Easy & Efficient Graphical User Interface (GUI) Creation for TI-Nspire™ Lua scripts

Jim Bauwens & Adrien Bertrand
(Presented by Adrien Bertrand)

Plan

- The Project Builder
- "Enhanced Lua"
- ETK 4
- Widgets and event-based scripting
  - Examples
  - Binding actions
- Whole UI demo
- What's next?
- Contributing

The *Project Builder*

What is it? It's a script that makes a tns from your Lua code

But it's also much more - it allows you to:

- ✓ Organize your project into several files
  Working with several classes? Say no to a "one-big-file" mess!
- ✓ Have an "include" system (like in other languages)
- ✓ Use "Enhanced Lua" with syntax additions
- ✓ Check and improve your code *(with linting and static analysis)*
- ✓ Output Lua code optimized the way you want it

The *"Enhanced Lua"*

It's the "normal" Lua, but with syntax additions. Examples:

Increment / Decrement

```
counter = counter + 1
```

```
counter++
```

String indexing

```
str2 = str1:sub(1,1)
```

```
str2 = str1[~1]
```

Bit operations

```
a = xor(a, b)
```

```
a ^= b
```

Lambda functions

```
myButton.onAction = function() myInput.value =
                                myInput.value + 1 end;
```

```
myButton.onAction = λ -> myInput.value++;
```

ETK 4

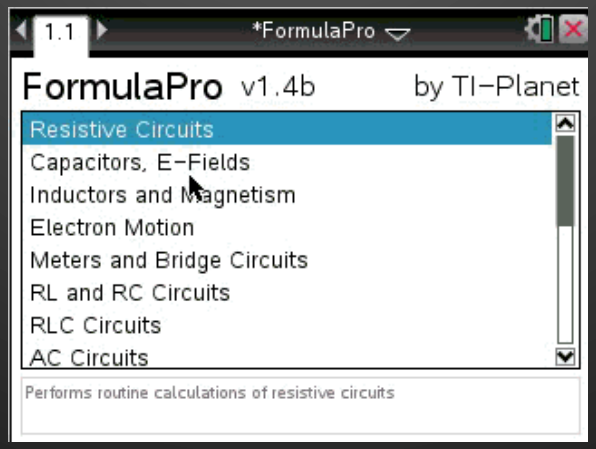
What is it?
A *framework* to quickly & easily build graphical scripts with, thanks to predefined elements (widgets) ready to be used.

Why would you use it?

- ✓ No more messing around with lots of code
- ✓ It provides an intuitive and an OS-like GUI to the user
- ✓ Easy & practical APIs and coding style for the dev.
- ✓ You can finally focus on your content - don't lose time on reinventing the wheel anymore ;)

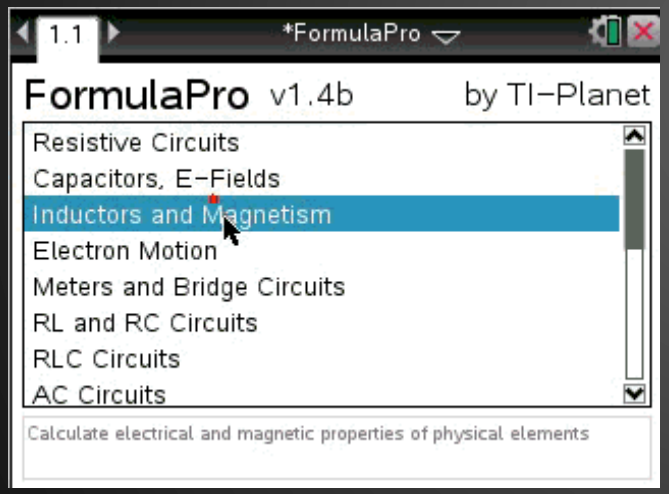
Widgets and event-based scripting

- Widgets are pieces of GUI elements that the user can interact with. They can be used to create forms, views, etc. to dynamically present data etc. to the user.
- Widgets in ETK work together in the same familiar event-based way you already know from the Nspire API: whenever an event is triggered, they can respond to it.
- This allows you to create powerful interactive content.



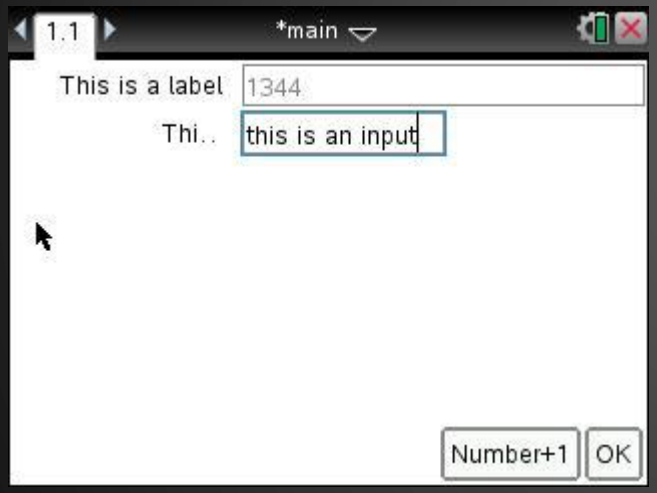
Widgets: *Frame*

A frame is like a container. It should hold together widgets (or even other frames) that are part of the same "group".



- (Root) Frame *(background)*
  - Labels *(the ones on top)*
  - List
    - List elements *(styled labels)*
    - Scrollbar
  - Textbox *(at the bottom)*

Widgets: *Buttons, Labels, Inputs*




```
button2.onAction = λ -> input1.value++;
```

```
local button1 = Button {
    position = Position {
        bottom  = "2px",
        right = "2px"
    },
    text = "OK"
}

-- ... Code for input1 ...   With:  input1.number = true

local input2 = Input {
    position = Position {
        top  = "4px",
        left = "0px",
        alignment = {
            {ref=input1, side=Position.Sides.Bottom},
            {ref=input1, side=Position.Sides.Left}
        }
    },
    value = "this is an input"
}

local label1 = Label {
    position = Position {
        top  = "2px",
        right = "10px",
        alignment = {{ref=input1, side=Position.Sides.Left}}
    },
    text = "This is a label"
}
```

Widgets: *Dialog (with sub-widgets)*


```
button1.onAction = function ()
    local dialog = etk.Dialog("Test Dialog",
        Position { top="40px", left="20px" },
        Dimension("-40px", "-80px")
    )

    local nameLabel = Label {
        position = Position { top = "30px", left = "4px" },
        text="Name: "
    }

    local nameInput = Input {position = Position { top = "30px", left = "50px"}}
    nameInput.dimension.width = "-54px"

    local closeButton = Button {
        position = Position { bottom = "4px", right = "4px" },
        text = "Close"
    }

    closeButton.onAction = function()
        input2.value = "Hi " .. nameInput.value
        etk.RootScreen:popScreen();
    end

    dialog:addChildren(nameLabel, nameInput, closeButton)

    etk.RootScreen:pushScreen(dialog)
end
```

Example: a whole UI demo

(see the demo .lua and .tns files)

What's next?

OpenSpire

Online Nspire-Lua development kit:

- ✓ The project builder and everything that comes with it
- ✓ A WYSIWYG GUI editor - no more interface code ;-)
- ✓ Real-time simulator
- ✓ Code autocompletion with syntax highlighting
- ✓ Templates (code and interfaces)
- ✓ Language help (Lua + Nspire-specific API)
- ✓ etc.

Compatible with all devices (made in HTML5/CSS3/JS)

Contributing

The Project Builder and ETK 4 are open-source and are available to everyone on: **https://github.com/TI-Planet/ETK**

We do welcome everyone willing to help, at any level:

- suggestions/feedback
- code
- testing
- …

Also, if you make/made an app with ETK, do share it!

Thank you!

Any questions?

More info and news about ETK 4 and the GUI builder will be available on:

`TI-Planet.org` and **`Inspired-Lua.org`**

You're welcome to ask for help there and/or by email:
bertrand.adrien@gmail.com ; jimbauwens@gmail.com