CISC 3120
C11: GUI and JavaFX

Hui Chen

Department of Computer & Information Science

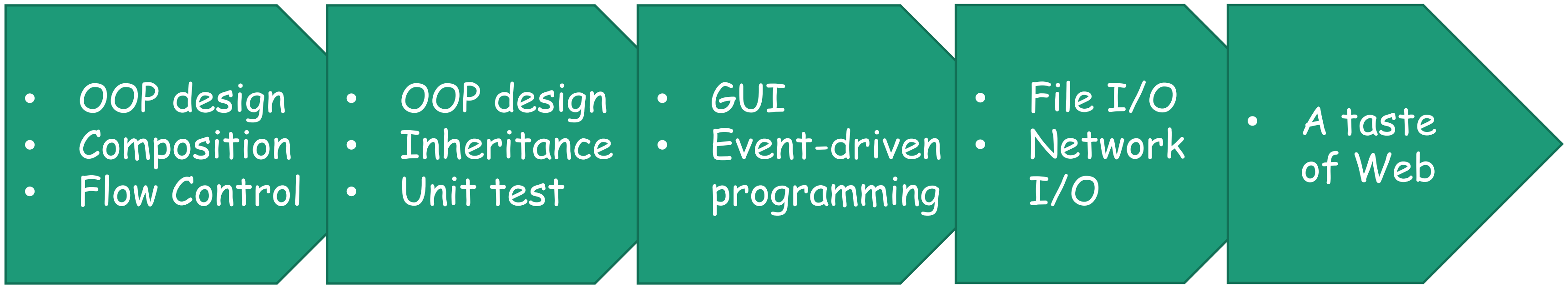
CUNY Brooklyn College

Outline

- Recap
    - OOP and Project 1?
- Project 2 activity
- Environment preparation
- Graphical user interface
- Introduction to JavaFX
- Assignments

Project 2: Design & Implement Applications

- 5 team projects (phases) to build a Treasure Hunt game application
  - Each built from start-up code or built from previous project (phase)
  - Text-based to GUI
  - Standalone to networked/distributed



- OOP design
- Composition
- Flow Control

- OOP design
- Inheritance
- Unit test

- GUI
- Event-driven programming

- File I/O
- Network I/O

- A taste of Web

Project 2

- Start to build the Treasure Hunt game application
  - A simple desktop application with text-based user interface (a.k.a., the command line user interface)
  - Game rules
    - A "treasure" is buried in a field, unearthing the treasure earns the player a score
    - Clues are given when the player solves a puzzler

Project 2: Objectives

- Apply the “inheritance” pattern in addition to the “composition” pattern
- Be proficient in flow controls

Design Applications: Classes & Objects

- Liskov substitution principle
  - Objects should be replaceable with instances of their subtypes without altering the correctness of that program

Treasure Hunt

- The game
  - Multiple types of puzzlers
  - Multiple types of treasures
- Design
  - Do we have to revise the code when we change puzzler types or treasure types in the code?
  - If not, what is the design?

Project 2 In-Class Team Discussion

- Select project coordinator: coordinator's responsibility
  - Accept the assignment
  - Make sure the start-up code is imported (if not automatically done by Github, do it manually)
  - Coordinate, monitor, and facilitate team collaboration and project progress
- Members' responsibility
  - Clone the repository and make a contribution as a team member
- Discuss initial tasks and steps:
  - Use Github issue tracking to create tasks (issues) and assign them team members
- Prepare questions for the class discussion

Questions?

- Project 2

Eclipse and E(fx)clipse

- Install e(fx)clipse plugin in Eclipse IDE

User Interface

- A system that allows two or more entities to exchange data
  - Typical entities are computers and humans
  - It includes both hardware and software

Types of User Interfaces

- Text-based user interface (or command-line interface)
- Graphical user interface

Text-based User Interface: Advantage

- Relies primarily on the keyboard and the terminal
  - Easy to customize options
  - Can do powerful tasks
  - Relatively easy to build
  - Require few resources (processor and memory) to support the interface

Text-based User Interface: Disadvantage

- Relies heavily on user’s recall rather than recognition
- Navigation is often more difficult

Text-based User Interface: Your 1st Project

- We run the game from the command line to control window size and game level:

```
$ java TreasureHuntConsoleApp --window-width 80 --
window-height 25 --level 2
```

Text-based User Interface: "javac" Example

- We use "javac" to compile Java programs
- Type "javac" on the command line

```
$ javac
Usage: javac <options> <source files>
where possible options include:
  -g                          Generate all debugging info
  -g:none                     Generate no debugging info
  -g:{lines,vars,source}      Generate only some debugging info
  -nowarn                     Generate no warnings
  -verbose                    Output messages about what the compiler is doing
  -deprecation                Output source locations where deprecated APIs are used
  …..
```

Interfacing with “javac“

- Display version of “javac“

```
$javac -version

javac 1.8.0_131
```

- Compile a Java program targeting at Java version 8 or newer

```
$javac -target 8 HelloWorld.java
```

Text-based User Interface: “ls” Example

- We can use “ls” to list files on a Unix/Unix-like operating systems (Linux, Mac OS X, etc.)
  - ls –l: list files and directories in long format
  - ls –F: append character to indicate file types
  - ls –l –F: list files and directories in long format and append character to indicate file types
  - Common combinations of options is 100+

Interfacing with “ls”

- Common combinations of options is 100+
- Either frequently look up them from the user’s manual or memorize them (recall other than recognition)
- Perhaps, we can create a program that has a menu or a list buttons
    - You need 100+ menu entries or buttons

Graphical User Interface

- Often use acronym: GUI
- Visualizes data for users graphically
- Often equipped with mouse, trackball, or touch pad

Graphical User Interface: Advantage

- Provides a friendly interface between user and program
  - Relies more on recognition than recall (less knowledge to use the application)
- Is Often equipped with point-and-click devices (mouse, trackball, joystick, touchpad …)
  - Allows user navigate easily

Graphical User Interface: Disadvantage

- Typically decreased options (less powerful)
- Typically less customizable..
    - Recall the “ls” example
    - Not easy to express many combinations of options in GUI
    - Not easy to use one set of button for many different options or combinations in GUI

Graphical User Interface

- More user friendly and easy navigation
- GUI applications are popular in modern computing
- Allows event-driven or reactive programming
- Often multi-threaded: allows multiple concurrent threads of executions

Questions

- Text-based user interface
- Graphical user interface
- Disadvantage and advantage

Event-Driven Programming

- The main body of the program is an event loop (in pseudo code)

```
do {
    e = getNextEvent()
    processEvent(e)
} while (e != EXIT_EVENT)
```

- This event loop often implemented by the platform
- Users write event handler routines to process events

Event-Driven, Application-Driven, and Algorithm-Driven

- Application-driven or algorithm-driven programs
    - A program expects inputs in a pre-determined order and timing
    - e.g., Project 1 and Project 2
- Event-driven programming
    - Program waits for input events when it loads
    - The programs runs particular code to response with an event
    - The overall flow of the execution is determined by the events that occur
    - The overall flow of what code is executed is determined by events in non-deterministic order and timing
- A type of reactive programming

GUI Event-Driven Programming

- GUI programming are typically event-driven
- Event
  - An object that represents a user's interaction with a GUI components (e.g., a button, a menu item)
- Event Listener
  - An object that waits for events and responds to them.
- Event Handler
  - An object that calls by the Event Listener to handle an event as a part of the response

GUI Event Handling

- Programmer attaches a listener to a component for an event (e.g., a button, a menu)
- Platform notifies the listener when the event occur (e.g., a button click)
- The listener calls the Event Handler's methods as a part of reponse

Questions

- Concept of event-driven programming
- Concepts of GUI event-driven programmingf

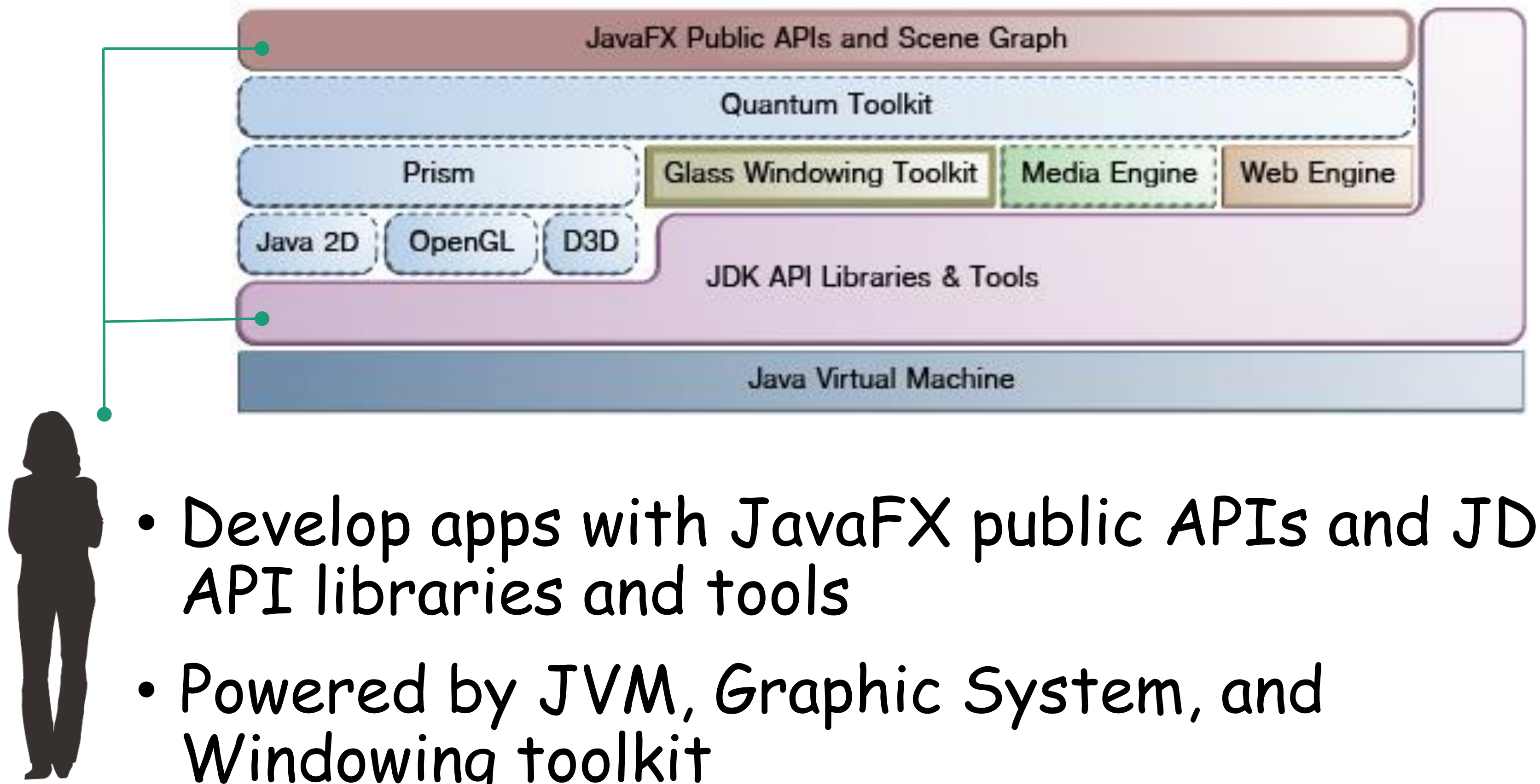
GUI Application in Java

- JavaFX is a Java API
  - "JavaFX is a set of graphics and media packages that enables developers to design, create, test, debug, and deploy rich client applications that operate consistently across diverse platforms."
  - Shipped with JRE since JRE 8
- Designed to replace Java AWT and Swing

JavaFX Overview

- A Java API
  - Consisting of classes & interfaces in a few Java packages
  - Dealing with graphics and media
  - for creating rich client applications
    - whose look & feel are customizable via Cascading Style Sheets (CSS)
  - cross platforms
    - Desktop, mobile, embedded, and the Web

JavaFX Architecture



- Develop apps with JavaFX public APIs and JDK API libraries and tools
- Powered by JVM, Graphic System, and Windowing toolkit

Features of JavaFX

- Graphics: supports *DirectX* and *OpenGL*, software render fallback (via Prism, OpenGL, Direct3D)
    - 3D graphics: supports light sources, material, camera, 3-D shapes and transformations; Common visual effects
- Interfacing with native operating systems to provide windows management, timers, and event queues (Glass windows toolkit)
- Multimedia: support playbacks of web multimedia content based on the GStreamer multimedia framework (Media engine)
- Web: provides a Web viewer and full browsing functionality based on WebKit (Web engine)
- Multi-threaded: concurrent application, Prism render, and media threads (Quantum toolkit)
- Text: supports bi-directional text and complex text scripts
- I/O devices: supports multi-touch and Hi-DPI
- Build-in UI controls, layouts, themes, and CSS styling
- Swing interoperability
- JavaFX API: application lifecycle; stage; scene; transition & animation; canvas; print; event; css; fxml; collections; utils; Java beans; javascript

JavaFX API

- Full package list at http://docs.oracle.com/javase/8/javafx/api/toc.htm
  - javafx.application: provides the application life-cycle classes.
  - javafx.stage: provides the top-level container classes for JavaFX content.
  - javafx.scene: provides the core set of base classes for the JavaFX Scene Graph API.
  - javafx.scene.control: prebuilt UI control classes
  - javafx.scene.text: provides the set of classes for fonts and renderable text.
  - javafx.scene.layout: prebuilt container classes defining user interface layout.
  - ……

The First JavaFX Application

- The HelloWorldFx Application
  - The application is part of JavaFX tutorial from Oracle and in the Sample Programs repository

"Everything should be built top-down, except the first time."

-- Alan Perlis

The HelloWorldFx Application

- Application
- Stage & Scene
- Event listener, registration, and handler
- UI controls

JavaFX Applications

- Must have a main class that extends the JavaFX Application class
  - javafx.application.Application
- The entry point is actually the "start" method
  - In IDE, you need the *main(String[] args)* method
  - If packaged as a Jar file with JavaFX packager tool, the main method is not necessary
    - To be discussed later in this lesson

Stage and Scene

"All the world's a stage, and all the men and women merely players."

-- As You Like It, Act II, Scene VII, William Shakespeare

JavaFX Stage

- A JavaFX runtime constructs a primary stage
  - java.stage.Stage: the top level JavaFX container
  - Visually represented by a "window" in windows-based operating systems (such as, Windows, Mac OS X)
  - Can receive and handle events
  - An applications can construct additional stage
  - The application needs to construct and set scenes for a stage
    - JavaFX scene graph
    - To be discussed next class

JavaFX Scene

- javafx.scene.Scene: The JavaFX Scene class is the container for all content.
- The content of the scene is represented as a hierarchical scene graph of nodes.
  - Need a root node to build a scene
- To be discussed next class

JavaFX Events

- Examples:
  - ActionEvent (mouse clicks, ENTER key presses)
  - InputEvent (drag, gesture, key, mouse, and touch by InputEvent's subclasses)
    - Example of subclass: MouseEvent
  - WindowEvent (window showing, hiding)

JavaFX Event Registration

- JavaFX UI components provide convenient means to register event handlers
  - Button (javafx.scene.control.Button)
    - setOnAction(EventHandler<ActionEvent> event)
  - Rectangle (javafx.scene.shape.Rectangle) and Circle (javafx.scene.shape.Circle)
    - setMouseEntered(EventHandler<MouseEvent> event)
    - setMouseExited(EventHandler<MouseEvent> event)
    - setMouseMoved(EventHandler<MouseEvent> event)

Examples: Stage, Scene, Event

- A few examples for Stage, Scene, and Events
    - In the Sample Programs repository

Questions

- Concept of user interfaces
- Comparison of command-line and graphical user interfaces
- First GUI program in Java
- JavaFX, JavaFX Stage and Scene, JavaFX Controls (Buttons and Shapes) and Events

Running the HelloWorldFx Program

- From the Eclipse IDE, as usual
- From the command line, two steps,
  - Export a runnable Jar file in Eclipse IDE
    - Project context menu → Export → Runnable Jar file.
    - Pending a bug fix in Eclipse, may not work for Java 9 Maven Project
  - Run the main class in the Jar file from the Command Line
    - One line (no line break)

```
java -classpath HelloWorldFx.jar edu.cuny.brooklyn.gui.HelloWorldFx
```

Assignments

- Project 2
- CodeLab
- Practice
- Review Guide #2 and Take-Home Test 2