```java
public Game(Model model, GUI gui, double tile2p, long seed) {
    _model = model;
    _playing = true;
    _gui = gui;
    _probOf2 = tile2p;

    if (seed == 0) {
        _random = new Random();
    } else {
        _random = new Random(seed);
    }
}
```

DIY 2048

Java

Goal:

Build a website using Google's NGram Database that identifies hypernyms and hyponyms for a word, finds common hyponyms between multiple words, and tracks the frequency of a word's usage in published texts from 1700 to 2020.

Steps and Challenges:

The process began with building the tiles and setting up the graphical user interface (GUI). Once that foundation was laid, the real challenge started: merging tiles and updating the score. This required careful consideration of all possible moves—handling empty spaces, merging doubles, ensuring the score was updated correctly, and adding random tiles after every move. The biggest challenge, however, was organizing the project effectively. With 16 squares constantly in motion, I needed to create functions that could handle these changes both accurately and efficiently. This required a lot of trial and error as I worked to find the best methods to manage the movement and interactions of the tiles. Despite the difficulties, the project was a fantastic learning experience. Seeing it all come together was not only a testament to the effort and persistence I put in, but it also deepened my appreciation for Java and the possibilities it offers for game development.

```java
private Side keyToSide(String key) {
    return switch (key) {
        case KeyEvent.VK_UP + "", "\u2191" -> NORTH;
        case KeyEvent.VK_DOWN + "", "\u2193" -> SOUTH
        case KeyEvent.VK_LEFT + "", "\u2190" -> WEST;
        case KeyEvent.VK_RIGHT+ "", "\u2192" -> EAST;
        default -> throw new IllegalArgumentException
    };
```

Takeaways:

- Helper functions clean coding projects.
  - This project had so many different moving parts, both metaphorically and literally speaking. After a while, I noticed a lot of repetition in my code that led to bugs from typos or slight errors. Catching this relatively early in the process, I began implementing helper functions and I continue to remind myself to use such functions when working on more large-scaled tasks.
- Reviewing code helps in learning.
  - The best tactic to simplify and clean my code was something that I picked up on walearn from my own coding experiences. By going through this project, line by line, I caught many things that I could improve on or re-write all together. This simple practice helped to drill in skills and continues to help improve my coding.
- Don't be afraid of trial and error.
  - At first I was intimidated by the sheer number of ways this project could be completed. However, I found it useful to lay out my possibilities and try to maximize efficiency through coding. This technique helped me in later projects when I was making similar coding decisions.

```java
private Tile generateNewTile(int size) {
    int c = _random.nextInt(size), r = _random.nextInt(size);
    int v = _random.nextDouble() <= _probOf2 ? 2 : 4;

    return Tile.create(v, c, r);
}
```

Given this is a class project, I cannot put my entire code online. However, I would be more than ready to discuss the code by request.