**UiEditor Manual**

1. **Introduction**

UiEditor is a program to create User Interface for game that uses BZBee in house game engine. It is mainly use by UI artist to create user interface document that is supported by the game engine. Total of 3 version of UiEditor has been released to artist to create UI layout. The first version was written in Flash and it is used in SD1. The second version was rewritten in C++ and it was using wxWidgets to ease the creation of the panels for the program. This version was used by Deck Heroes and later being enhance to Third version which will be use by our future projects.

1. **Menu Functions.**

As usual, each program will have its own menu, control panel and any functionality. In this section, we will explain some of the main panel in the program.

* 1. **Menu Panel**



There are a bunch of controls on this menu panel.

* + 1. File

New Project - It is **NOT SUPPORTED** in current version.

Open Project - Open a predefined projects (.proj) to edit.

Save Project - Save any modified ui layout (.ui)

Close Project - It is **NOT SUPPORTED** in current version.

Exit - Exit UiEditor.

* + 1. Edit

Undo - undo previous actions.

Redo - redo previous undo actions.

Copy - copy layout into clipboard.

Paste - paste layout from clipboard.

Replace - replace selected layout with the one in copied clipboard.

Delete - delete selected layout.

Make Best Size - make best fit to any layout with text. **(Note: this functions work best for layout with text, layout without text will be scale to its original size read from style sheet).**

Align - Functions to align layout.

Layer - Functions to swift layout in layer hierarchy.

* + 1. Dialogs

**Note: This is the place where you add new ui layout to the project. Ui layout are called Dialogs in this program.**

Add Dialog - Add new ui layout into project.

Remove Dialog - Remove selected ui layout from project.

Rename Dialog - Rename selected ui layout.

Comment Dialog - ???

* + 1. Attribute

Add Attribute - Functions to add attribute to selected element in a layout. **(Note: uint, int64, uint64 doesn’t work)**

Remove Attribute - Remove selected attribute from the layout. **(Note: only those dynamically added attribute)**

* + 1. View

Zoom In - Zoom in preview window.

Zoom Out - Zoom out preview window.

Zoom 100 - reset to default zoom distance.

Set Background - ???

Clear Background - ???

* + 1. Tools

Preview - Preview the selected layout with animated elements, particles or 3D meshes with animation.

Preview en\_us - Preview layout using English translated words.

Preview zh\_cn - Preview layout using Chinese translated words.

Show Selection - Toggle display to show selected layout element with a red bounding box.

(Show Current

Using Textures) - Show texture used by the selected layout element in a popup dialog **(Note: It doesn’t work for this version)**

(Duplicate

Windows) - ???

(Rename

Selections) - Rename selected layout element.

(Generate

Binding Code

C++) - Generate C++ binding code for the selected layout element. The generated code will be use by ui programmer to implement any binding functions in the game.

(Get Control

Path) - Functions to get selected layout element pointer path. This function is to ease ui programmer to find out the relationship of the layout hierarchy of this element.

(Generate

Charset) - ???

* + 1. Help

About - Information about this version.

1. **Control Panel (Panel on the Left)**

In this panel, there are 3 different tabs for different types of control.

Dialogs - This tab shows all available ui saved in this projects. When new dialog (ui layout) is added, it will be added to this hierarchy in the project.

Layer - This tab shows the hierarchy of elements for the selected ui layout. Layer hierarchy can be changed by using “Menu->Edit->Layer”.

Controls - This tab shows all supported control that can be used to create the ui layout. Just Drag and drop the selected control to the preview window to add it to the layout.

**4. Preview Panel (Panel on the center)**

This panel will show any selected ui layout in Dialogs tab for ui artist to design the ui. Drag and drop any controls from the Controls tab to add layout element. Ui artist can move any selected element in this panel using mouse click. **It is highly recommended** to use the property panel to enter the precise value for each element instead of dragging with mouse.

**5. Property Panel (Panel on the Right)**

When user select a particular element for a ui layout in Layer Tab, those available properties for this element will be populated in this property panel for manipulation. Different types of ui element has its own properties set.

**6. Technical Usage**

There are few things that ui programmer need to take notes before using UiEditor.

* 1. Things to take note before using UiEditor.

UiEditor doesn’t support creation of a new project. Which means a predefined ui.proj needs to be created before we can start to add more layout to the project.

Ui.proj is an xml base document with .proj as extension. Example of data store in .proj file as follow:



Resource Tag: this tag stored predefined path for those resources required by the UiEditor.

sprite\_dir, mesh\_dir, skinMesh\_dir, actions\_dir, fonts\_dir, particle\_dir are the predefine directory for the specific file types.

ImagePieces - atlas information of an image.

NineGridStyle - atlas information of an image.

BitmapStyle - states for a bitmap using ImagePieces id.

ColorStyle - predefined color state.

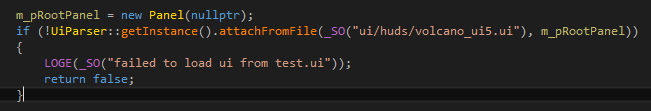
Huds - all ui layout for this project.

We need to have a predefined ui.proj file with at least Resources information to start using UiEditor.

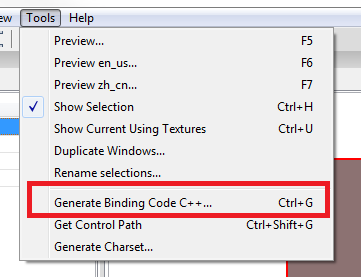
* 1. Implementing ui into program.

When a ui file is done by artist, ui programmer needs to implement and bind this ui into game with necessary controls functions.

We can start by loading the ui into reference pointer using some helper functions in the engine. Eg:



Now, we can start to register event trigger to the ui element in this file. Some of the code can be generated by UiEditor using Generate Binding code C++

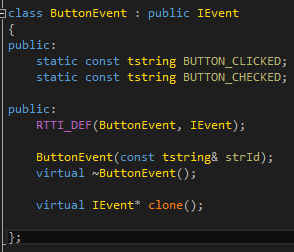


Register event like the example below.



Check the class implementation to find out more about the available event. You can create your own event by following the same blueprint of the event class.

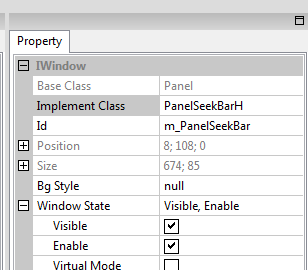
Example event blueprint as below:



* 1. Implement Class

Other than using default controls available in UiEditor, ui programmer can extends those available controls and add more functionality to it. To do this, ui programmer needs to ask the artist to fill up the extended class name into the ui element properties “Implement Class” field.

Example of Implement Class Property:



Ui programmer need to implement this class and inherit the base class to create functions to manipulate the behaviors.

Over write the virtual class to create different result from the base class. For more functions that can be overwritten by this implement class, please check the Base Class for more information.