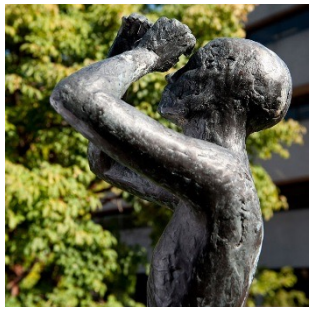


Lecture 17 – tkinter

CS2513

Cathal Hoare

A TRADITION OF INDEPENDENT THINKING

UCC

University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Lecture Contents

More TKInter

Other forms of Input

- As well as clicks on controls we can also handle key strokes and mouse clicks on the overall application window.
- For a mouse click - we can bind a mouse button to some action.

Function we bind to

Where we bind to the interface component

```
import tkinter as tk

def mouse_function(event):

    myCanvas.create_oval(event.x, event.y, event.x + 20, event.y + 20, fill='red')

root = tk.Tk()
myCanvas = tk.Canvas(root, bg="white", height=300, width=300)
myCanvas.bind("<Button-1>", mouse_function)

myCanvas.pack()
root.mainloop()
```

Organizing our code in an OO Way

- When we develop a TKInter App there are two phases:
  - Setting up the interface, laying it out and creating actions to execute
  - Providing actions to execute

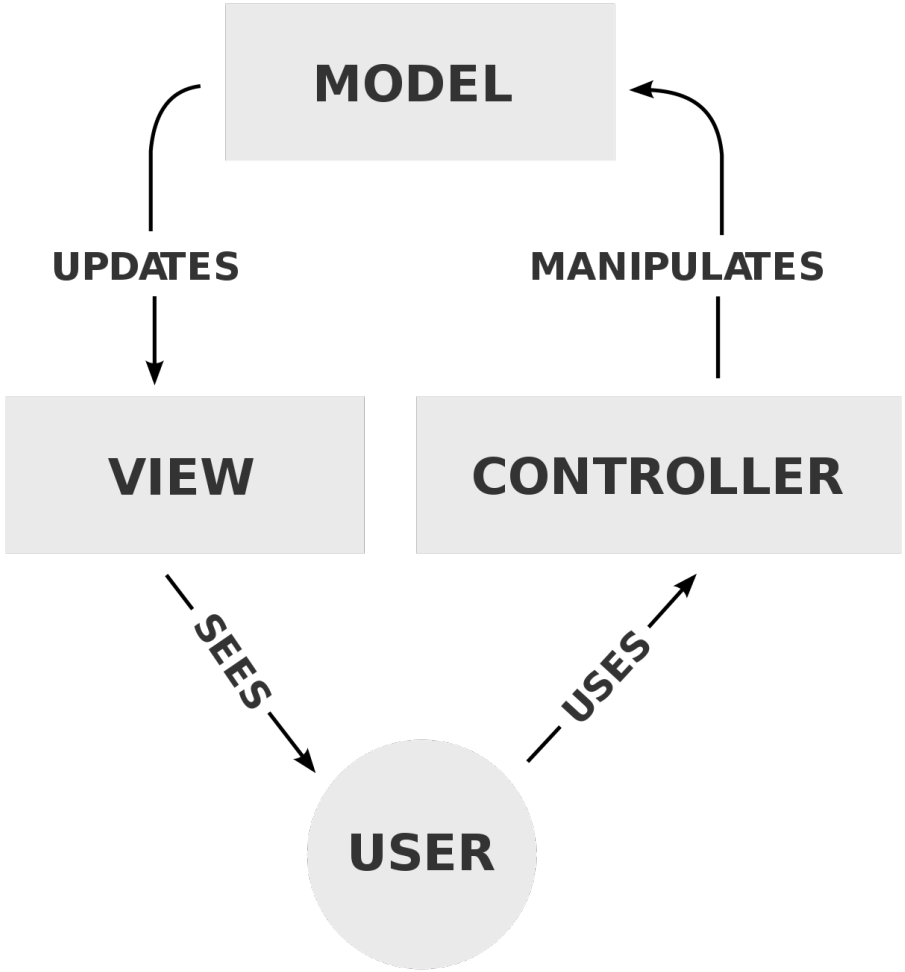
- This maps to a simple class definition:
  - Setup can be done wither entirely or by method calls from the constructor
  - The actions can be added as methods to the class.

Design Patterns

- One pattern is called Model View Controller
  - It seeks to achieve a ‘separation of concerns’ – essentially decoupling the data representation from the interface
  - This makes it easy to present different views of the data to different apps

- The *Model* is responsible for managing data in the application. It is manipulated by the controller and provides information to the view.
- The Controller receives inputs from the user, validates those and makes changes to the model.
- The View presents information to the user – several views of the same data can be presented

Design Patterns




Next Time:
Class Test