Dr. Sarah Abraham
University of Texas at Austin
Computer Science Department

Event-driven Programming: GUIs

Elements of Graphics
CS324e

Event-driven Programming

- Programming model where code runs based on **events**
- Events occur asynchronously throughout program execution
  - System-generated events
  - User-generated events
- Some part of system signaled/messaged when event is triggered
- Change program flow based on user input, sensor output, or system messages

System-generated Events

- System initiates an event outside of user's control
- Generated by:
  - External hardware beyond application (e.g. a system timer)
  - Internal software within application (e.g. notification of task completion)
- Application responds to event

User-generated Events

- System initiates an event based on user input onto connected hardware
  - Keyboard press
  - Mouse movement / click
  - Joystick control
- Operating system stores user input as event in a queue
- UI toolkits provide checks and responses to events
- Programmer determines behavior based on events

GUI and Menus

- Graphical user interfaces (GUIs) determine input based on mouse (or stylus) position on the screen
- Standard events already built into system
  - Window minimize, window close, etc
- Custom events added by programmer
  - Game paused, change music volume, etc
- User interacts with elements at any point of the program execution

How to Use with Callbacks?

- Tells the system what to do when particular event arrives
- Necessary code executes automatically
- Standard technique for a GUI system:
  1. Application implements function to handle event
  2. Application notifies GUI which function to call
  3. GUI handles this functionality when user interacts with the system

Graphical User Interface

- Computer interface with a visual component
- Direct interaction with the screen rather than interactions via command line
- Designed for easier, more intuitive experience
- Based on event-driven programming

Uses

- Text editors
- Web browsers
- Music controls
- Video games
- Many, many more…

(iMovie)

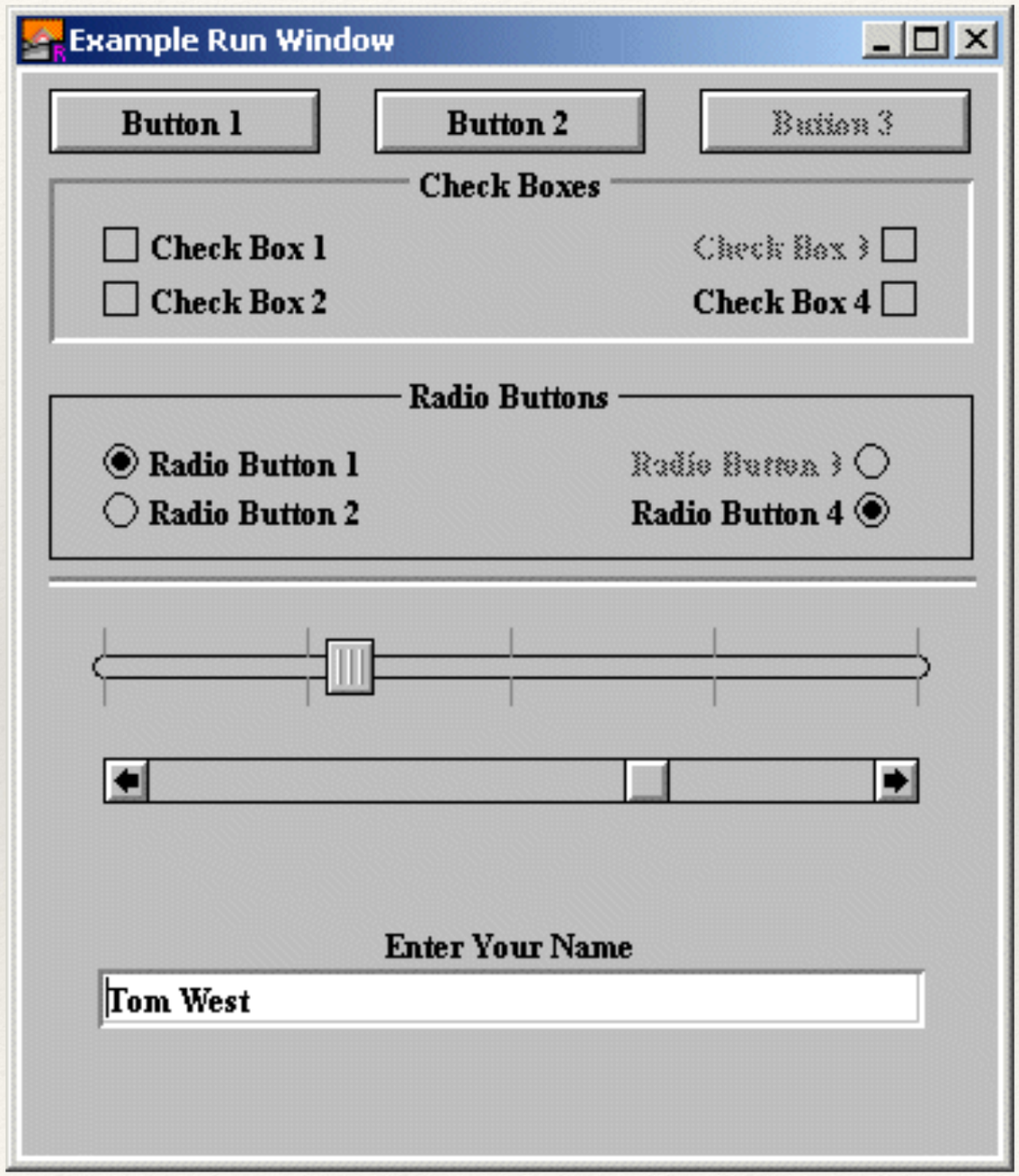
Consider

- How are some ways we can interact with a GUI?

Widgets

- Common interactable objects within a GUI:
  - Buttons
  - Check boxes
  - Radio buttons
  - Sliders
- Provide different ways of interacting with program behavior

Example Widget



(http://compsci.ca)

Buttons

- Allow for functionality upon mouse click
- Must be aware of mouse position and button boundary
- Circles and rectangles have accessible formulae to determine boundaries
  - Circles check based on radius from center position
  - Rectangles check based on width/height distance from corner (or center) position
- What do these equations look like?

Hands-on: Buttons

- Today's activities:

  1. Implement a `Button` class that checks when the mouse is over it, and when the mouse clicks on it
  2. Create both rectangular and circular buttons
  3. Experiment with the `mousePressed` and `mouseReleased` event calls
  4. Add functionality so that the sketch's background color changes every time a button is pressed