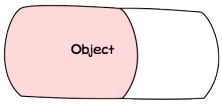
Graphical User Interfaces: Event-Driven Programming

by Andrew Cain and Willem van Straten

Objects are a great way of encapsulating functionality



This makes objects a great way of distributing functionality to others

Classes that do ...

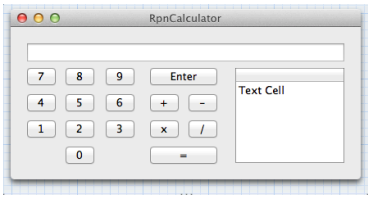
Learning to use other's libraries will make development much easier

Use these classes to help you build...

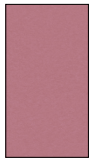
Many libraries are available to help build Graphical User Interfaces

Activity: What objects would you provide in a GUI library?

Use abstraction to think about the things GUIs may need



Are there any generalised or specialised roles you could add?

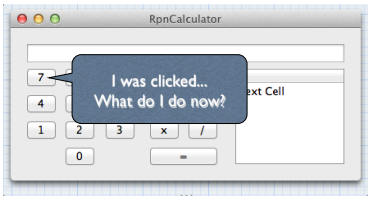


Inherits Object characteristics

Inherits Shape characteristics

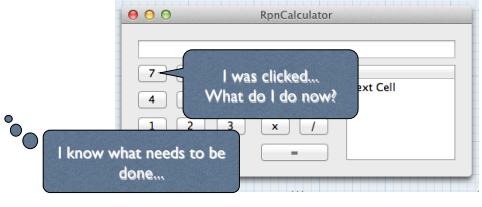
Includes Rectangle characteristics

What are the main issues in relation to handling user input?



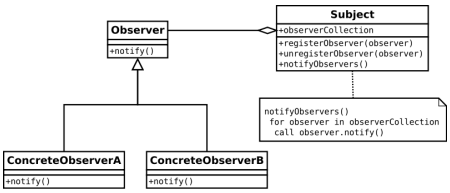
Can the OO principles help us to implement event handling?

GUI frameworks provide graphical objects, but actions come from their application



Activity: think up multiple ways this can be addressed

"Callbacks" are an example of the "observer" design pattern



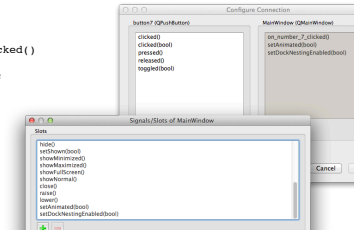
Examine how libraries implement callbacks and event-handlers

C# uses **delegates**, type safe method pointers

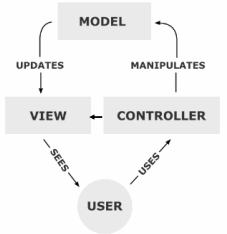
```
this.button1.Click += new System.EventHandler(this.NumberClicked);

private void NumberClicked(object sender, EventArgs evt)
{ ... }
```

C++ has a variety of different strategies e.g. slots and signals in Qt

```
void MainWindow::on_button7_clicked()
{
    number_clicked(ui->button7);
}
```



Model – View – Controller Separation of Concerns



Learn to use GUI frameworks:
1. learn to use class libraries
2. learn good design principles

Objects provide a natural packaging mechanism for distributing functionality

Class libraries provide a large range of reusable object classes

This Week's Tasks

Semester Test: Prepare for Final Opportunity

Portfolio Deadline: 4 November