Outline

1. Graphical User Interfaces
   - a sliding puzzle
   - characteristics and components

2. The Tk GUI Toolkit in Python
   - tkinter, Tk, and Tcl
   - hello world again
   - using grid, Radiobutton, and Checkbutton
   - handling mouse events

MCS 260 Lecture 32
Introduction to Computer Science
Jan Verschelde, 19 July 2023
graphical user interfaces using tkinter

1 Graphical User Interfaces
   - a sliding puzzle
   - characteristics and components

2 The Tk GUI Toolkit in Python
   - tkinter, Tk, and Tcl
   - hello world again
   - using grid, Radiobutton, and Checkbutton
   - handling mouse events
A Sliding Puzzle
Elements of the GUI

The GUI for the sliding puzzle contains:

1. buttons to start, scramble, unscramble, and clear;
2. a canvas widget to display the sliding puzzle;
3. a list of lists stores the data of the puzzle;
4. the scramble/unscramble buttons are animations;
5. mouse events allow the user to solve the puzzle.

Functionality:

1. the start button of the GUI shows an ordered board;
2. the scramble button perturbs the current board;
3. pushing unscramble shows how to solve the puzzle;
4. the user can move rectangles by clicking on a rectangle that can slide to the free position.

Object oriented programming is used to implement GUIs.
1. Graphical User Interfaces
   - a sliding puzzle
   - characteristics and components

2. The Tk GUI Toolkit in Python
   - tkinter, Tk, and Tcl
   - hello world again
   - using grid, Radiobutton, and Checkbutton
   - handling mouse events
User Interfaces (UI)

GUI, WUI, and HUI

Interfaces to our programs so far were command line.

Consequences:

1. strict linear control of execution order imposed on the user by program;
2. user must remember a lot, e.g.: file names.

Modern interfaces are mostly *event driven* and give the user control over the actions.

Command line interfaces are call non-GUI. Specific categories of UI are

WUI: *Web UI* and HUI: *Handheld UI*.

Goal: increase *usability* of software.
Characteristics of GUls

The most significant features of a GUI are

**Window**: area of display device to view and interact with an object. Information is *viewed*.

**Icon**: pictorial representation of an object.

**Menu**: collection of choices, typically to perform actions on an object.

**Pointer**: is similar to a typing cursor, controlled by a pointing device, typically a mouse.

**Client Area**: area within a window where the user can enter data, strings or numbers.

**Directness**: direct manipulation of objects via the pointer, such as moving and dragging windows.

The development of GUls should be *user centered*. Goals: simple, aesthetic, productive, and customizable.
GUI components

GUI components are *widgets* = window gadgets.

Some GUI components are

- **Button**: to trigger an event
- **Label**: to display text or icons
- **Entry**: to accept single line user input
- **Text**: to accept multiple line user input
- **Menu**: to display list of items
- **Listbox**: to display list of text options
- **Canvas**: to draw geometric figures
graphical user interfaces using tkinter

1. Graphical User Interfaces
   - a sliding puzzle
   - characteristics and components

2. The Tk GUI Toolkit in Python
   - tkinter, Tk, and Tcl
   - hello world again
   - using grid, Radiobutton, and Checkbutton
   - handling mouse events
The `tkinter` (= Tk interface) library provides an object-oriented interface to the Tk GUI toolkit, the graphical interface development tool for Tcl, Tk = Tool Kit, Tcl = Tool Command Language.

Benefit: platform independent GUI development.
Graphical User Interfaces
- a sliding puzzle
- characteristics and components

The Tk GUI Toolkit in Python
- tkinter, Tk, and Tcl
- hello world again
- using grid, Radiobutton, and Checkbutton
- handling mouse events
Hello World!
our first GUI

Five steps:
1. `from tkinter import Tk, Label`
2. create a new object of the class Tk
3. a label defines the text message
4. apply the geometry manager to the label
5. enter the main event loop

Until we close the window, the program stays in the loop.
Dialogue with User – prompting for a name

We prompt the user to enter a name:

After the user typed in a name:

Pushing the enter button pops up the window:
Graphical User Interfaces

- a sliding puzzle
- characteristics and components

The Tk GUI Toolkit in Python

- tkinter, Tk, and Tcl
- hello world again
- using grid, Radiobutton, and Checkbutton
- handling mouse events
the Geometry Manager grid

To layout the geometry like this

Observe:

1. the title of the window
2. buttons spanning multiple columns
3. an entry field containing text
use of Radiobuttons

The GUI should do the following:

1. the entry field starts at 0
2. when user clicks on +1: add one
3. when user clicks on -1: subtract one
Using CheckButtons

Functionality:
1. the user can check one or two boxes
2. click on the enter button
3. to see a message displayed in the Entry box
graphical user interfaces using tkinter

1. Graphical User Interfaces
   - a sliding puzzle
   - characteristics and components

2. The Tk GUI Toolkit in Python
   - tkinter, Tk, and Tcl
   - hello world again
   - using grid, Radiobutton, and Checkbutton
   - handling mouse events
handling mouse events

Often the amount of data we generate is too huge for an orderly display in a classical terminal window. Much more data can be stored in an image on canvas and via the mouse we may interact with the data.
Exercises

1. Write Python code to display:

You should not provide any functionality.

2. Add functionality to the calculator shown above.

3. Design a GUI to convert temperatures between Fahrenheit and Celsius. Draw the layout and decide what widgets you will use.

4. Give Python code for the previous exercise.