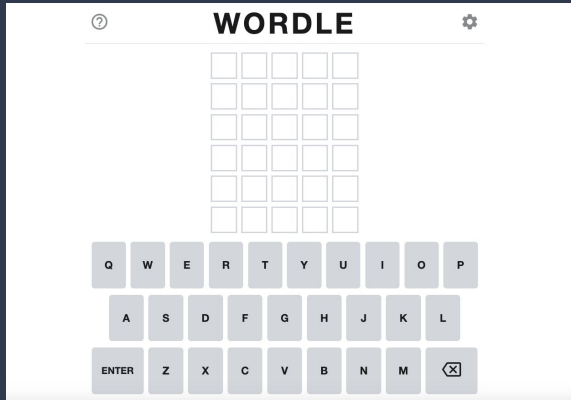


# Cozmo Wordle

Minwoo Eom and Hongfei Ji

A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the slide.

# Project Purpose

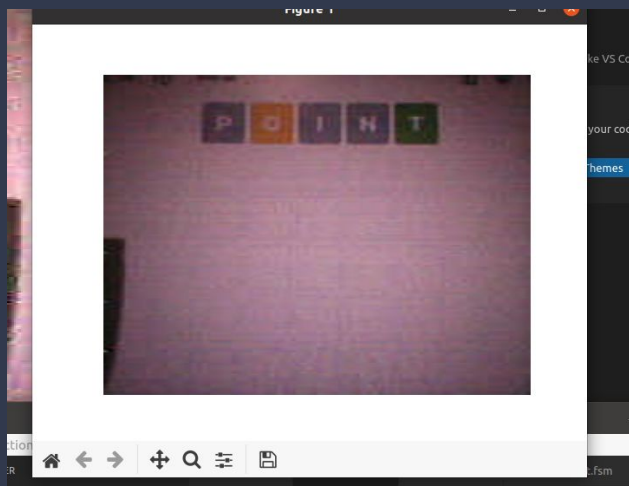


- Integrate ChatGPT with Cozmo to play NYT game Wordle.
- Interact directly with Cozmo by allowing users to select their word choice by tapping cubes to make the game more engaging and interactive.
- Enhance Cozmo's interaction by programming it to express emotions such as excitement, frustration, or joy, reflecting the game's progress.

# Implementation Workflow

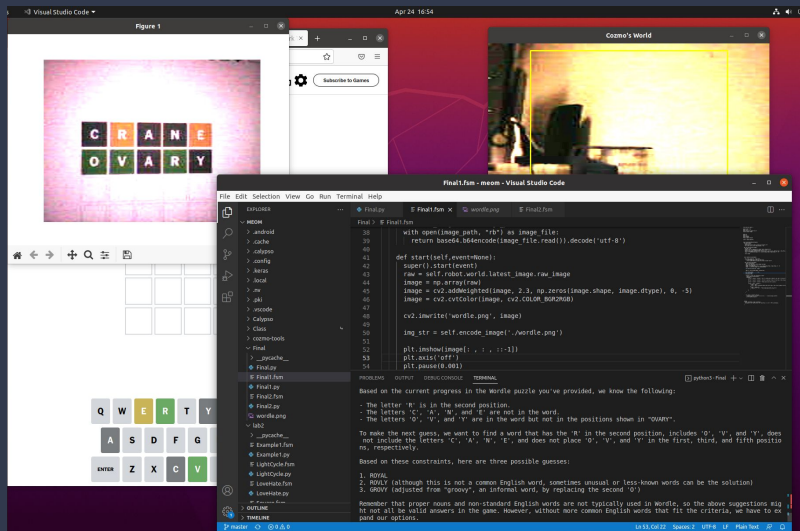
- At the start of the game, Cozmo presents three Wordle guesses for the user to consider.
- Users interact with Cozmo by tapping one of three designated cubes to select their preferred guess.
- After entering the word in Wordle, the user provides feedback to indicate the color results from Wordle
- Cozmo expresses emotions based on the feedback and uses this feedback to update its knowledge base, enhancing its ability to make more accurate guesses as the game progresses, and presents three new guesses based on the accumulated data.

# 1st Approach



- Used GPT-4's vision capabilities to analyze a picture of the game's progress for generating the top three Wordle guesses.
- Image quality from Cozmo's camera was too low and faced issues GPT struggled to interpret the unclear images provided by Cozmo's camera.

# 2nd Approach



- Applied image processing techniques using the cv2 library to enhance brightness, contrast, saturation, and sharpness of the game's images captured by Cozmo.
- Achieved significant improvements in image quality, enabling GPT to discern the letters more effectively.
- Despite the clearer images, continued to face challenges with accurate color differentiation, a critical element in the Wordle game.

# Results

Substitute for  
image recognition:  
color input

Premises for  
each guess

The image shows a browser window displaying a word game on `ordlegame.org`. The game board consists of a 5x5 grid of colored squares. The letters on the board are: Row 1: S (yellow), P (gray), I (yellow), T (gray), E (yellow); Row 2: I (yellow), S (gray), L (yellow), E (gray), S (yellow); Row 3: S (yellow), I (gray), E (yellow), G (gray), E (yellow); Row 4: R (yellow), I (gray), S (yellow), E (gray), S (yellow); Row 5: R (yellow), I (gray), S (yellow), E (gray), R (yellow). Below the board is a keyboard with keys colored to match the board. A "You Won!" message is displayed above the keyboard. Below the keyboard is an advertisement for "Word Finder".

To the right of the browser window is a terminal window showing the game's internal state and player input. The terminal output includes the following text:

```
Speaking: 'Choose your guess'
spite, s, p, i, t, e
Speaking: 'spite, s, p, i, t, e'
Speaking: 'Waiting for you to enter'

result? ybyby
ybyby
C> current color in process: ybyby, round: 1
[[['s=yellow', 'p=gray', 'i=yellow', 't=gray', 'e=yellow']]
Speaking: 'Thinking of a new guess'
['slide', 'aisle', 'isles']
Speaking: 'Choose your guess'
isles, i, s, l, e, s
Speaking: 'isles, t, s, l, e, s'
Speaking: 'Waiting for you to enter'

result? yygb
yygb
C> current color in process: yygb, round: 2
[[['s=yellow', 'p=gray', 'i=yellow', 't=gray', 'e=yellow'], ['i=yellow', 's=yellow', 'l=gray', 'e=green', 's=gray']]
Speaking: 'Thinking of a new guess'
['stege', 'setze', 'petse']
Speaking: 'Choose your guess'
stege, s, t, e, g, e
Speaking: 'stege, s, t, e, g, e'
Speaking: 'Waiting for you to enter'

result? ygybb
ygybb
C> current color in process: ygybb, round: 3
[[['s=yellow', 'p=gray', 'i=yellow', 't=gray', 'e=yellow'], ['i=yellow', 's=yellow', 'l=gray', 'e=green', 's=gray'], ['s=yellow', 't=green', 'e=yellow', 'g=gray', 'e=gray']]
Speaking: 'Thinking of a new guess'
['resin', 'isles', 'rises']
Speaking: 'Choose your guess'
rises, r, t, s, e, s
Speaking: 'rises, r, t, s, e, s'
Speaking: 'Waiting for you to enter'

result? ggggb
ggggb
C> current color in process: ggggb, round: 4
[[['s=yellow', 'p=gray', 'i=yellow', 't=gray', 'e=yellow'], ['i=yellow', 's=yellow', 'l=gray', 'e=green', 's=gray'], ['s=yellow', 'l=green', 'e=yellow', 'g=gray', 'e=gray'], ['r=green', 't=green', 's=green', 'e=green', 's=gray']]
Speaking: 'Thinking of a new guess'
['riser', 'rates', 'reins']
Speaking: 'Choose your guess'
Marker rejected! id= 35 tvecc= [ 6.60030157 -16.52919329 162.76848978] rvecc= [ 0.10791762 0.02631775 -3.09897907]
riser, r, t, s, e, r
Speaking: 'riser, r, t, s, e, r'
Speaking: 'Waiting for you to enter'

result? ggggg
ggggg
C> current color in process: ggggg, round: 5
current color: ggggg
Speaking: 'Good job!'
Quit (core dumped)
hongfelj@oin: ~/final$
```

# Limitations

Hardware constraints: Cozmo's camera

Algorithm constraints: GPT's intrinsic flaw in leveraging context

- The API can't recognize the correct color even when given a clear screenshot.
- The API can't give correct guesses even after elaborating the game rule, excluding the corner cases, and illustrating the context's data structure.

# Future Extensions and Improvements

## Improving Color Recognition:

- Explore advanced color correction algorithms or machine learning models specifically trained to differentiate between subtle shades in poor lighting and electronic screen lighting conditions.

## Enhancing Word Guess Quality:

- Implement a specialized Wordle-solving algorithm or use a pre-trained model that is optimized for word puzzle games rather than relying solely on general-purpose GPT predictions.
- Develop a feedback loop where Cozmo learns from past games to improve word predictions over time based on what has worked and what hasn't.



# Project Demo

