Programming Project / C++
-=Wordle Clone=-

Student 1: Mihăilescu Bogdan **Student 2: Dunca Daniel-Răzvan**

I.Task:

Student 1 develops the core game logic:

-selecting a random 5-letter word from a dictionary

-validating each guess against the secret word

-computing the feedback pattern (G = green, Y = yellow, B = grey)

-storing the selected word in secret.txt

Student 2 builds the user-facing tools:

-taking player guesses from command-line arguments

-displaying feedback in a readable format (with ANSI coloring if desired)

-logging guess history and computing scores

-saving results in scores.txt and printing the top 10 leaderboard

II. We use the following classes:

Letter: char ch; LetterState state

Word: std::array<Letter, 5> letters

Game: Word secret; std::vector<Word> guesses; int maxTries

Score: std::string playerName; int tries; double timeElapsed; std::string date

Relationships:

Word has 5 × Letter (composition)

Game uses Word as its secret and aggregates guesses

Score is independent and serializable

III.File Structure:

secret.txt:

holds the secret word for the current game.
<word>

guesses.txt:

holds the guesses and their feedback pattern
<guess1>

<guess2>

words.txt:

dictionary containing available words to pick from
<word1>
<word2>

scores.txt:

holds the scores for the players

<player name> <right guesses> <wrong guesses> <number of times played>
…

IV. Executable interaction:

wordle_core.exe:

randomly picks a word from words.txt and saves it to secret.txt

wordle_core.exe:

returns 0 (invalid) or 1 (valid) depending on words.txt

wordle_guess.exe:

reads secret.txt, compares with guessed word, outputs GYBBG pattern and logs the guess in guesses.txt (green-right(G), yellow-right letter wrong place(Y), grey - wrong letter, wrong place(B).

wordle_stats.exe:

# displays scores from scores.txt

wordle_reset.exe:

# clears secret.txt and guesses.txt for a new game