

Lecture 25: Event-Driven Programming

CS 2110

November 19, 2026

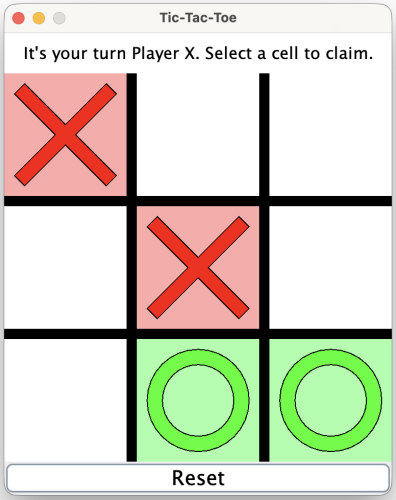
Today's Learning Outcomes

86. Categorize aspects of a GUI application as part of its *model*, *view*, or *controller*.

91. Describe the *observer pattern* and use it to write event listeners for graphical applications.

92. Explain the execution of a Swing application and its use of the event dispatch thread.

Where We Left Off...



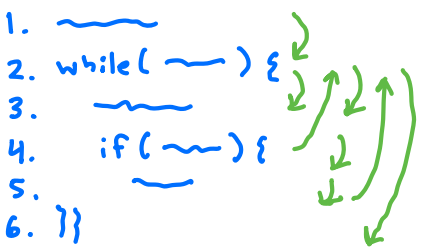
GUI Applications can be organized into 3 aspects:

- **Model:** "back end" state representation
- **View:** "front end" windows and widgets
- **Controller:** application logic (today's topic)

Programming Paradigms

Imperative

- write out sequence of instructions our code should follow
- "lower level"
- most code you've written



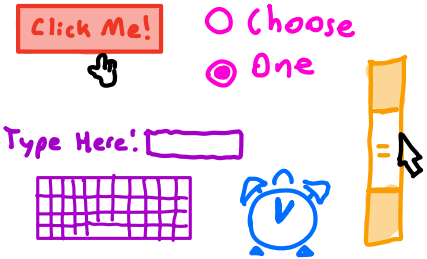
Declarative

- describe what you want the code to do, let it figure out how to make it happen
- "higher level"
- frameworks enable this



Event-Driven (subset of declarative)

- describe what a program should do in response to different "events"
  - mouse clicks
  - keyboard input
  - timers
  - state changes



The Observer Pattern

Interactive programs "invert control"

- We write code that describes what should happen *in response* to an event
- The running application determines when our code is actually executed

This inversion of control realizes the **observer pattern.**

The Swing Event Loop

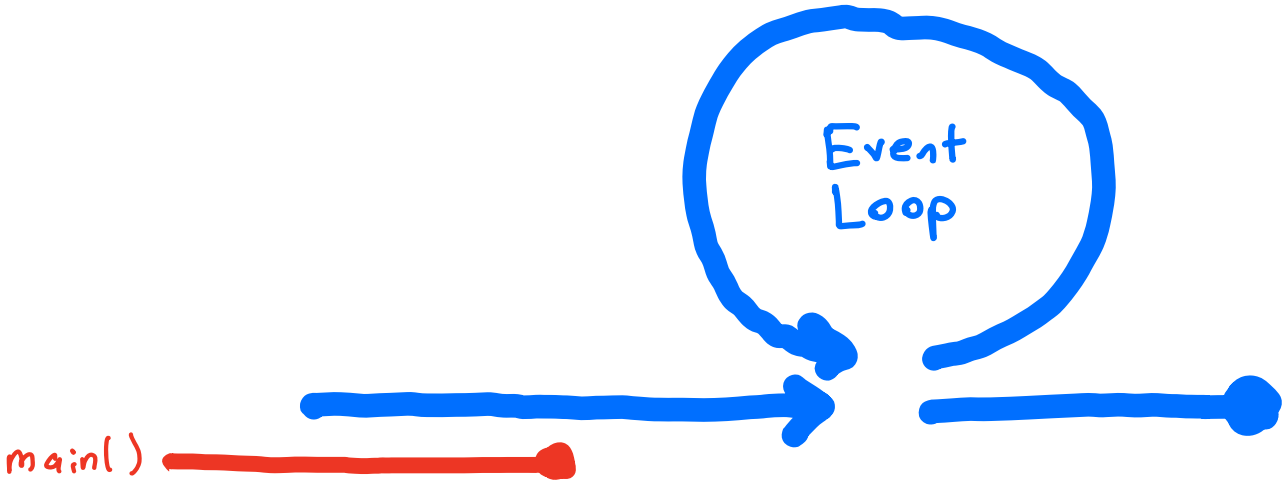
The subject in Swing's realization of the observer pattern is its **event loop.**

Rough diagram of a running Swing application:

Events and Listeners

An **event** is an object that models a scenario that can trigger a response in our program.

Visualizing Events





Coding Demo: The Reset Button



Simplified Listener Creation



Coding Demo: The Cell Buttons



Property Changes

Sometimes, we need a state update in one class to trigger an update in another class.

Directly linking (or *coupling*) all the classes to enable this (through an instance method call) becomes messy.

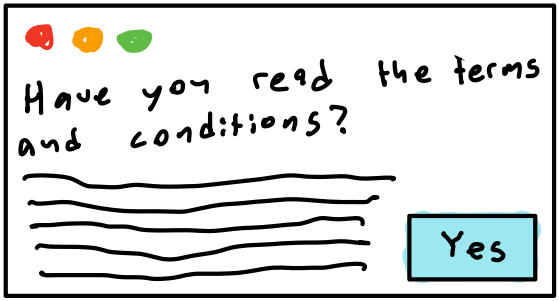


Coding Demo: The Turn Label



Dialog Boxes

= Temporary, secondary windows that prompt users at particular points in the application.



Handling Many Events

So far, our illustrations of the event loop have been simplified. In reality, hundreds of events may be queued in the event loop per second.

