|  |  |
| --- | --- |
| 4D SUMMIT 2020  User Interface Modernization with Ease  Presented by: **Gabriel Inzirillo** | page1image27590656 |

Introduction

4D application may look from another time from the point of view of young developers and if we look at the design of certain applications, we cannot totally disagree with them.

4D have a great legacy which can be its strength but can also be its weakness sometime. We can build great things with 4D in a small amount of time. But sometime the functionalities and speed of development is not enough to convince new people to use your product. People want something that looks great. There is no well-known application today that doesn’t get full redesign time to time. We have more and more web application that come with some complicated interface and people want to get something that looks fresh even in a desktop app. It is not because it is a business application that it doesn’t have to look great.

State of today

Today in 4D we have more existing applications than new applications. Some exists more than 20 years ago and the modifications in terms of interface are sometime near to zero. Those applications will not survive if some work is not done. Your client may be satisfied with the functionalities and like the application, but when new people come in the company and see an old tool, they will fight it and try to replace it. Existing applications must continue to satisfy existing users but must attract new clients and users to use it.

In this matter, modernizing your interface is a must.

For new application it is important to start with a well-designed interface to make your application a concurrent to existing ones.

Here are some points that 4D developers may say :

* We don’t have time allowed to be spent in the design of our forms.
  + Time must be allowed to developers to design their forms, it is not something that take 10 minutes and that has to be done quickly, it may be done in several steps during the development and you should not be afraid to rethink the interface when adding functionalities.
* It is a business application and doesn’t need to be sexy
  + If it is a business app, it must be sexy to attract clients and to continue your business.
* We are developers and not designers
  + It is important to be able, as a 4D developer, to be able to code, but also to build good interface. 4D developers are multi-stack, it is very rare that a 4D developer only write code, it exists but it is rare. So knowing how to build a. great interface is important.
* There was nothing in the specifications about design and interface
  + As pointed in the previous point, you must be able to translate specifications into a. well-designed interface that will make the end user happy.
* The end user never complains about the design
  + It is too late when the end user starts to complain about the design, they are maybe already looking to a more modern solution to replace your product.
* 4D form objects doesn’t allow us to build nice forms
  + Well, this is not totally false, 4D has a long legacy and certain form objects should not be used anymore. You must know how to use the modern form objects and how to build a nice interface.

Why interface is important?

Interface is important because in 2020 clients may like the functionalities of your application but can be repelled by a not modern design.

If your application is in place by many years now, newcomer in the company may not like an old fashion software and will put a bad spirit on your software.

Not only for design but also because old fashion designs don’t meet today priorities. A form with dozens of fields on top of each other doesn’t help the user to focus on the important information. Also opening too many windows will confuse the user, it is important to be able to navigate via a single window, for most of the cases it is enough.

A well-designed interface must help the end user to focus on the important information, we should not overload the form with information that will lose the user’s attention. Sometime we need to change the design during development because we get feedback from the end user and specifications was not good enough to cover all the details to provide the best design at first.

How can we help You?

At AJAR SA, we develop a suite of components that will help you to build nice forms. This suite of components are grouped under the name AJUI\_Suite.

Here is the list of the components :

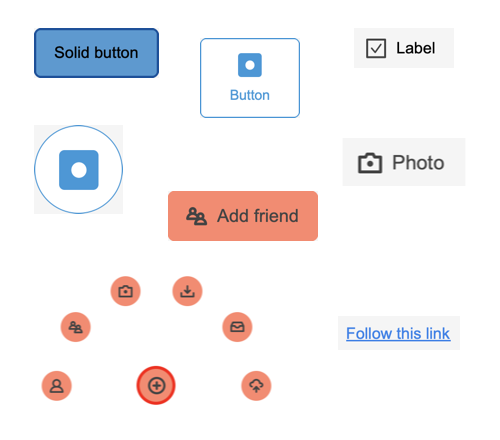
AJUI\_Button

Create full customizable buttons. Have control over the border, background, label and icon colors, control the border radius and size and much more options to design your application buttons to fit your design. You can define a specific design for 5 states : default, hover, activated, disable and focus. Those state follows the same logic as in CSS. Each state has the same values as the “default” state, you can override values by specifying them. For example, you only want a lighter background color for the on hover, you can define only the background color for the hover state.

The button is then rendered via SVG and displayed inside a picture variable that you have drawn in your form. It can handle all the events as a standard button.

A Lab application contains many How Do I and a complex lab that helps you to create templates to be reused easily in your applications. It shows an example on how to make your button tabable and to use the space bar (or other key) to trigger the on click.

The last added feature is called Floating Action Button (FAB), it allows you to define some buttons that will be displayed only when you click on one main button. Those new kinds of buttons can save you some space in your form to display button only when necessary.



1 Button Examples

AJUI\_Tip

As for AJUI\_Button, AJUI\_Tip help you to manage tips in your application by controlling every aspect. You can even go further by including your own form inside a tip to provide a fully new way of entering data. Some animations can be used to force the focus of the user on something. A fade in, fade out, jump and blink effect can be defined. An auto hide timer can be set to auto hide after a certain amount of time.

The tip will be created via a subform object that you will have to put in your main form. It will be used to create as many tips as you need. The tip is rendered via SVG.

A Lab is also included that contains many HDI and a Lab that helps you to configure tips and create templates.



2 Tip Examples

AJUI\_Breadcrumb

AJUI\_Breadcrumb is a complex object that can contain different individual sections (steps), each section can have its own type. 5 different types can be used (standard, current, first, next and previous) each type has 4 states (default, hover, active and disable). Thanks to those configurations, you can create complex breadcrumbs that can cover many different situations.

We have 3 models of breadcrumb available :

Simple

It is a simple breadcrumb with a separator as text.

A close up of a logo

Description automatically generated

3 Simple Breadcrumb

Arrow

An arrow will be used as a separator.



4 Arrow Breadcrumb

Grouped buttons

Grouped buttons are special and can be used to behave as a kind of radio button with a more advanced look. It will automatically assign the “current” type to the selected section.

A screenshot of a cell phone

Description automatically generated

5 Grouped Buttons

A Lab is also included that contains many HDI and a Lab that helps you to configure breadcrumbs and create templates.

AJUI\_Banner

AJUI\_Banner will help you to draw 2 types of banner in your form.

### Window Banner

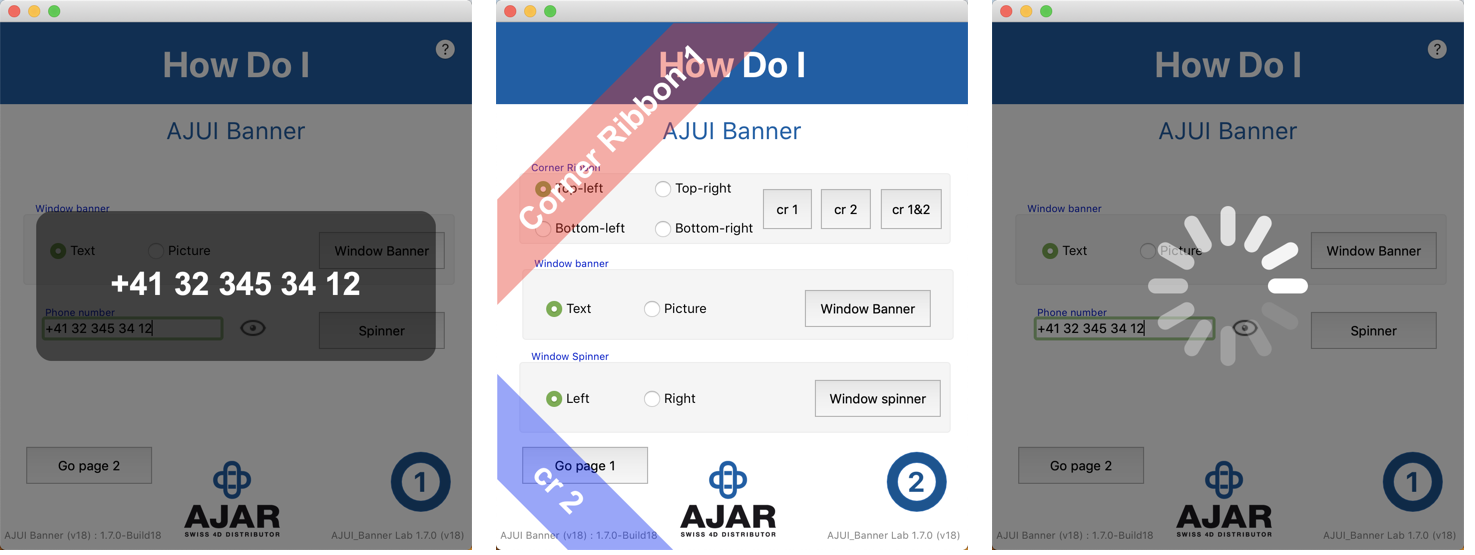
A window banner will cover your form with a certain opacity, and you will be able to show a message in the center. This is useful to force the attention on a specific information

### Corner Ribbon.

A corner ribbon will be a ribbon displayed in one corner of your form, it can also be useful to bring the user attention to something important.

### Spinner

Display a spinner to show that something is happening in the background.

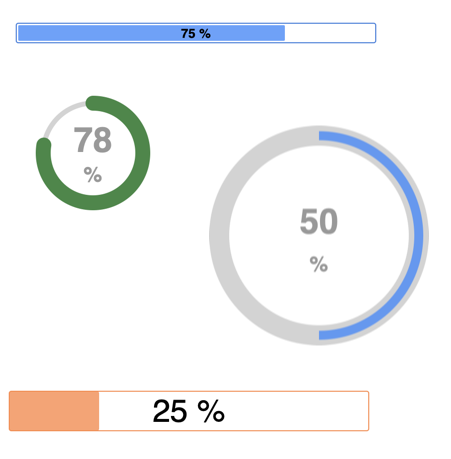


6 Banner Examples

AJUI\_Progressbar

Provide all the tools to draw fully customizable progress bar. Two types of progress bar are available, linear and circular.

Same as previous components it is rendered via SVG inside a picture variable.



7 Progressbar Examples

AJUI\_FloatingLabel

Floating label is a smaller component that will allow you to display a label only when data are already entered in a form field. If the field is not yet entered, the label will be a placeholder.

When the value is entered, the placeholder is not anymore visible and so the label is displayed.

All the Lab applications are open and you can check the code to see how the HDI are implemented.

## Using the object notation

Thanks to the object notation, we have designed all our components to work object oriented. Each kind of element will be created thanks to a “New AJUI\_...” method that will return an object containing all the properties and formulas available. You can then manipulate and display your element via the API available.

Here is an example with AJUI\_Button component.

**Case of**

**:** (**Form event code**=On Load)

**Form**.btn3:=***New AJUI\_Button***

**Form**.btn3.***Name***("btn3")

*//default*

**Form**.btn3.***BGColor***(AJUI\_btn\_default;"lightgrey")

**Form**.btn3.***Label***(AJUI\_btn\_default;"BTN 3")

**Form**.btn3.***BorderSize***(AJUI\_btn\_default;2)

*//hover*

**Form**.btn3.***BGColor***(AJUI\_btn\_hover;"darkgrey")

**Form**.btn3.***Label***(AJUI\_btn\_hover;"On Hover")

*//active*

**Form**.btn3.***BGColor***(AJUI\_btn\_active;"grey")

**Form**.btn3.***Label***(AJUI\_btn\_active;"On Click")

**End case**

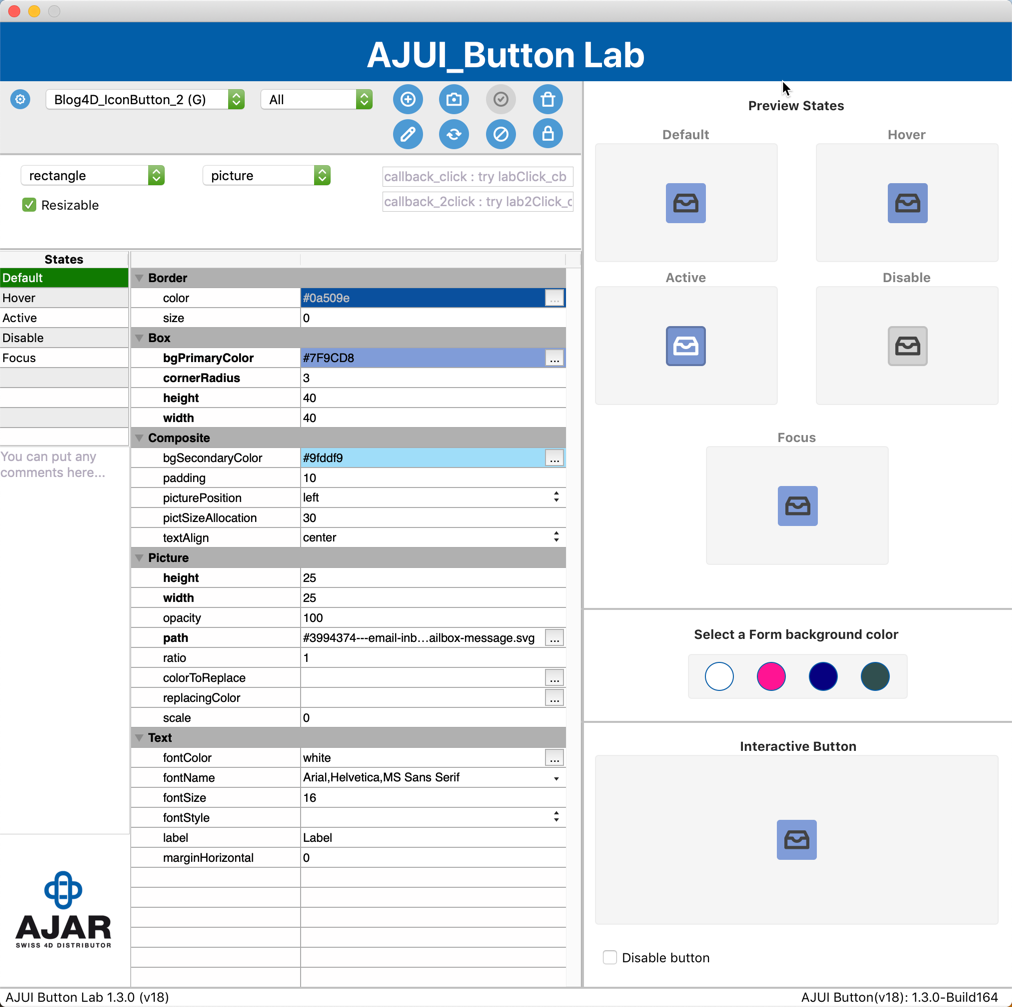
**Form**.btn3.***Draw***()

You can see that we store our button object inside “Form.btn3” and we define some properties via formulas. We provide formulas as it is easier to use than going through many levels of property, some formulas do heavier tasks and controls.

## Templates

AJUI\_Tip, AJUI\_Button and AJUI\_Breadcrumb Labs provide a sophisticated interface that help you to create your design with a direct feedback. You can save templates and import them directly in your application. This is really helpful not to have to define each property by code as some design can become complex. The template will be saved in json and you can place it in the default folder or in your specific folder in the resources of your application. If you are using pictures in your template, a folder will be created with all the pictures and the template. The template gonna save the path of the picture to be relative to the template location in your final application.

You can save each template you are working on it to reuse it later, thanks to the separation of the lab and the component, you will be able to simply drag and drop the new template in your application to modify instantly all the buttons of your application.



8 AJUI\_Button's Lab

Demonstration

During the demonstration we take a database example with some existing forms :

## OldForm

This form is an old “fake” form that works great and isn’t that ugly but we see together how we can improve it by using the different components from AJ suite and how to use them.

## Main Explorer

This is another example of an explorer built with AJ suite.

## Pay Station

This example shows a pay station at a shop. It tries to imitate what we have in the today’s shop when you can scan your own articles.