

## Chapter 1:: Our First AUI Application.

A **Graphical User Interface (GUI)** is a visual interface that allows users to interact with software through graphical elements such as buttons, labels, text fields, and windows. The **event loop** is a core mechanism that drives GUI applications. When a GUI application starts, it initializes the interface and enters a continuous loop, waiting for events like mouse clicks, key presses, or system updates. Each time an event occurs, the event loop captures it and dispatches it to the relevant part of the program, often triggering specific actions like executing a function or updating the UI. For example, when a user clicks a button, the event loop detects this click event and calls the function associated with that button (often through a callback or event handler). This event-driven model allows GUIs to be responsive, as the program doesn't run in a linear sequence but instead reacts to user inputs in real-time. The event loop keeps the application alive and responsive until the user decides to close the window, at which point the event loop ends and the application terminates.

```
MyWindow() : Awindow("Sample Window", 700_dp, 200_dp)
```

In this example, an instance of `MyWindow` is created and displayed using the `show()` method. The `AUI_ENTRY` macro marks the entry point of the application, basically providing the event loop for the GUI.

```
AUI_ENTRY {
    _new<MyWindow>()->show();
    return 0;
}
```

Here is our first Window made with AUI:

```
#include <AUI/Platform/Entry.h>
#include <AUI/Platform/AWindow.h>
#include <AUI/Util/UIBuildingHelpers.h>
#include <AUI/View/ALabel.h>
#include <AUI/View/AButton.h>
#include <AUI/Platform/APlatform.h>

class MyWindow : public AWindow {
public:
    MyWindow() : AWindow("Example Window", 700_dp, 200_dp)
    {
    }

};

AUI_ENTRY{
    _new<MyWindow>()->show();
    return 0;
}
```

Once We have compiled everything, we will see our first AUI Application in action.

