilder)

Copies the contents of the string into the internal buffer and returns a ReadOnlyString that points to the memory's contents.

Declaration

```
public ImString Request(StringBuilder builder)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Text.StringBuilder	builder	The StringBuilder to read from.

Returns

TYPE	DESCRIPTION
ImString	A readonly string from the Words buffer.

Request(StringBuilder, UInt16)

Copies the contents of the string into the internal buffer and returns a ReadOnlyString that points to the memory's contents.

Declaration

```
public ImString Request(StringBuilder builder, ushort startIndex)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Text.StringBuilder	builder	The StringBuilder to read from.
System.UInt16	startIndex	The first index of the character of the slice.

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
ImString	A readonly string from the Words buffer.

Reset()

Resets the internal pointer to the buffer. This allows the buffer to be reused multiple times without reallocating new memory.

Declaration

```
public void Reset()
```

Implements

System.IDisposable

Class SerializedFontData

A project wide asset which stores the FontFace and potential glyphs that the font can render.

This should generally only be constructed in the Editor.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.ScriptableObject
SerializedFontData

Inherited Members

UnityEngine.ScriptableObject.SetDirty()
UnityEngine.ScriptableObject.CreateInstance(System.String)
UnityEngine.ScriptableObject.CreateInstance(System.Type)
UnityEngine.ScriptableObject.CreateInstance<T>()
UnityEngine.Object.GetInstanceID()
UnityEngine.Object.GetHashCode()
UnityEngine.Object.Equals(System.Object)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Vector3, UnityEngine.Quaternion)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Vector3, UnityEngine.Quaternion, UnityEngine.Transform)
UnityEngine.Object.Instantiate(UnityEngine.Object)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Transform)
UnityEngine.Object.Instantiate(UnityEngine.Object, UnityEngine.Transform, System.Boolean)
UnityEngine.Object.Instantiate<T>(T)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Vector3, UnityEngine.Quaternion)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Vector3, UnityEngine.Quaternion, UnityEngine.Transform)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Transform)
UnityEngine.Object.Instantiate<T>(T, UnityEngine.Transform, System.Boolean)
UnityEngine.Object.Destroy(UnityEngine.Object, System.Single)
UnityEngine.Object.Destroy(UnityEngine.Object)
UnityEngine.Object.DestroyImmediate(UnityEngine.Object, System.Boolean)
UnityEngine.Object.DestroyImmediate(UnityEngine.Object)
UnityEngine.Object.FindObjectsOfType(System.Type)
UnityEngine.Object.FindObjectsOfType(System.Type, System.Boolean)
UnityEngine.Object.DontDestroyOnLoad(UnityEngine.Object)
UnityEngine.Object.DestroyObject(UnityEngine.Object, System.Single)
UnityEngine.Object.DestroyObject(UnityEngine.Object)
UnityEngine.Object.FindSceneObjectsOfType(System.Type)
UnityEngine.Object.FindObjectsOfTypeIncludingAssets(System.Type)
UnityEngine.Object.FindObjectsOfType<T>()
UnityEngine.Object.FindObjectsOfType<T>(System.Boolean)
UnityEngine.Object.FindObjectOfType<T>()
UnityEngine.Object.FindObjectOfType<T>(System.Boolean)
UnityEngine.Object.FindObjectsOfTypeAll(System.Type)
UnityEngine.Object.FindObjectOfType(System.Type)
UnityEngine.Object.FindObjectOfType(System.Type, System.Boolean)
UnityEngine.Object.ToString()
UnityEngine.Object.name
UnityEngine.Object.hideFlags
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()

System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui.Text

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public class SerializedFontData : ScriptableObject
```

Fields

FontFaceInfo

Declaration

```
public ImFontFace FontFaceInfo
```

Field Value

TYPE	DESCRIPTION
ImFontFace	

Glyphs

Declaration

```
public ImGlyph[] Glyphs
```

Field Value

TYPE	DESCRIPTION
ImGlyph[]	

Class TextUtils

Inheritance

System.Object
TextUtils

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: InitialPrefabs.NimGui.Text

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public static class TextUtils
```

Methods

AlignHorizontally(in Single, in Single, in ImRect, in HorizontalAlignment)

Declaration

```
public static float AlignHorizontally(in float lineWidth, in float width, in ImRect rect, in
HorizontalAlignment alignment)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	lineWidth	
System.Single	width	
ImRect	rect	
HorizontalAlignment	alignment	

Returns

TYPE	DESCRIPTION
System.Single	

AlignVertically(in HeightInfo, in ImRect, in VerticalAlignment)

Declaration

```
public static float AlignVertically(in HeightInfo heightInfo, in ImRect rect, in VerticalAlignment alignment)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
HeightInfo	heightInfo	
ImRect	rect	
VerticalAlignment	alignment	

Returns

TYPE	DESCRIPTION
System.Single	

CountLines(in ImString, in UnsafeArray<ImGlyph>, in float2, in Single, ref NativeList<TextUtils.LineInfo>)

Declaration

```
public static void CountLines(in ImString text, in UnsafeArray<ImGlyph> glyphs, in float2 dimensions, in float scale, ref NativeList<TextUtils.LineInfo> lines)
```

Parameters

TYPE	NAME	DESCRIPTION
ImString	text	
InitialPrefabs.NimGui.Collections.UnsafeArray<ImGlyph>	glyphs	
Unity.Mathematics.float2	dimensions	
System.Single	scale	
Unity.Collections.NativeList<TextUtils.LineInfo>	lines	

GetStringHash(in ImString)

Declaration

```
public static uint GetStringHash(in ImString content)
```

Parameters

TYPE	NAME	DESCRIPTION
ImString	content	

Returns

TYPE	DESCRIPTION
System.UInt32	

GetStringHash(String)

Declaration

```
public static uint GetStringHash(string content)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	content	

Returns

TYPE	DESCRIPTION
System.UInt32	

Struct TextUtils.LineInfo

Inherited Members

System.ValueType.Equals(System.Object)
System.ValueType.GetHashCode()
System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: InitialPrefabs.NimGui.Text

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public struct LineInfo
```

Fields

Length

Declaration

```
public int Length
```

Field Value

TYPE	DESCRIPTION
System.Int32	

LineWidth

Declaration

```
public float LineWidth
```

Field Value

TYPE	DESCRIPTION
System.Single	

StartOffset

Declaration

```
public int StartOffset
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Enum VerticalAlignment

Namespace: InitialPrefabs.NimGui.Text

Assembly: InitialPrefabs.ImGui.dll

Syntax

```
public enum VerticalAlignment : byte
```

Fields

NAME	DESCRIPTION
Bottom	
Center	
Top	