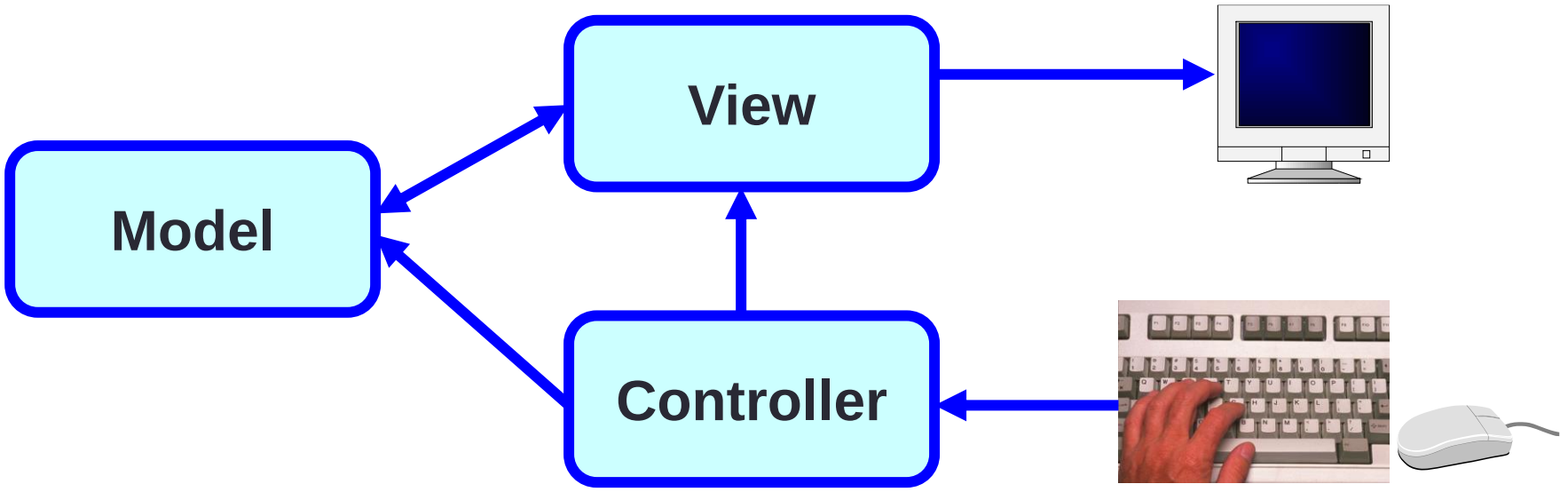
CMSC 132: OBJECT-ORIENTED PROGRAMMING II

Graphical User Interfaces (GUIs)

Department of Computer Science
University of Maryland, College Park

Model-View-Controller (MVC)

- Model for GUI programming (Xerox PARC ’78)
- Separates GUI into 3 components
  - Model ⇒ application data
  - View ⇒ visual interface
  - Controller ⇒ user interaction



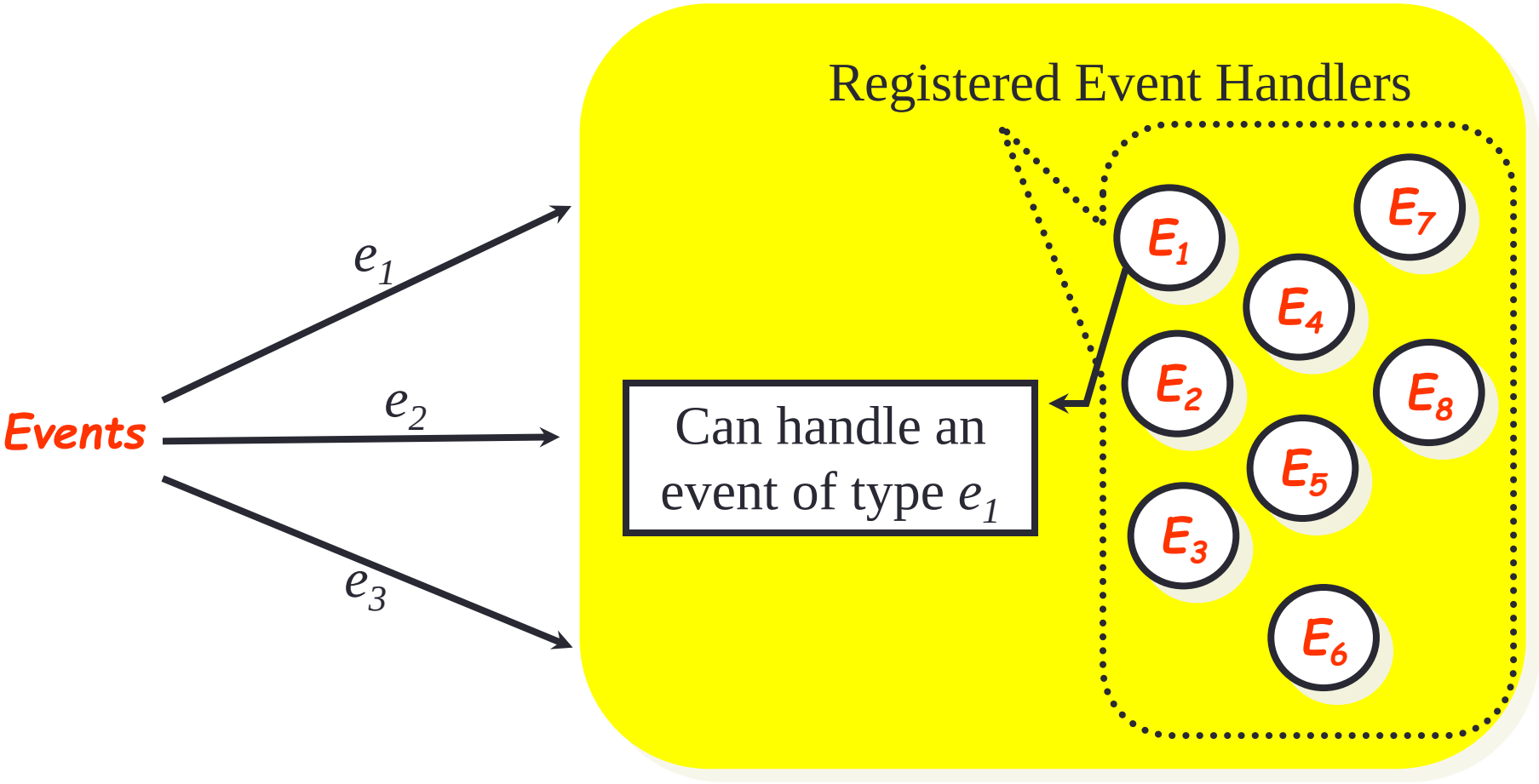
MVC Model of GUI Design

- **Model**
  - Should perform actual work
  - Should be independent of the GUI
    - But can provide access methods
- **Controller**
  - Lets user control what work the program is doing
  - Design of controller depends on model
- **View**
  - Lets user see what the program is doing
  - Should not display what controller thinks is happening (base display on model, not controller)

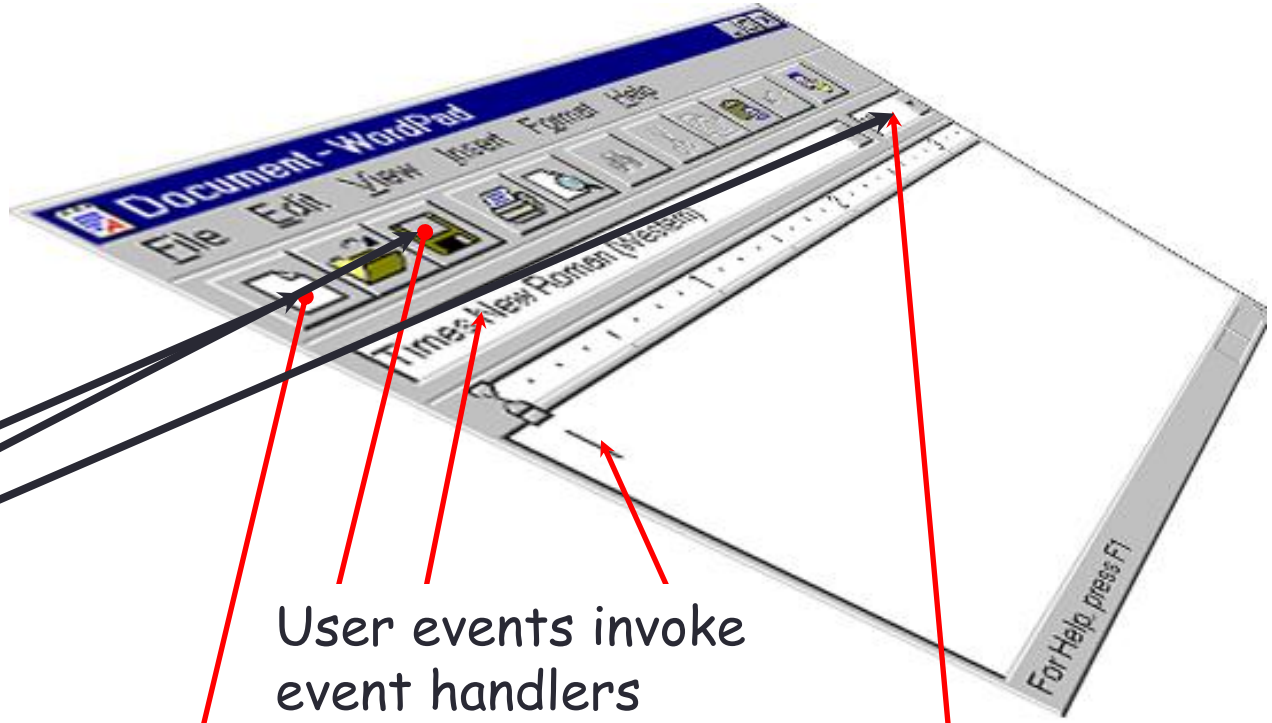
Programming Models

- **Normal (control flow-based) Programming**
  - Approach
    - Start at main()
    - Continue until end of program or exit()
- **Event-driven Programming**
  - Event → Action or condition occurring outside normal flow of control of program (e.g., mouse clicks, keyboard input, etc.)
  - Unable to predict time & occurrence of event
  - Approach
    - Start with main()
    - Define system elements and register event listeners
    - Await events (& perform associated computation)

Event Handling in Action



GUIs are Event-Driven Software



GUI

User Events

User events invoke event handlers

E_3 E_4 E_5 ..... E_2 E_1

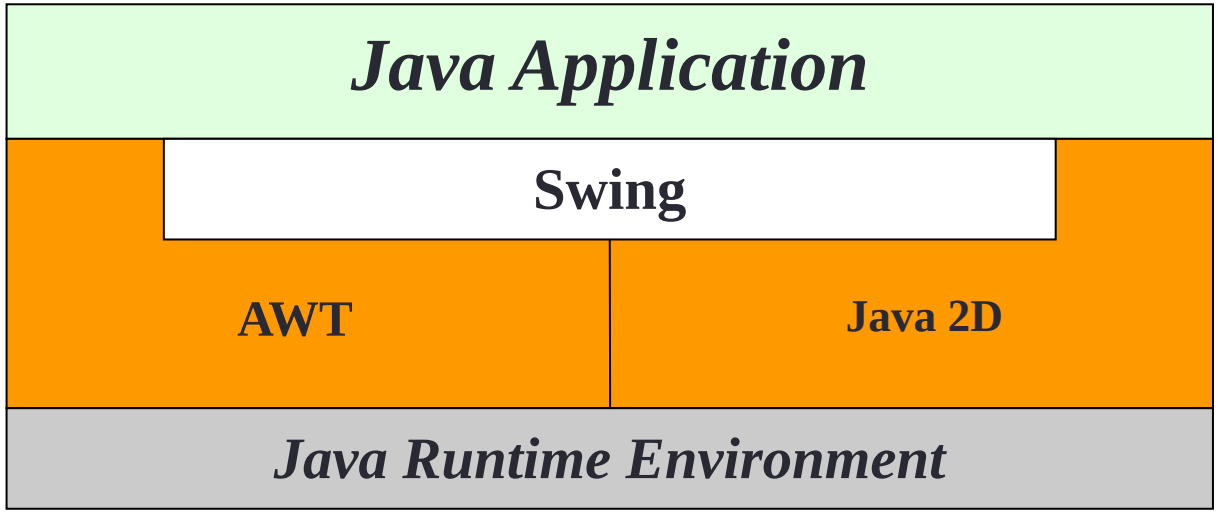
Event Handlers

newDocActionPerformed (java.awt.event.ActionEvent evt)

fileSaveActionPerformed (java.awt.event.ActionEvent evt)

changeFontSizeActionPerformed (java.awt.event.ActionEvent evt)

GUIs in Java



Desktop Java Graphics APIs: From “Filthy Rich Clients”
by Chet Haase and Romain Guy, Chap1, Page 12
ISBN-978-0-13-241393-0
Book Web Site: http://www.filthyrichclients.org/

Java FX

- New framework for GUI development in Java
- Swing and AWT are replaced by JavaFX
- JavaFX application can run on a desktop or a web browser
- JavaFX
  - Built-in 2D, 3D
  - Support for touch-enabled devices
  - Animation support
- JavaFX is not part of the Java SDK
- To install JavaFX see: http://www.cs.umd.edu/eclipse/javafx
- The examples we cover in this presentation can be found in the above link (**GUICodeFX.zip**)

JavaFX Program Structure

- **JavaFX Program**
  - Class extending javafx.application.Application
  - **Example:** Basics.java
- launch method - launches stand-alone JavaFX application
- main method is not needed if you run the program using the command line
  - JVM invokes the launch method in this case
- Stage - is a window
- When the program is run the JVM creates a stage (primary stage/primary window)
- Stage displays a scene
- A scene can contain nodes
- Nodes - Visual component or a pane
  - Pane - container classes that lays out nodes in a particular location
  - Visual components - Shapes, GUI control (e.g., button), image views
- FlowPane - nodes are placed row-by-row, column-by-column
- getChilden() - returns list of nodes in a pane
- **Example:** Pane.java

JavaFX (Panes)

- Panes are for storing and organizing nodes
- **FlowPane** - nodes placed row-by-row or column-by-column
- **BorderPane** - defines placement locations as
  - top, right, bottom, left, and center
  - setTop, setRight, setBottom, setLeft, setCenter
- **StackPane** - Nodes placed on top of each other
- **GridPane** - Places nodes in a two-dimensional grid
- **HBox** - Nodes are placed in a row
- **VBox** - Nodes are placed in a column
- **Example:** FlowPageEx.java, GridPaneEx.java, BorderPaneEx.java, HBoxEx.java

JavaFX (Handling Events)

- Objects that handles an action event object must be instance of EventHandler<T extends Event>
- EventHandler object must be registered with event source object
- **Example:** HandlingEvent.java, HandlingEventAnon.java, HandlingEventLambda.java

Examples

- **Example:** ScrollableArea.java**,** ImageExample.java, SlideShow.java, FadeAnimation.java, SliderExample.java

Additional Resources / References

- Introduction to Java Programming, Comprehensive Edition, 10th Edition, by Y. Daniel Liang
- https://openjfx.io/