Act20: Event-Based Game Loop and Custom Exceptions

Today, we look at how to create event-based game loops using threads and timers in Java. We will also explore custom exceptions, learning how to define our own classes in the `Exception` hierarchy and use them effectively in our code.

Content Learning Targets

After completing this activity, you should be able to say:

- I can use a `Timer` object to create an event-based game loop.
- I can create, throw, and catch custom exception classes that extend `Exception` (or an appropriate subclass).
- I can trace code that uses exception handling.

Process Skill Goals

During the activity, you should make progress toward:

- N/A

Facilitation Notes

First Hour: Event-based game loop:

- intro to threads and timers
- live coding: raindrops game
- concurrent modification exception

Second Hour: Custom exceptions:

- review of exception handling: FileAverage
- exception class hierarchy
- try-catch-finally
- creating and throwing custom exceptions
- live coding: FileBestScore example
- any remaining time: project work

Model 1 Live Coding: Raindrops Game

1. See the `Part1-EventBasedGameLoop.pptx` slides for more info about threads in Java.

2. Follow along as we live code the raindrops game in `src/gameEventLoop`.

3. Brainstorm with your final project team: Does your project need an event-based game loop? If so, what events should occur, and how often? How will you update the game state after each tick?

Model 2 Custom Exceptions

See the `Part2-Files.pptx` slides for more info about file I/O, including a useful built-in class called `JFileChooser`. See also the `Part3-ExceptionHandling.pptx` slides.

4. Together, let's review exception handling using the `FileAverage.java` example.

5. Live coding: `FileBestScore.java` example that uses a custom exception to handle invalid scores in a file.

6. Practice tracing code with exceptions in the Custom Exceptions Practice Moodle activity.

7. Brainstorm with your team:

a) What kind of exceptions should occur?

b) What should happen when they occur?

8. Use any remaining time to work on your final project.