What’s New with Plugins

Doug Banks
About Me

Doug Banks
CycloneUprising
About You

- How many have tried writing a plugin?
- How many have published a plugin?
- How many self-rate as “Expert” at writing plugins?
Overview

- Dock Widgets
- Plugin Actions
- What’s next
Dock Widgets
Dock Widgets

Plugins can now create GUI in docking widgets instead of in the 3d viewport.
Tired...

Plugin GUI in 3d viewport

- Obscures view of 3d space.
- All plugins compete for GUI space, GUI elements would overlay each other.
- Inconsistent look & feel, compared to other Studio tools.
Wired!

Plugin Gui in dock widget

- Each plugin can have its own separate GUI.
- Uses handy, familiar dock widget paradigm to manage families of controls.
- Look & feel similar to native widgets.
Using a Dock Widget

Allocate a DockWidgetPluginGuiInfo with initialization details:

```
DockWidgetPluginGuiInfo.new(InitialDockState initDockState,
  bool initEnabled,
  bool overrideEnabledRestore,
  int floatXSize,
  int floatYSize,
  int minWidth,
  int minHeight)
```
Using a Dock Widget

Create a new PluginGui using this descriptor:

```csharp
PluginGui CreateDockWidgetPluginGui ( string pluginGuiId,
DockWidgetPluginGuiInfo dockWidgetPluginGuiInfo)
```
Using a Dock Widget

Populate dock widget with GUI elements (just like a Screen GUI):

```lua
-- Put GUI elements into plugin gui.
local button = Instance.new("TextButton")
button.Text = "Hello"
button.Position = UDim2.new(0, 10, 0, 20)
button.Size = UDim2.new(0, 100, 0, 30)
button.Parent = pluginGui
button.MouseButton1Click:connect(function()
    print("Clicked!")
end)
```
Using a Dock Widget

Add a toolbar & button to toggle dock widget on/off.

Listen to “Enabled” state of PluginGui, keep toolbar button in sync with this state. (User may close Dock Widget using native controls)
Tips

Design for a dock widget (think like the Explorer):

- User may leave it on all the time.
- Not a “mode”
- Will be restored when user closes then reopens
Tips

Proper connection with toolbar buttons to toggle on/off.
Tips

Use our shared github libraries for GUI elements:
https://github.com/Roblox/StudioWidgets

- Consistent look & feel.
- We will be updating these to properly handle Dark Theme.
Limitations

UserInputService only deals with main 3d viewport.
Limitations

Can’t do “flyover” GUI elements (combo box, tooltips): anything that would go beyond borders of dock widget.
Example- New Terrain Tools
Example- Animation Editor
Plugin Actions
Plugin Actions

Plugins can create “Actions”, as understood and exposed by the keyboard and mouse shortcuts.
Old School…

Adding your own keyboard shortcuts in plugin code.

- May collide with shortcuts in other plugins.
- Users can’t see your mapping when viewing global list of keyboard shortcuts.
- Users can’t easily remap to preferred shortcut.
- If user maps a native action to this shortcut, the key presses may not ever get passed along to lua.
- Only works if 3d viewport has focus.
- Doesn’t work with PluginGui (no UserInputService)
So Cool!!

Create PluginAction which can be bound to keyboard shortcuts.

PluginAction ties in with native Shortcut support, so:

- No collisions.
- Exposed and customized along with all other actions.
- Keyboard shortcuts work as long as Studio app has focus.
Using Plugin Actions

local pluginAction = plugin:CreatePluginAction("printHello", "Print Hello", "This will print Hello.")

pluginAction.Triggered:connect(function()
    print("Hello")
end)
Using Plugin Actions
Friendly reminder!

When adding a PluginAction, consider integrating with ChangeHistory service so users can Undo/Redo your action!
What’s Next
What’s next?

- Update to UserInputService so that key-related stuff works with PluginGuis.
- StudioTheme API:
  - Query font, color, weight, etc for backgrounds, borders, etc.
  - Event when theme changes.
All Done
Summary

You can make your plugins look & feel like native Studio features:

● House the GUI in a Dock Widget.
● Use shared library of GUI elements to style-match Studio design & color themes.
● Use Plugin Actions to add shortcut support.
● Stay tuned for more!
Get Involved!

- Update your old plugins!
- Write some new plugins!
- Use and extend our library of “Studio Look & Feel” widgets on github! https://github.com/Roblox/StudioWidgets
- Dream big! Anything we can do, you can do better....
Re-implemented entire Explorer