Wordly Game - Project Update

Renn Gilbert

CSC1061: Computer Science II

March 28, 2024

Section 1

Requirements

Requirements

1. The project shall only allow the user to guess 5 letter words.
2. The project shall allow the user to enter 6 guesses.
3. The project shall compare users' guesses to the correct word, giving them feedback using colors.
4. The project shall read a list of words from **a** list of upcoming words. [1]
5. The project shall store user data and statistics in a file.
6. The project should validate users' guesses by comparing them with a dictionary file.
7. The project should allow the user to continue playing after the day's word is complete.
8. The project will use the Model-View-Controller-Interactor architecture.
   - Article on PragmaticCoding
9. The project will use AtlantaFX for GUI controls.

[1] this requirement was changed

Requirement Change

Requirement #4 was changed.

4. The project shall read a list of words from **Wordle's** list of upcoming words.

↓

4. The project shall read a list of words from **a** list of upcoming words.

Why?

After the New York Times bought Wordle, they switched the word lists many times. Now, they are curating a word each day.

Section 2

Class Diagrams

Class Diagrams

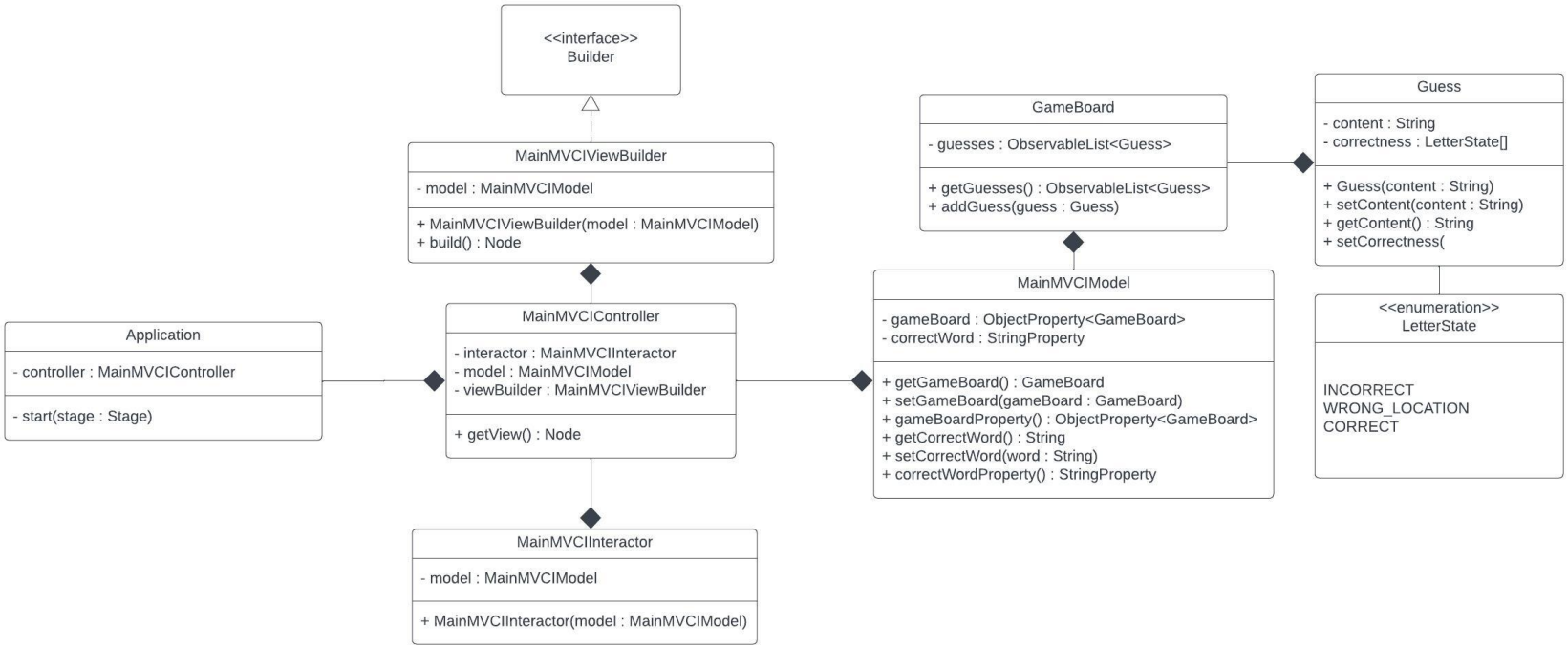


Figure 1: UML Class Diagram

Section 3

Flow Diagrams

GUI Flow Diagram

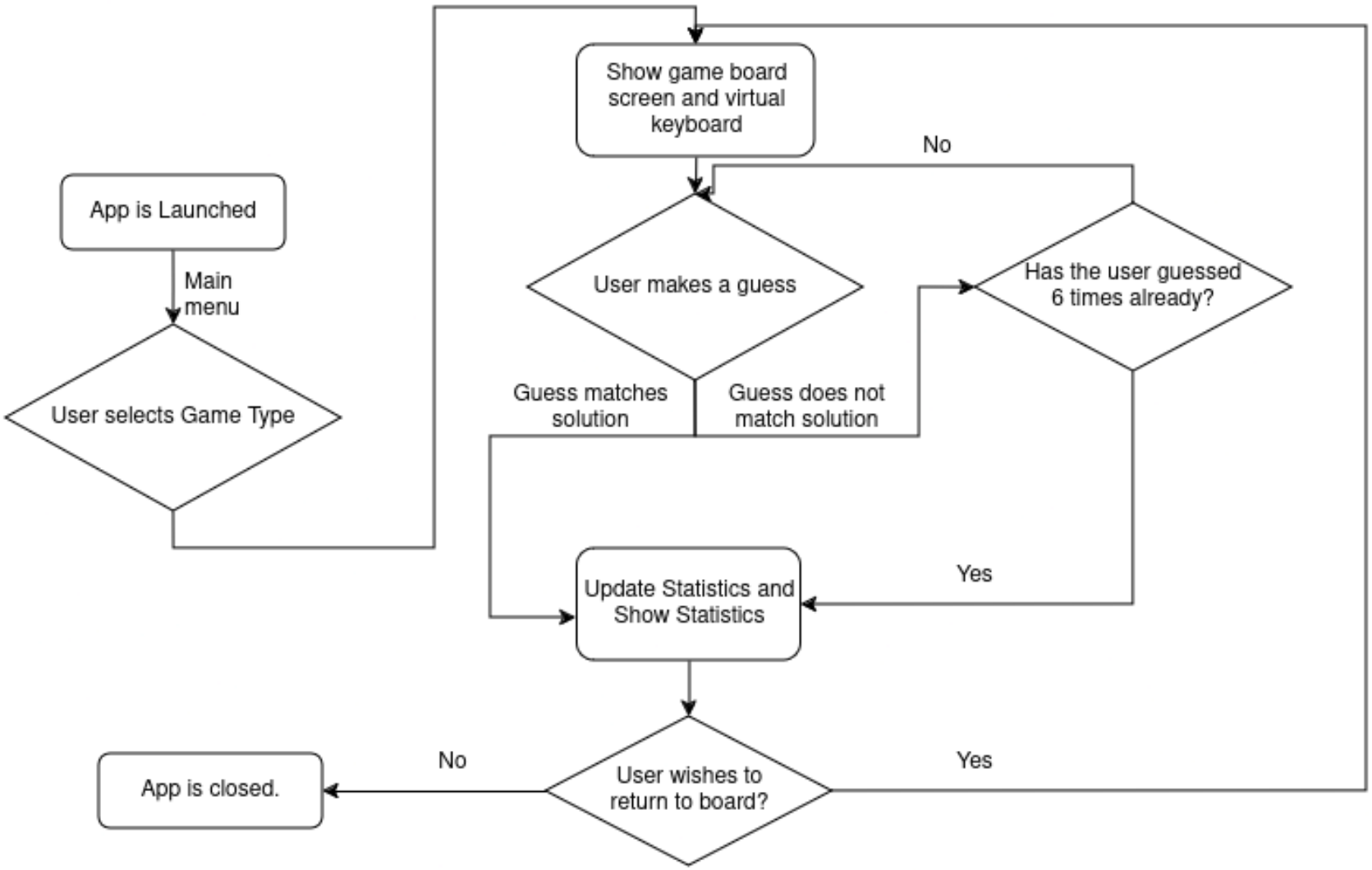
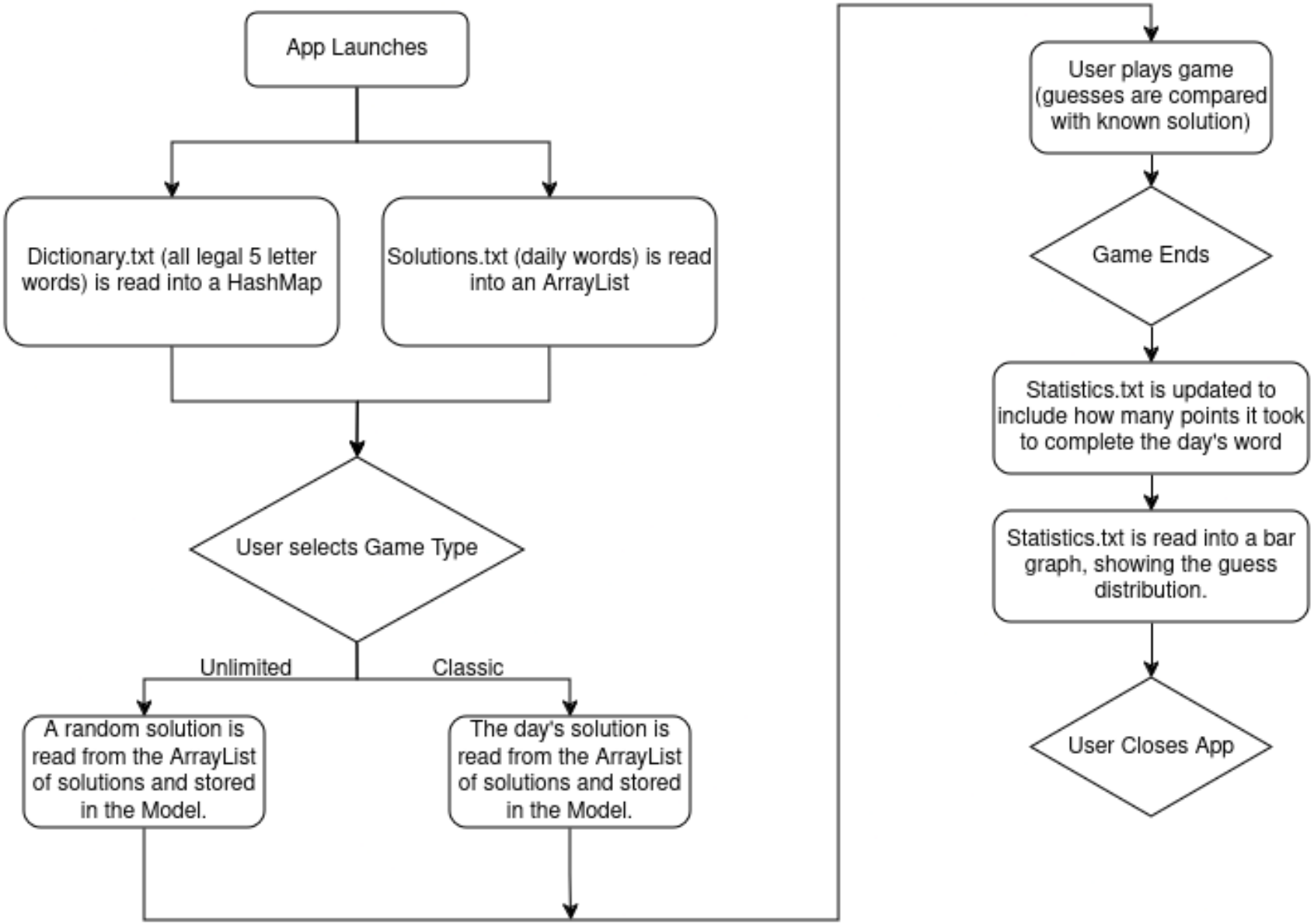


Figure 2: GUI Flow Diagram

File Flow Diagram



Data Flow Diagram

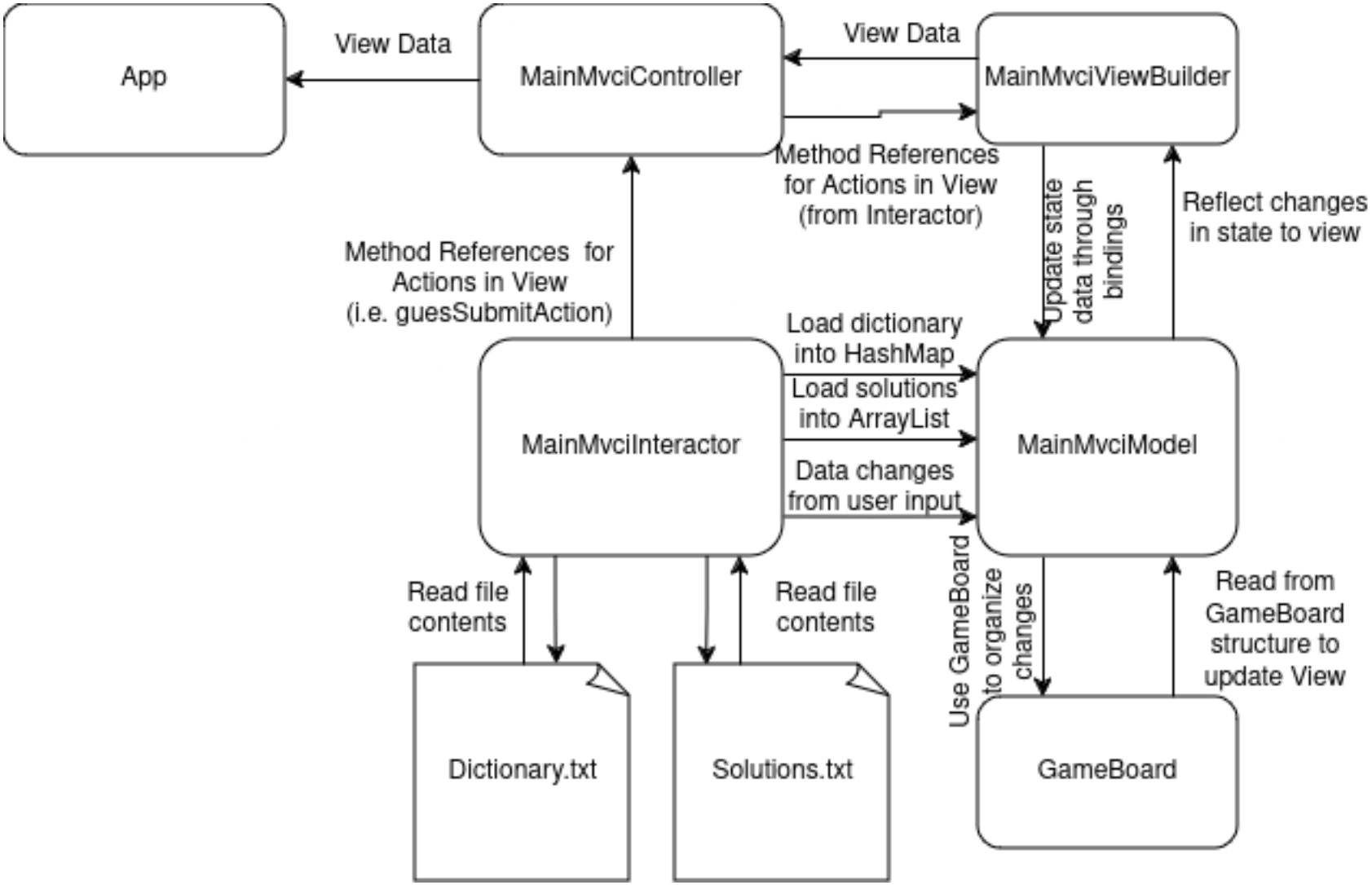


Figure 4: Data Flow Diagram

Section 4

Project Setup

GitHub Repository

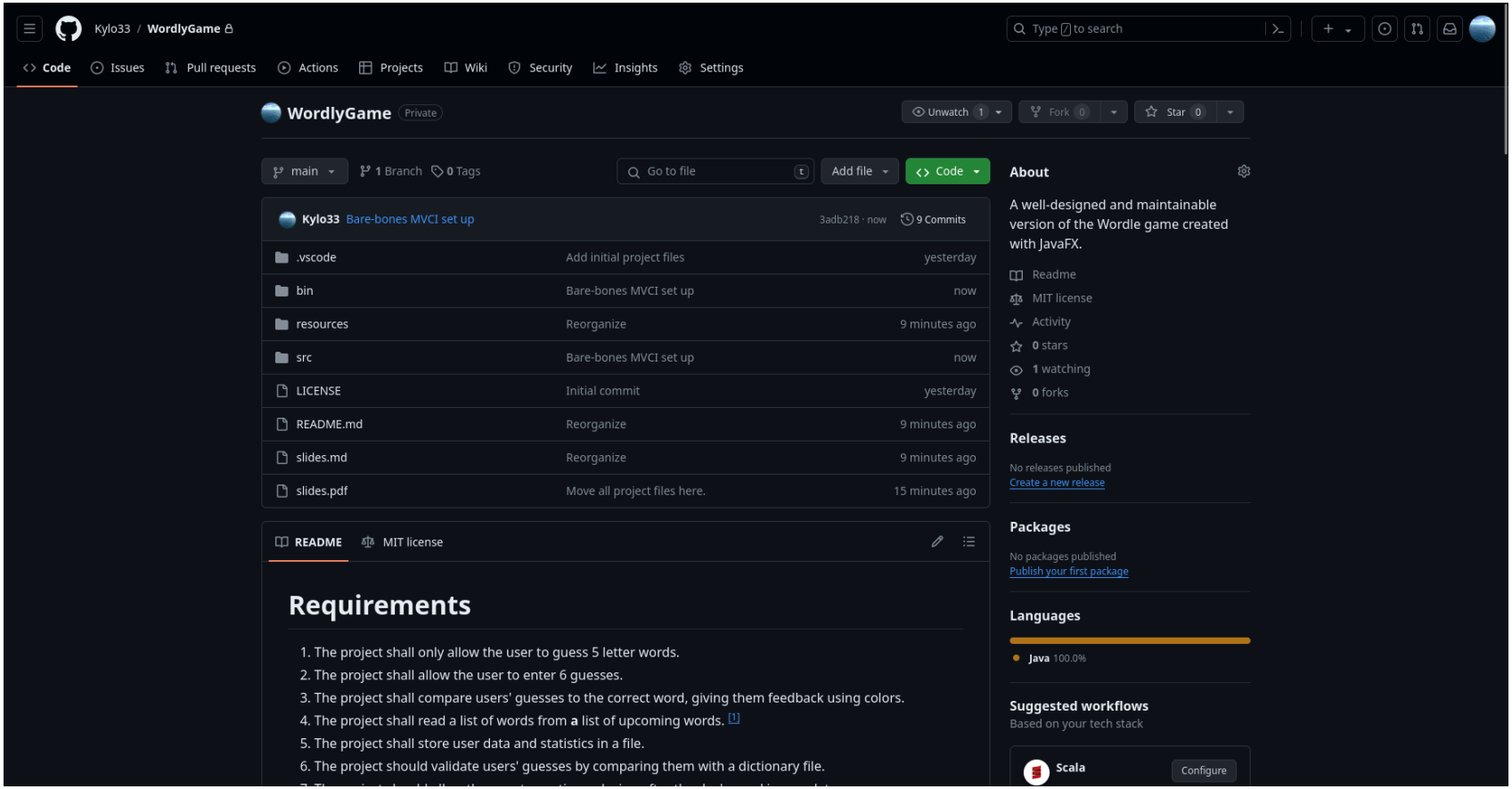


Figure 5: GitHub Initial Setup

Bare-bones setup

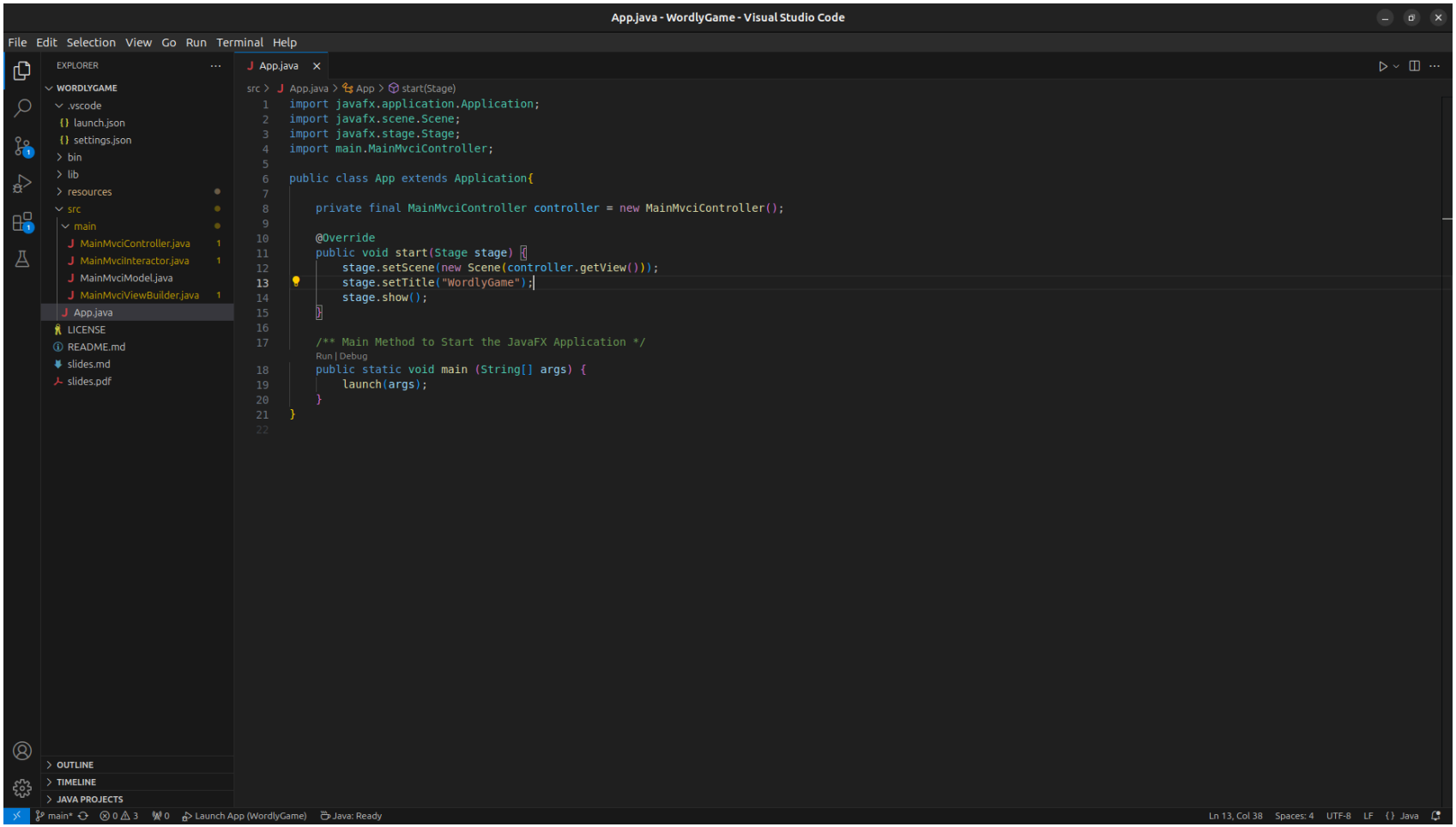


Figure 6: Bare-bones project setup with VSCode

Section 5

Research

Compiling Word Lists

For my research, I decided to compile my list of words and solutions.

dictionary.txt

To compile `dictionary.txt`, a list of valid words, I manually scraped the data from the official Wordle valid word list. Link. The resulting list is over 14,000 words long.

solutions.txt

To compile `solutions.txt`, a sequential list of upcoming solutions, I used FrequencyWords, a GitHub repository with words sorted by their frequency in a corpus (in various languages). I filtered it for words that are in `dictionary.txt`, limited it to the 2,000 most common, and shuffled it.

Script

```python
valid_guesses = []
with open("wordle-full-list.txt", "r") as infile:
    valid_guesses.extend([line.strip() for line in infile])

with open("solutions.txt", "w") as outfile:
    with open("frequencylist.txt", "r") as infile:
        result = [
            line.lower() + "\n"
            for line in [l.split(" ")[0] for l in infile]
            if line.lower() in valid_guesses
        ][:2000]
        random.shuffle(result)
        outfile.writelines(result)
```

Figure 7: Python script to filter solutions

Sample Data

dictionary.txt

```
aahed
aalii
aapas
aargh
aarti
abaca
abaci
abacs
abaft
abaht
```

solutions.txt

```
whine
spell
saved
parts
drank
sites
ducky
fatty
steak
stock
```

Section 6

The End

The End

Figure 8: NYT Wordle Image