



Java Project Demo

SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

11712019 余添诚

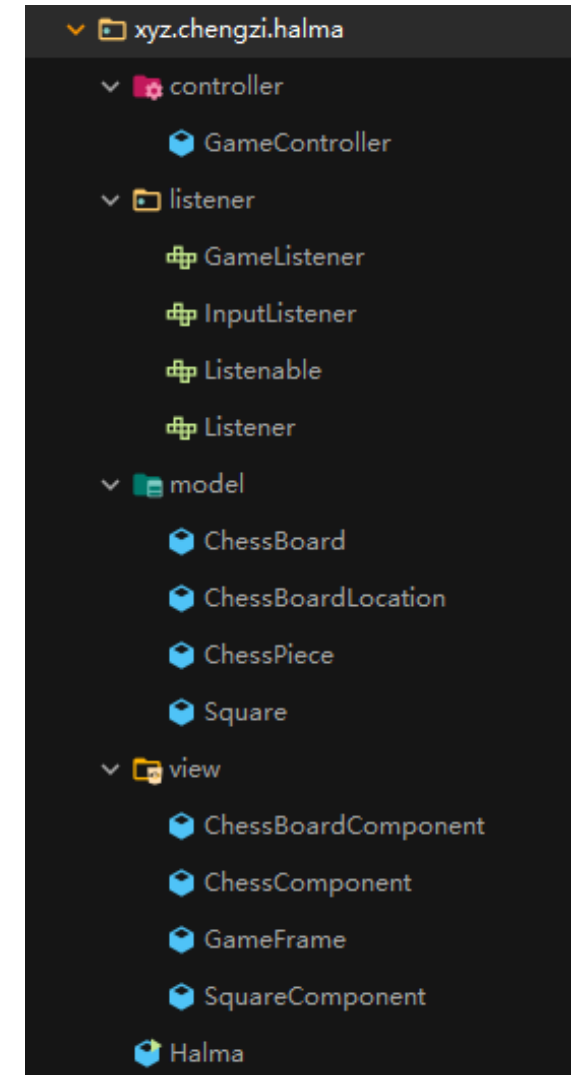
What Are Used in This Project

- The Java Development Kit (JDK) 8+
- The Java Swing GUI Library
- A computer with monitor, keyboard and mouse

Project Structure

- xyz.chengzi.halma: The parent package of the project
 - controller: Interact between views and models to handle user input and update data structures
 - listener: Trigger when event (e.g. player clicks) happens
 - model: Hold data structures and handle game logics
 - view: Render the GUI interfaces and display on screen

Note: Despite of the naming, the project does NOT strictly follow the MVC design pattern for simplicity.



Intelligent Computer Players

- Random moves?
 - Randomly choose a piece, and randomly move to one of the available square...
- Alpha-beta search
 - Reference: <https://github.com/aimacode/aima-java>
- Monte-Carlo tree search
 -

Save / Load Games

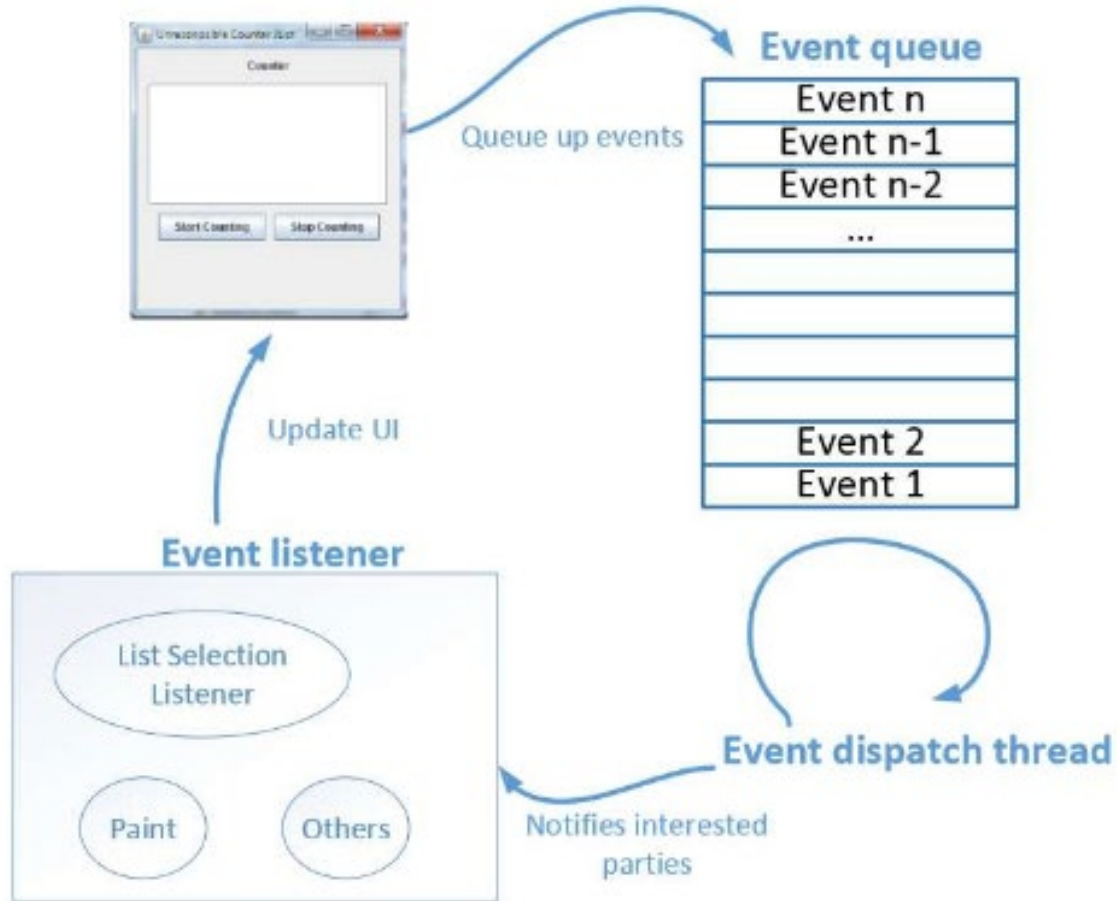
- Simplest Method
 - Implement the `java.io.Serializable` interface in the model classes
 - When saving, dump the object to file using `ObjectOutputStream`
 - Upon loading, load the object from file using `ObjectInputStream`
- More elegant ways
 - Store the data in yaml or json format.
 - Or use databases like SQLite or MySQL.

Online Multi-player

- WebSocket is an easy way to communicate between client and server.
 - It also works in browser!
- Recommended framework:
 - <https://javalin.io>

```
app.ws("/websocket/:path", ws -> {
    ws.onConnect(ctx -> System.out.println("Connected"));
    ws.onMessage(ctx -> {
        User user = ctx.message(User.class); // convert from json
        ctx.send(user); // convert to json and send back
    });
    ws.onBinaryMessage(ctx -> System.out.println("Message"))
    ws.onClose(ctx -> System.out.println("Closed"));
    ws.onError(ctx -> System.out.println("Errored"));
});
```

How the Swing works:



How The Swing Works?

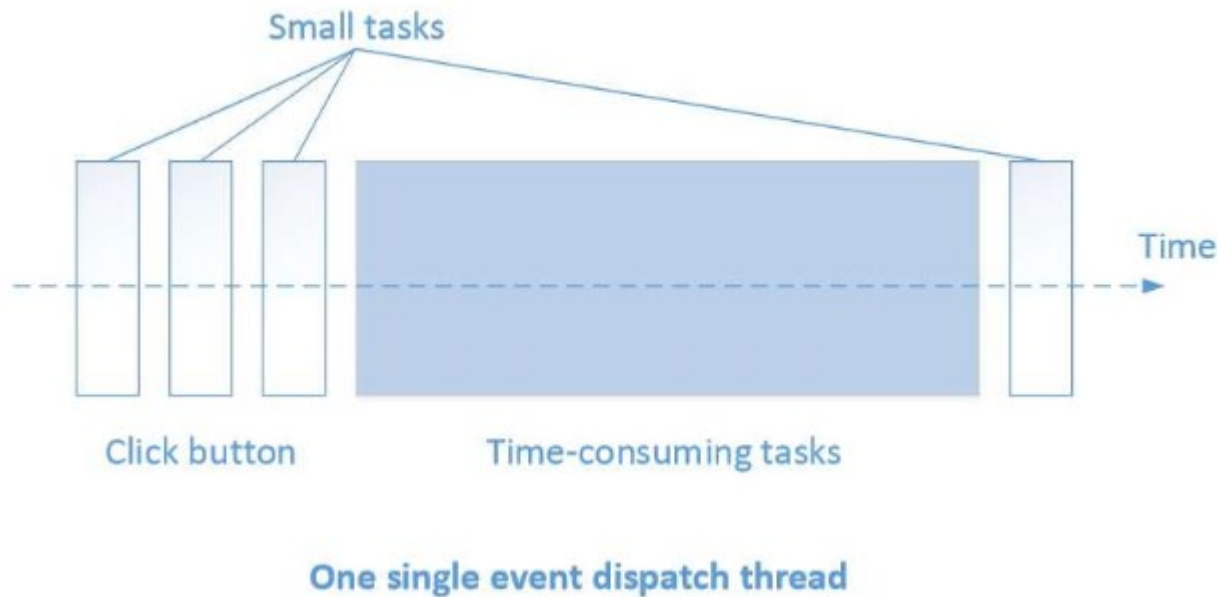
Multi-threading in Swing

- After program started
 - Main Thread: Generated by JVM executing main() method
 - AWT-Window: Listen to UI events from AWT windows
 - AWT-Shutdown: Handle termination of AWT windows
 - AWT-EventQueue-0: Event dispatching thread, manage event processing
 - DestroyJavaVM: Cleanup work for JVM after program exits
- Why we need multi-threading?
 - Avoid user interface not responding when program is busy.
- Reference:
 - https://www.cs.bham.ac.uk/~szh/teaching/ssc/lecturenotes/Concurrency/Lecture5_Multithreaded_Swing.pdf

Why Multi-threading in Swing?

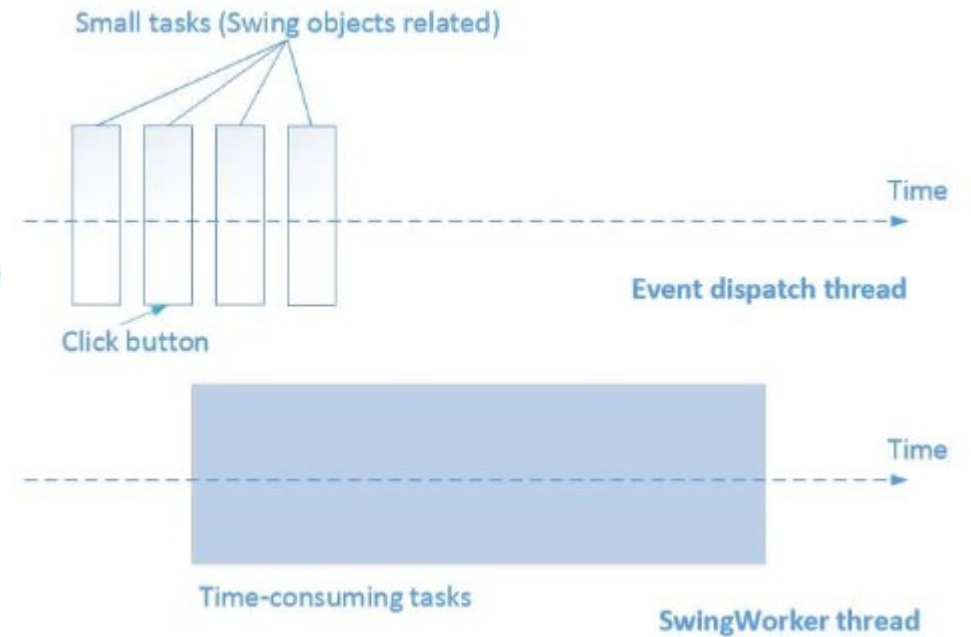
Terrible UI program:

No response after clicking buttons



Good UI program:

Can do other things with background tasks



What's More? (Multi-threading)

- When to opt for multi-threading?
 - Processing content which may require long waiting
 - Loading and decoding the saved game
 - Waiting for network in client or server
 - Calculating AI moves
 - Playing background music or sound effects
- How to use multi-threading?
 - `SwingWorker<T, V>`
 - `SwingUtilities` also provides helper functions
 -

The End