

Wordlemaxxers

...

Izaak White, Matt Taylor, Josh Moner
&
Conner Casey

Team Information

- Everyone on the team is using Windows 11.
- Each member has a personal laptop to access code.
- Our team meets every Saturday at 14:00 EST.

Project Info:

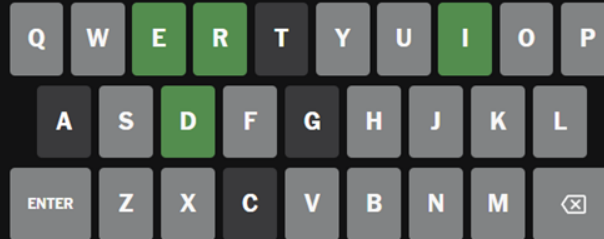
- Description:
 - Recreate the game Wordle
 - Create an AI that solves the games' puzzles.
- Goals:
 - Fully implement the game and webapp design.
 - Have an AI that works efficiently.

GUI Mockup

WordleMaxxers

Wordle Bot Guess:
Fried

T	R	A	C	E
G	R	I	D	E
D	R	I	E	D



Initial Code Base:

```
#AUTHOR: IZAAK WHITE
#THIS IS THE CURRENT FILE

import random
import pygame
import sys

pygame.init()

SCREEN_WIDTH = 600
SCREEN_HEIGHT = 600 # Increased height
WHITE = (255, 255, 255)
BLACK = (0, 0, 0)

screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
pygame.display.set_caption("Wordle Game")

font = pygame.font.Font(None, 36)

def draw_text(text, color, x, y):
    text_surface = font.render(text, True, color)
    text_rect = text_surface.get_rect(center=(x, y))
    screen.blit(text_surface, text_rect)

def processGuess(T_answer, T_guess, num_of_guesses):
    T_answer = T_answer.lower()
    T_guess = T_guess.lower()

    if len(T_guess) != len(T_answer):
        return "Please enter a {}-letter word.".format(len(T_answer)), num_of_guesses

    if T_guess not in word_list:
        return "The word is not in the word list.", num_of_guesses

    clue = ""
    correct_positions = set()

    max_attempts = 6
    input_text = ""
    feedback_message = ""
    guess_list = []
    run = True
    while run:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                run = False
            elif event.type == pygame.KEYDOWN:
                if event.key == pygame.K_RETURN:
                    guess = input_text
                    feedback, num_of_guesses = processGuess(answer, guess, num_of_guesses)
                    feedback_message = feedback
                    if feedback == "" * len(answer) or num_of_guesses >= max_attempts:
                        run = False

                    input_text = "" # Clear the input box

                elif event.key == pygame.K_BACKSPACE:
                    input_text = input_text[:-1]
                else:
                    input_text += event.unicode

        screen.fill(BLACK)

        draw_text("Wordle Game", WHITE, SCREEN_WIDTH // 2, 50)
        draw_text("Attempts left: {}".format(max_attempts - num_of_guesses), WHITE, SCREEN_WIDTH // 2, 100)

        draw_text("Type a {}-letter word:".format(len(answer)), WHITE, SCREEN_WIDTH // 2, SCREEN_HEIGHT - 400)
        draw_text(input_text, WHITE, SCREEN_WIDTH // 2, SCREEN_HEIGHT - 350 + num_of_guesses*30)
        draw_text(feedback_message, WHITE, SCREEN_WIDTH // 2, SCREEN_HEIGHT - 300 + num_of_guesses * 30) # Adjusted position

        pygame.display.flip()

    pygame.quit()
    sys.exit()
```

Projected Tool Stack:

- Potential Candidates:
 - VScode in Windows - Python with Pygame - Django Web API (current)
 - VScode in WSL - HTML - Tailwind CSS - React - Next.js - REST API with Supabase

Work Summary:

- Previous work:
 - Initial code base for game written by Izaak, testing of code base done by Matt
 - Recreation of current code base using different tool stack written by Josh
- Future projections:
 - We will decide as a team what Tool stack will work best moving forward
 - we will work on refining the AI