1

Outline

- In this lesson, we will:
  - Review frames and dialogs
  - Consider how to display content in these windows
  - Describe the ideal of widgets and panels
  - Observe that if both these objects are derived from the same base class that this will make our implementation much easier
  - This is a very useful application of polymorphism

2

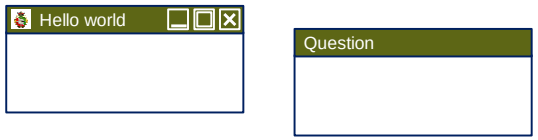
Warning

- Please remember, we don't expect you know how to implement a graphical user interface
  - This is the result of years of experience and a team of software engineers and developers working together
- We are presenting this as an example of where inheritance could be useful in the real world
  - Try to understand why we are suggesting and describing

3

A review of windows, frames and dialogs

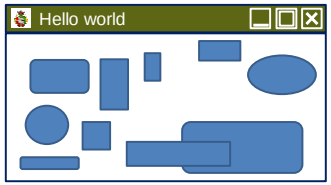
- We have seen how we can use inheritance to simplify our design of our frame and dialog windows
  - The base window class contains that information common to the two classes we discussed
  - Any changes to the base class is immediately shared to both classes
    - This ensure features are uniformly introduced
- However, we have yet to describe how content is stored



4

Displaying content

- In the panel, the programmer can include various *widgets*
  - How does the programmer lay these out?

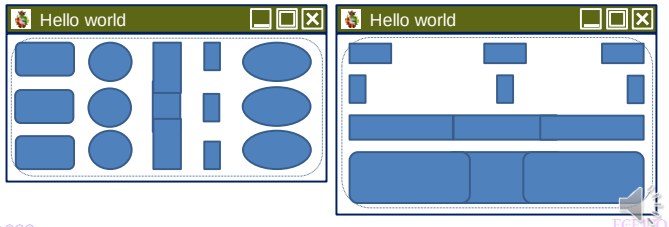


  - Does the programmer simply specify where each item goes?
    - This would be a nightmare for novice programmers and even experienced programmers
    - It would also lead to wildly varying and very inconsistent interfaces

5

Ordered display with panels

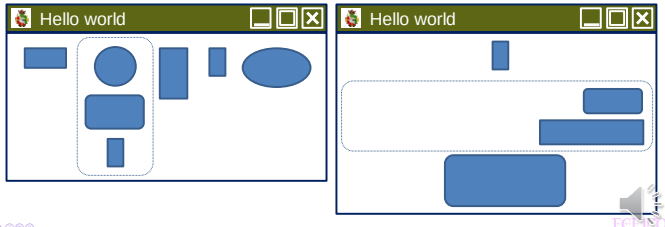
- Here is the approach:
  - The panel contains an ordered list of widgets
  - The widgets will be displayed in order with the window deciding how to lay these out
  - The panel has an *orientation*, either left-to-right or up-down
  - The user can set the alignment middle/center, top/left or bottom/right



6

Ordered display with panels

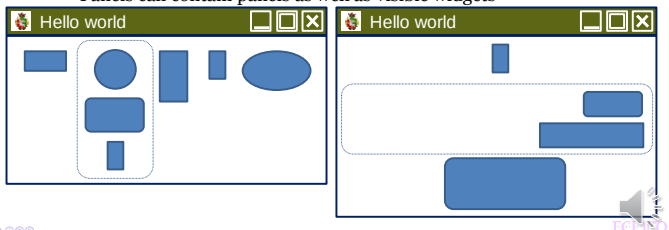
- One possible widget a panel can contain is another panel



7

Ordered display with panels

- Before we continue:
  - Both window frames and window dialogs contain a single panel
  - A panel contains an ordered list of widgets
  - The panel has an orientation, either horizontal or vertical
  - Within the panel, any widgets may be aligned
  - Panels can contain panels as well as visible widgets



8

The widget class

- The widget class would be

```
class Widget {
    public:
        virtual void display() const = 0;
};
```

- This is an abstract class and cannot be instantiated
  - A derived class must override the `display()` member function if that class is to be instantiated

The panel class

- We could implement the panel class as follows:

```
class Panel : public Widget {
    public:
        Panel();
        virtual void display() const override;
        virtual void append_widget( Widget *p_new_widget );
        virtual void set_orientation( … );
        virtual void set_alignment( … );

    protected:
        Widget     *widget_array_[128];
        std::size_t widget_count_;
        bool        horizontal_orientation_;
        int         alignment_;  // 0      left/top
                                 // 1    center/middle
                                 // 2     right/bottom
};
```

The panel class

- Here is the constructor defined:

```
Panel::Panel():
widget_count_{ 0 },
horizontal_orientation_{ true },
alignment{ 1 } {
    // Empty constructor
}
```

The panel class

- Here are some of the member functions defined:

```
void Panel::append_widget( Widget *p_new_widget ) {
    if ( widget_count == 128 ) {
        throw invalid_argument{ "Panels limited to 128 widgets" };
    } else {
        widget_array[widget_count] = p_new_widget;
        ++widget_count;
    }
}

void Panel::display() const {
    for ( std::size_t k{0}; k < widget_count; ++k ) {
        widget_array[k]->display();
    }
}
```

Possible widgets?

- What are possible visible widgets?
  - A *check box* contains text and has a state of being selected or not
  - A *toggle button* contains text and has a state of being selected or not, and only differs from the check box in appearance
    - These both have Boolean values
  - A *slider* allows the user to select one of many differ integral values
  - A *spinner* allows the user to cycle through a number of values
    - These both have integer values
  - A *drop-down menu* allows the user to chose one of many entries from a list
  - A *combo-box* allows the user to either chose one of many entries from a list or to also enter a value
    - These both have string values

Possible widgets?

- Those widgets that store a state can thus be accessed

```
class Boolean_widget : public Widget {
    public:
        bool value() const = 0;
};

class Integer_widget : public Widget {
    public:
        int value() const = 0;
};

class String_widget : public Widget {
    public:
        std::string value() const = 0;
};
```


Possible widgets?

- Therefore:
  - The check box and toggle button classes would extend the `Boolean_widget` class
  - The spinner and slider classes would extend the `Integer_widget` class
  - The drop-down menu and combo box classes would extend the `String_widget` class
- Each class that is derived from one of these must implement the `display()` and `value()` member functions

- To add such a widget to the content of a frame or dialog:
  - Dynamically allocate the memory for the widget and initialize it
  - On the appropriate panel, call `append_widget( … )` with the address as the argument
  - Similarly, a panel could be dynamically allocated and or appended

Important

- Please note, this presentation is slightly naïve of some of the more modern means of creating such classes
  - These examples have been significantly simplified in order to help in understanding how inheritance can be used in programming
  - Hopefully you understand the need for having a base widget class, a class from which such differing classes such as panels and check boxes can be derived

Summary

- Following this lesson, you now
  - Understand a common approach to displaying graphical widgets on the screen
  - Know that a panel class groups widgets and displays them either horizontally or vertically with an orientation
  - Understand that both the panel class and all other visible widget classes are derived from an abstract widget class
  - Are aware that classes need to override the `display()` and in some cases the `value()` member functions in order to make it possible to create instances of such classes

References

[1] https://en.wikipedia.org/wiki/Graphical_widget

Colophon

These slides were prepared using the Georgia typeface. Mathematical equations use Times New Roman, and source code is presented using `Consolas`.

The photographs of lilacs in bloom appearing on the title slide and accenting the top of each other slide were taken at the Royal Botanical Gardens on May 27, 2018 by Douglas Wilhelm Harder. Please see

https://www.rbg.ca/

for more information.



Disclaimer