Wordle - DesignManual

Introduction:

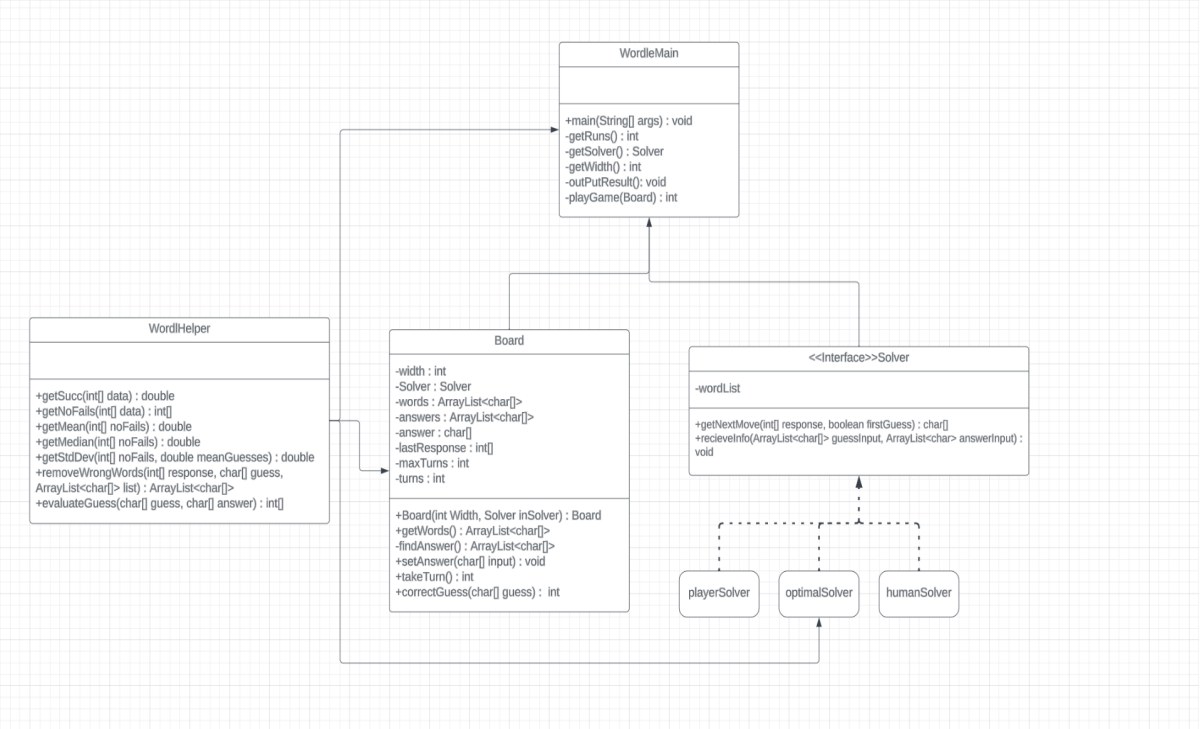
The system is broken up into three main classes. The WordleMain class controls and manages settings and creates instances of the Board Class and runs them. The aforementioned Board Class runs the actual instances of the games within the specifications provided by WordleMain. It uses the four Solver classes which all implement the Solver interface and provide guesses based on responses provided by the Board. All of this is aided by WorldeHelper which contains various methods to help many of the classes within this project.

User Stories:

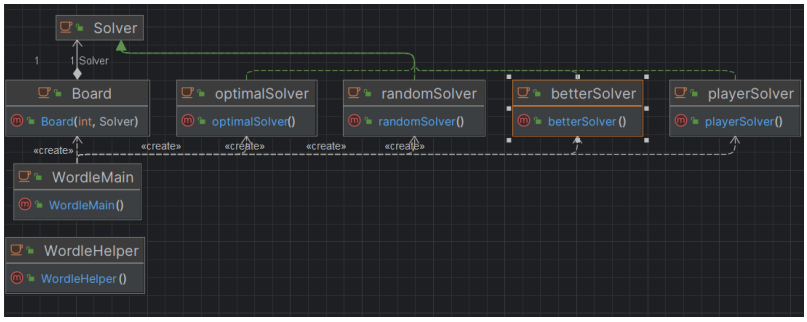
Casual Fun Play: People can quickly open up a game and be prompted with a brief series up questions and then launch into a personal game of Wordle. They can also customize the lengths of the Words. This is one aspect that was seriously held back by incomplete work. The experience of running the program through the console is far inferior to a proper UI. It is also lacking customizability features such as the number of guesses and Word difficulty.

Performance Analysis: The current iteration of the program contains three Algorithmic Solver which are described in detail within the Readme and their respective Files. Users can use these to run simulations on the different game modes and gather data on their performance. This can be used to get better yourself or try and design an ever-better algorithm. They have also been designed in such a way that they will easily integrate with new features. If variable term limits are added it would cause no problems. If you wanted to devise a training tool that would tell you what to guess in certain situations you could do that as well.

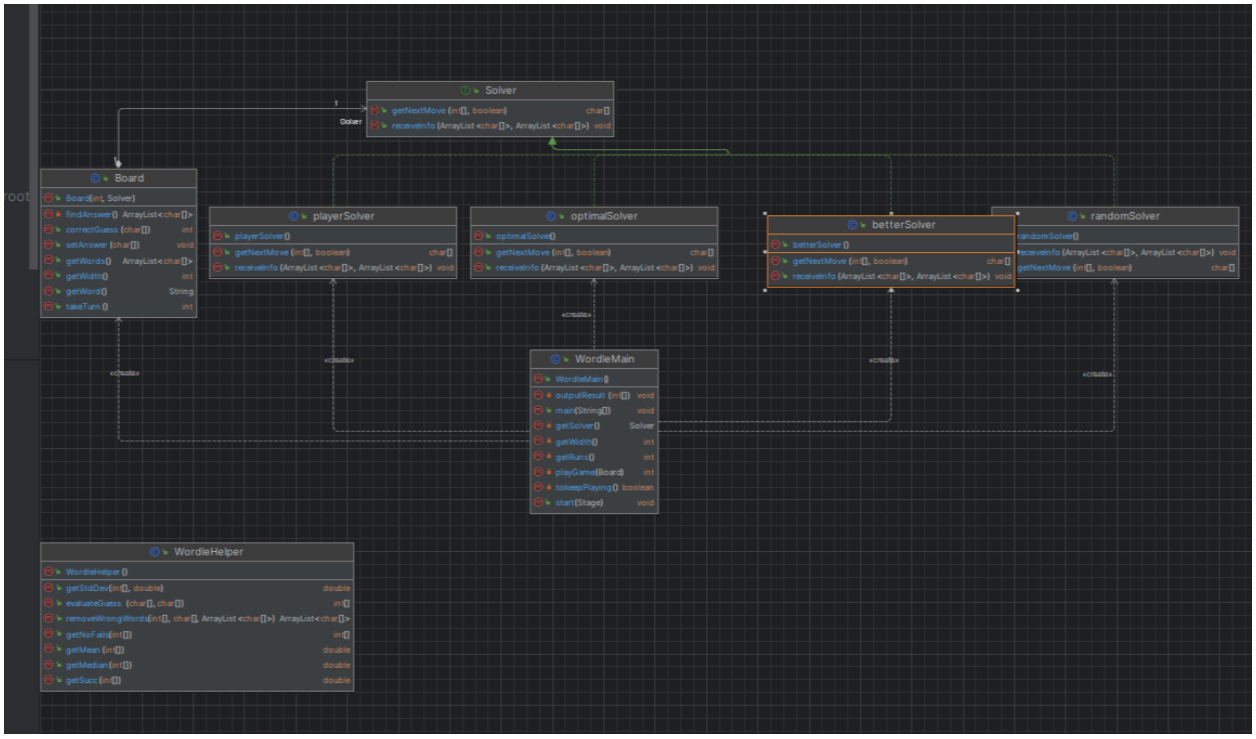
Object-Oriented-Design:



This UML Class Diagram showcases the relationship between the Main, Board, and Helper Class along with the Solver interface.



This version of the IntelliJ UML class diagrams showcases dependencies between class. Notice Worlde Helper is not connected to anything.



Citations

Sources and Inspiration for Solvers:

OptimalSolver: https://www.youtube.com/watch?v=v68zYyaEmEA

BetterSolver: https://www.youtube.com/watch?v=yuGUa-krYDA&t=266s

4-letter vocab txt: https://github.com/cherdt/fourxfour/blob/master/sorted4.dictionary

5-letter vocab txt: https://github.com/Morgenstern2573/wordle_clone/blob/master/build/words.js

6-letter vocab txt: https://github.com/CameronDeweerd/Wordle-VI/blob/main/answer_list.txt

Fxml style example: https://github.com/jpkhawam/WordleFX/tree/master

https://www.freecodecamp.org/news/build-a-wordle-clone-in-javascript/