Wordly Game - Project Update

Renn Gilbert

CSC1061: Computer Science II

March 28, 2024

Section 1

Requirements

Requirements

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

Section 2

Demo

Section 3

Source Code

Exciting Piece of Code

I made a simple NotificationSystem class as a general way to handle
sending notifications:

```java
public class NotificationSystem {
    private final List<Consumer<String>> messageRecievers =

    public void sendMessage(String message) {
        messageRecievers.forEach(c -> c.accept(message));
    }

    public void addMessageReciever(Consumer<String> stringC
        messageRecievers.add(stringConsumer);
    }

    public void removeMessageReciever(Consumer<String> stri
        messageRecievers.remove(stringConsumer);
    }
```

How I Used It

GameMvciModel.java

```java
private final NotificationSystem notificationSystem = new N
```

GameMvciInteractor.java

```java
if (!model.getDictionary().contains( guessWord.toLowerCase())) {
model.getNotificationSystem().sendMessage( "Not in word list");
return; };
## How I Used It

### GameMvciViewBuilder.java

```java
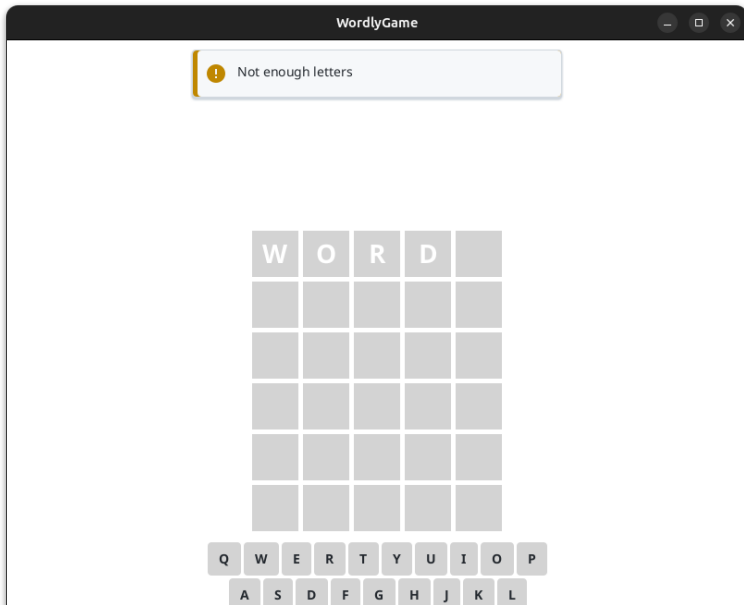private void addPopups(StackPane result) {
      model.getNotificationSystem().addMessageReciever(mess
          Notification notification = new Notification(messa

          // Styles omitted
```

End Result

Section 4

The End

The End

Figure 2: NYT Wordle Image