Controls and Containers

Adding controls such as buttons and containers to store these controls

Adding Controls and Containers

```
1   #pragma once
2
3   #include <wx/wx.h>
4
5   class Frame : public wxFrame
6   {
7   public:
8       Frame();
9
10  private:
11      DECLARE_EVENT_TABLE()
12
13      wxPanel* panelOne;
14      wxPanel* panelTwo;
15
16      wxButton* exitButton;
17      void onExit(wxCommandEvent&);
18      wxButton* okayButton;
19      wxButton* emptyButtonOne;
20      wxButton* emptyButtonTwo;
21  };
```

Frame.h

Creating and Initializing Objects and Containers in wxWidgets

1) Set your desired variables within your frame's class (as pointers)
2) Initialize these variables within your frame's source file (Frame.cpp)
    a) wxPanel's constructor arguments
        i) this **~ Parent window, or rather the frame this panel is within**
        ii) wxID_ANY **~ Inactionable, any ID may be set**
        iii) Point on the screen and size
            (1) For any program with only such variable, such as a panel or a button will take up the whole screen automatically.
    b) wxButton's constructor arguments
        i) panelOne / panelTwo **~ Where we're placing our button**
        ii) wxID_ANY **~ Button's ID, buttons with no action have no ID**
        iii) "emptyButtonOne" **~ What the button's text will be**

Event Tables

Creating and Adding Events to an Event Table

1) Declare an event table within your frame's class
    a) Within Frame.h, DECLARE_EVENT_TABLE() does this
2) Initialize your event table within your frame's source file
    a) Within Frame.cpp, lines 3 and 5 start and end this event table.
    b) (**Line 3**) Explaining arguments
        i) Frame **~ Class (frame) connected to this event table**
        ii) wxFrame **~ Class classification**
3) Add a unique ID to an object, and connect this ID to a function
    a) Within the variable **exitButton**, wxID_EXIT (a present wxWidgets ID) is placed, as it's function will be to exit the program
    b) We create a function for the button **onExit**, with a simple close command
    c) Within line 4 (**EVT_BUTTON(...)**), we connect our desired function with the ID we want associated to it.