

# Timers, Transitions, and Custom Components

Josh Wilcox (jw14g24)

April 30, 2025

# Table of Contents

① Stages

② Timers and Scheduling

③ Reflection

# Stages

## New Stages and New Windows

- A Stage is the container for a native window
- A Stage is required with a single Scene in the `start()` method of a javafx project in order for it to run
- If we want to spawn a new window, we can create a new Stage and call `show()` in it

```
1 private void openScreen(String title, String message) {  
2     Stage newStage = new Stage();  
3     VBox layout = new VBox(10, new Text(message));  
4     layout.setPadding(new Insets(20));  
5     Scene scene = new Scene(layout, 300, 200);  
6     newStage.setScene(scene);  
7     newStage.setTitle(title);  
8     newStage.show();  
9 }
```

## Switching Scenes on One window

- Just use the `previousStage.setScene()` method

# Timers and Scheduling

## Timeline Class

- **Timeline** is an animation class that allows actions to be scheduled over time
  - Does this by managing a sequence of **KeyFrame** objects
  - Each **KeyFrame** specifies what should happen and when
- The **Timeline** just specifies the over all timing - starting, stopping, repeating, and the speed

## KeyFrame Class

- Each keyframe needs a **Duration** and an **EventHandler**
  - **Duration** - How long after the start of the timeline instantiation this keyframe should trigger
  - **EventHandler** - The action to perform after this time has elapsed - Same process as button event handlers

# Reflection

## What is Reflection?

- The ability of an object to examine itself
- Allows for dynamic information fetching about:
  - Data fields
  - Methods
  - Constructors
- Reflection is all based around the `Class` class

## The Class class

- The `Class` class contains information about a certain class such as:
  - The data fields
  - The constructors
  - The methods