CISC 3115 TY3

C29c: Overview of JavaFX Application: Views

Hui Chen

Department of Computer & Information Science

CUNY Brooklyn College

Outline

- Overview of a JavaFX application
  - JavaFX Stage and Scene
    - The HelloWorldFX program
    - The StagesFXApp program
    - The ScenesFXApp program
  - First exposure to the JavaFX Scene graph

Examining The HelloWorldFx Application

- Application
- Stage & Scene
- Scene graph

JavaFX Applications

- Must have a main class that extends the JavaFX Application class
  - javafx.application.Application
- The entry point is actually the "start" method since the main method is always implemented the same

Stage and Scene

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

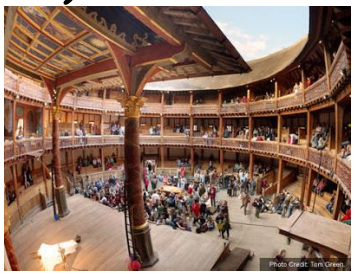
-- As You Like It, Act II, Scene VII, William Shakespeare

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)
  - An applications can construct additional stage
  - The application needs to construct and set scenes for a stage
    - JavaFX scene graph
  - Can receive and handle events

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.
  - A scene graph is actually a tree
  - We need a root node to build a scene, usually a Pane (more often one of its subclasses)



Questions

- Overview of a JavaFX application
- JavaFX Stage, Scene, Scene graph

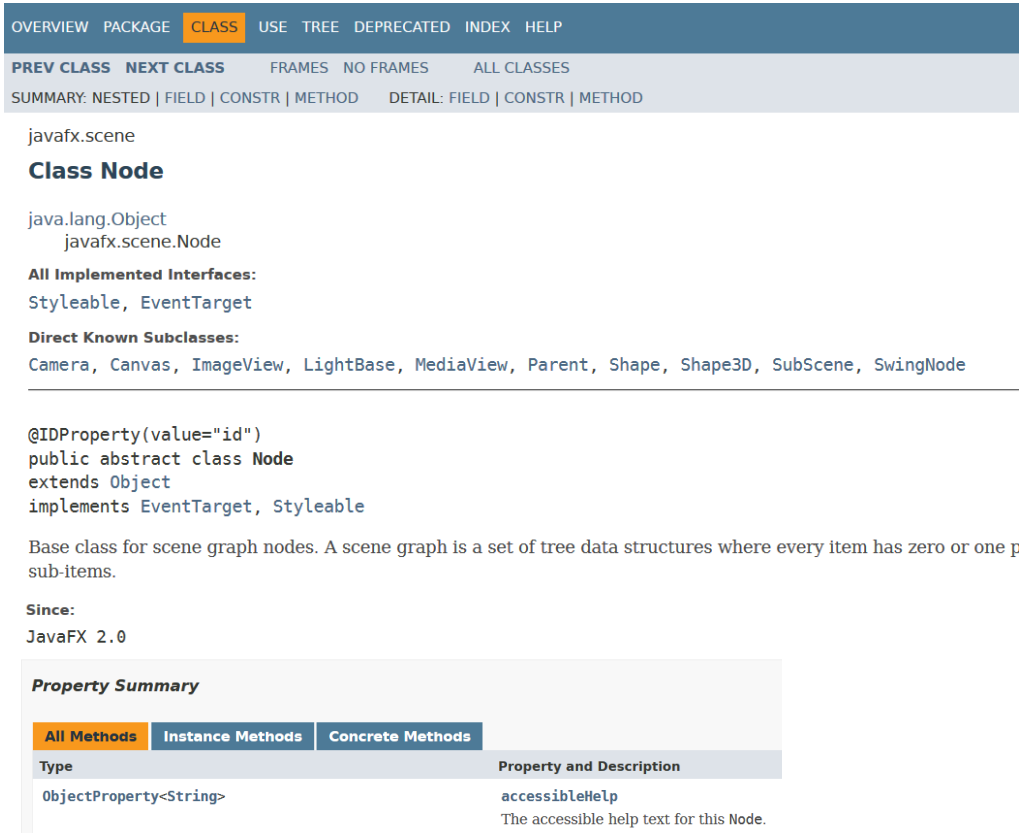
More on JavaFX GUI Application

- Learn to write JavaFX application
  - Learn to use Java API documentation
  - Learn a few concepts in GUI and computer graphics
- JavaFX application life cycle
- JavaFX application structure
- JavaFX event processing
- JavaFX build-in UI components

Java API Documentation

- Class documentation
  - Package hierarchy
  - Class name
  - Implemented interfaces
  - Known subclasses
  - Class declaration line
    - Abstract or concrete
    - Super class
  - Description
  - Compatibility
- Properties
  - Public instance variables
- Fields
  - Public class variables and constants
- Constructors
- Methods
  - Method summary
  - Methods inherited
- Property detail

Java API Documentation



OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

javafx.scene

Class Node

java.lang.Object
    javafx.scene.Node

All Implemented Interfaces:

Styleable, EventTarget

Direct Known Subclasses:

Camera, Canvas, ImageView, LightBase, MediaView, Parent, Shape, Shape3D, SubScene, SwingNode

---

```
@IDProperty(value="id")
public abstract class Node
extends Object
implements EventTarget, Styleable
```

Base class for scene graph nodes. A scene graph is a set of tree data structures where every item has zero or one p
sub-items.

Since:

JavaFX 2.0

Property Summary

All Methods | Instance Methods | Concrete Methods

Type	Property and Description
ObjectProperty<String>	accessibleHelp The accessible help text for this Node.

JavaFX Application

- JavaFX platform is the environment where JavaFX applications run
  - javafx.application.Platform: Application platform support class
    - Control & query platforms: e.g., accessibility, implicit exit
- Entry point: the Application class
  - javafx.application.Application
    - abstract void start(Stage primaryStage)

JavaFX Application Life-Cycle

- JavaFX runtime does the following, in order,
  - Constructs an object of the specified Application class (via the launch(String[] args) method), with regard to the Application object:
  - Calls the *init()* method that can be overridden
  - Calls the *start(javafx.stage.Stage)* method that must be overridden in subclass)
  - Waits for the application to finish, which happens when either of the following occur:
    - the application calls *Platform.exit()*
    - the last window has been closed and the implicitExit attribute on Platform is true
  - Calls the stop() method (can be overridden)

JavaFX Application: Remarks

- The *start(javafx.stage.Stage)* is an abstract method, and must be overridden in the subclass
- The init() and stop() method have concrete implementations, but do nothing, and can be overridden.
- Explicitly terminating JavaFX application
  - calling Platform.exit() is the preferred method
  - Calling System.exit(int) is acceptable, but the *stop()* method will ***not*** run, so better to avoid it.
- JavaFX should not and cannot be used after System.ext(int) is called or the stop() is returned.

Questions?

- JavaFX Platform and Application
- Main agenda when developing JavaFX applications

More on Stage and Scene

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

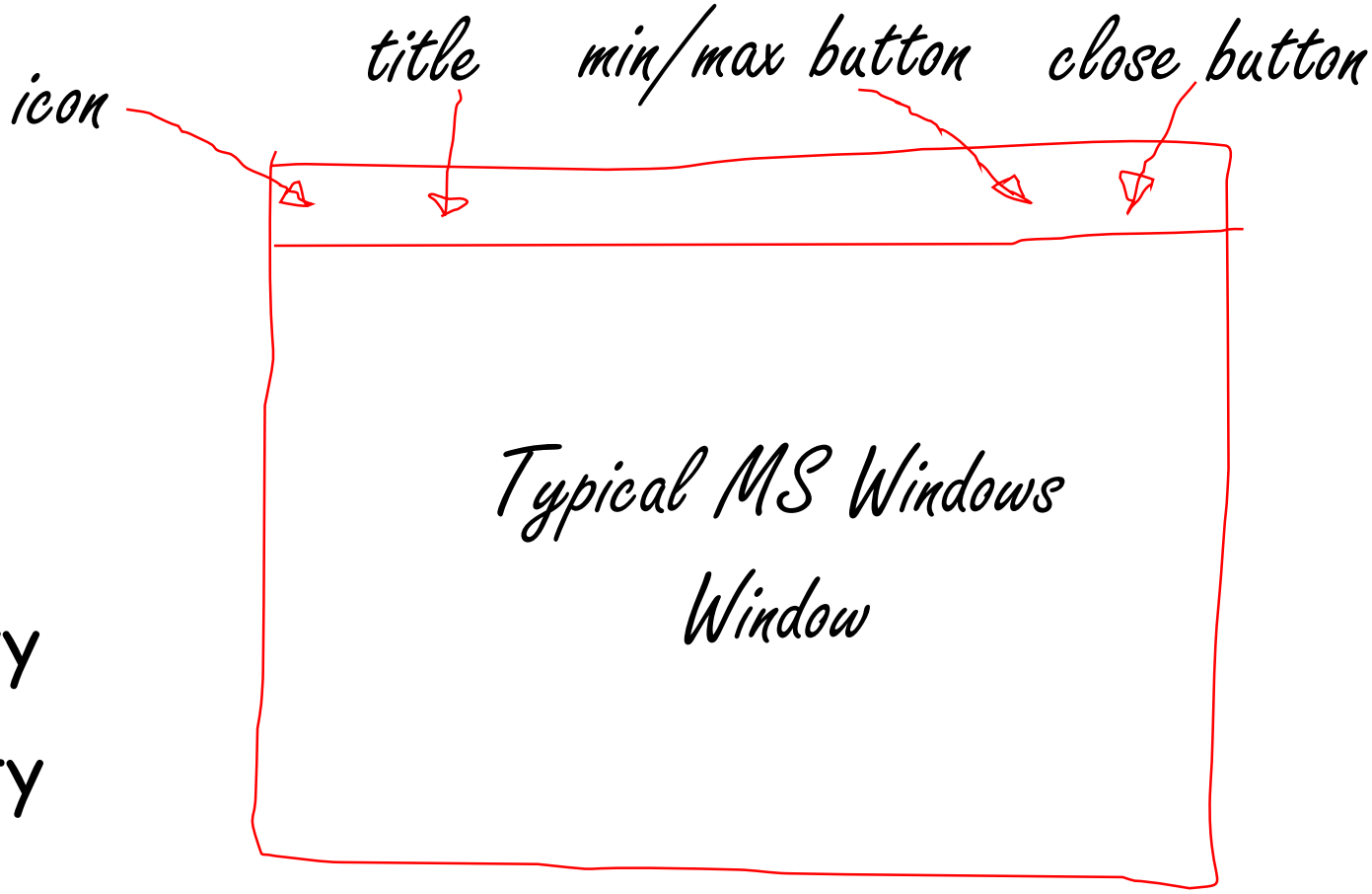
-- As You Like It, Act II, Scene VII, William Shakespeare

JavaFX Stage

- Top level JavaFX container
  - Can have a Scene
  - Associated with a Window
- Primary Stage
  - First Stage constructed by the Platform
- Additional Stage
  - Constructed by the application

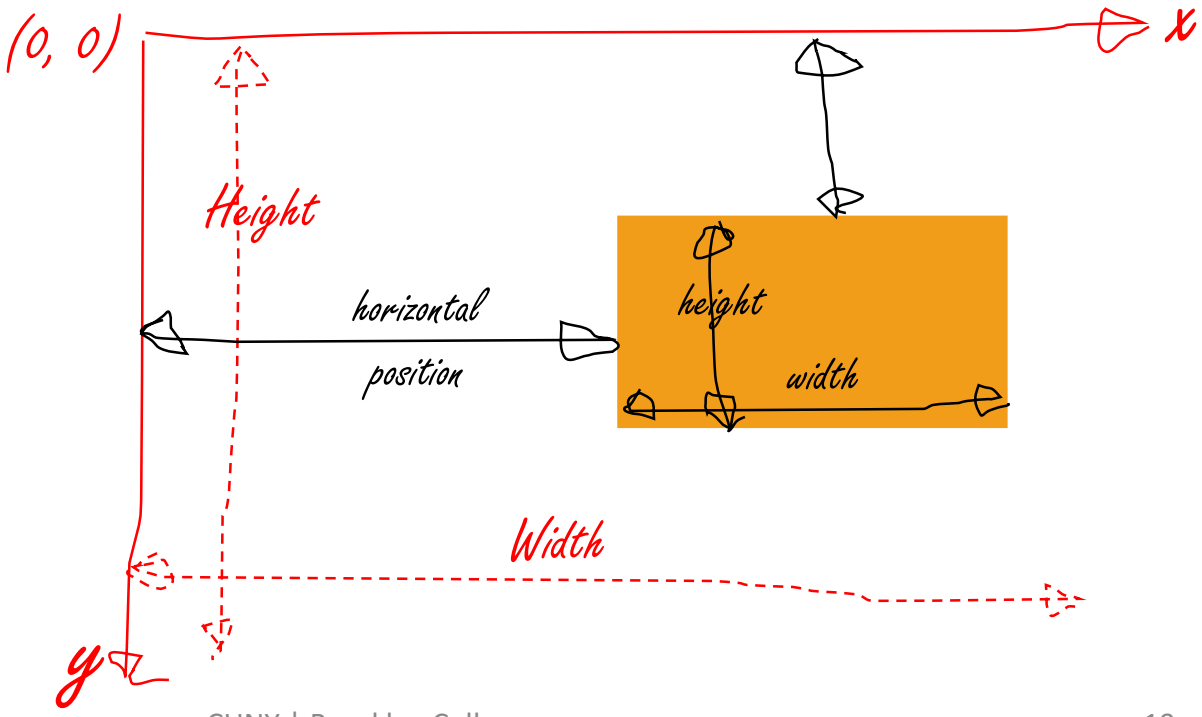
Stage (or Window)

- Size
- Shape
- Title
- Icon
- Modality
- Visibility



Scene Node Coordinate System

- A traditional computer graphics "local" coordinate system (javafx.scene.node)



Stage Style

- A stage can be one of a few styles
  - StageStyle.DECORATED
  - StageStyle.UNDECORATED
  - StageStyle.TRANSPARENT
    StageStyle.UTILITY

Stage Modality

- Modality.NONE
- Modality.WINDOW_MODAL
- Modality.APPLICATION_MODAL

Example Application

- The StatesFXApp application

Questions?

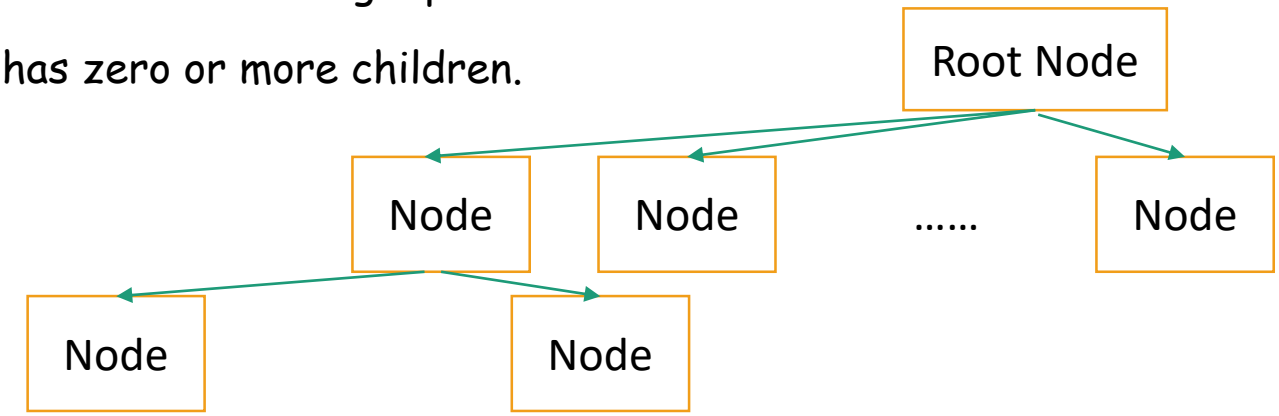
- Window and Stage
  - Style
  - Modality
  - Visibility
  - Size

JavaFX Scene

- Represent visual elements of user interface.
  - Elements can be displayed inside a window (on a Stage)
  - Scene graph
    - The elements form a graph called a scene graph
- Can be rendered
- Can handle events

Scene Graph

- Elements organized as a hierarchical structure, like a tree (a tree is a graph)
  - A graph is understood as a collection of nodes (or vertices), and edges (representing some connection or association)
  - An element in a scene graph is called a node.
    - Each non-root node has a single parent.
    - Each node has zero or more children.



Node in Scene Graph

- Example nodes
  - a layout container, a group, a shape, a button …
- Each node has an ID, style class, bounding volume, and other attributes
  - Effects, such as blurs and shadows
  - Opacity
  - Transforms
  - Event handlers (such as mouse, key and input method)
  - An application-specific state
- javafx.scene.Node: abstract class

Building Scene Graph

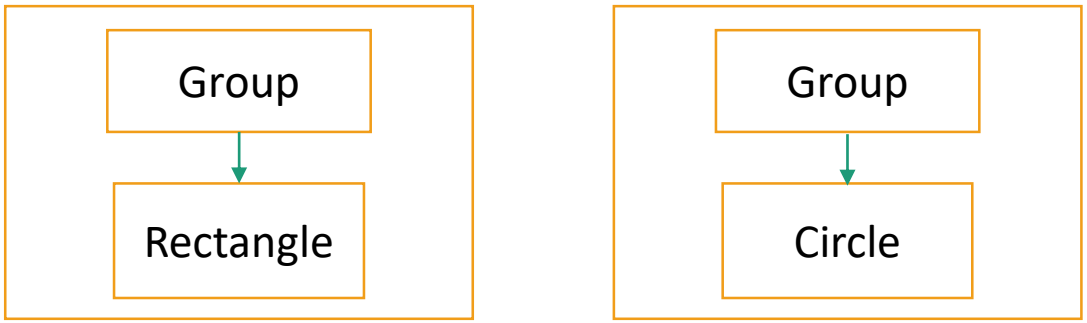
- Create a root Node
- Add children Nodes to root Node
- Register event handlers
- Set it on a Stage

Example Application

- The ScenesFXApp application

Discussion: Stage and Scene, and Scene Graph

- Can we have multiple scenes?
- Can we have multiple stages (windows)?
- Can we add more children to a scene graph?
  - A Parent node can have children.



Recap: Writing the (Simple) JavaFX Application

- Create a concrete subclass extending the JavaFX Application class (javafx.application.Application)
- (Curtains down) Construct a scene graph containing a tree of nodes
  - The simplest tree contains a single root node (select a concrete subclass of nodes)
    - http://docs.oracle.com/javase/8/javafx/api/javafx/scene/Node.html
  - Register some events to handle
- Set scene for the stage
- (Curtains up) Show the scene

Questions

- JavaFX Stage
- JavaFX Scene
- Simple JavaFX applications
- Example applications demonstrating Stage and Scene

Building Scene Graph

- Packaged in javafx.scene
- Nodes (elements)
  - Examples: text, charts, containers, shapes (2-D and 3-D), images, media, embedded web browser, and groups
- Transforms
  - e.g., positioning and orientation of nodes
- Effects
  - Visual effects (algorithm resulting in an image)
  - Objects that change the appearance of scene graph nodes, such as blurs, shadows, and color adjustment
- A scene graph must have a root node

Scene Graph Root Node

- Must a concrete subclass of javafx.scene.Parent
- Can be a Group or a Region object
  - Group
    - effects and transforms to be applied to a collection of child nodes.
  - Region
    - class for nodes that can be styled with CSS and layout children.
    - Layouts and Controls

Layouts and Controls

- Layouts
  - Classes support user interface layout
    - Examples: horizontal layout, vertical layout, grid layout, back-to-front
- Controls
  - A node in the scene graph that can be manipulated by the user
    - Labeled: buttons, labels, text fields, toggle button, checkbox, menu button, …
    - List view, combo box, menu bar, scroll bar, progress indicator, spinner, slider, …

Questions?

- Stage and Scene
- Scene graph
- GUI windows and Scene node