Sizers

Introduction to Sizers in wxWidgets

Configuring a Sizer

(1) Constructors

wxBoxSizer

- Box sizers only has one constructor, and it's one argument is wxVERTICAL or wxHORIZONTAL. This means that our sizer will automatically stretch either vertically or horizontally by it's new default.

wxGridSizer

- wxGridSizers have many constructors, however on **line 130** of **Frame.cpp**, we used the constructor with the arguments: rows (2), columns (2), vertical padding (1), and horizontal padding (1).
  - Rows: Sets the rows of the sizer.
  - Columns: Sets the columns of the sizer.
  - Vertical padding: Spacing between vertical neighbour objects.
  - Horizontal padding: Spacing between horizontal neighbour objects.

(2) Adding Objects

For adding objects to a sizer, **Add(...)** will typically be used. In our program, adding an object to one of our sizers contains the arguments: The object itself, proportional resizing, styling flags, and a border width (usually).

On **line 21** of **Frame.cpp**, we may visualize adding an object to a box sizer:

1. We add our wxButton object.
2. it's proportional resizing argument is 0, meaning that it's size won't change when the window is resized (vertically in our box sizer's cases)
3. wxFIXED_MINSIZE | wxALIGN_CENTER ~ Styling flag options
   1. wxFIXED_MINSIZE: Fixed size
   2. wxALIGN_CENTER: Centered
4. And finally, our border width is 10 for the sizer

On **line 131** of **Frame.cpp**, we may visualize adding another sizer to in this case, our grid sizer:

- After our styling flags, we do not need a border width (although you may add one) since our sizers already have one.

(3) Setting a Sizer

SetSizer(sizerName) and **SetSizerAndFit(sizerName)** are the main two commands used to set a sizer to your window. SetSizer will place the sizer according to your window width and height, while SetSizerAndFit will place the sizer and resize the window according to fit the sizer objects perfectly.