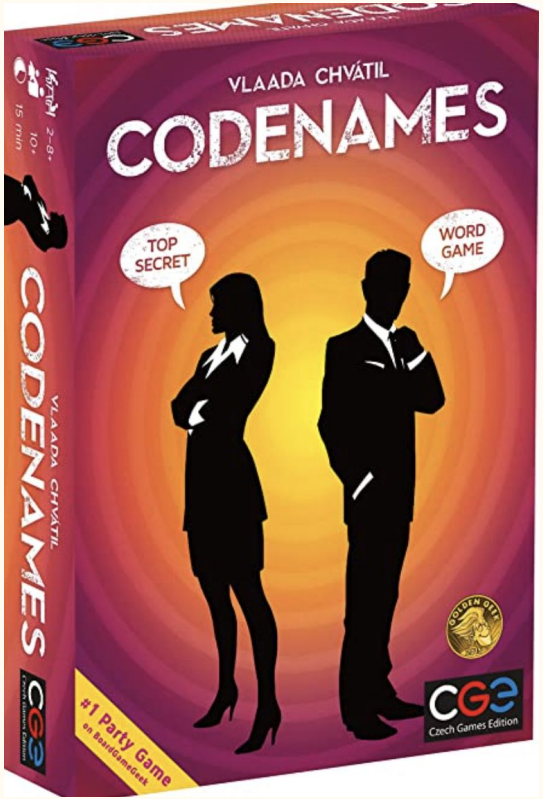
Cheating Codenames

Author: Nelli Aydinyan

Special gratitude to all instructors for their help

Thesis

- Language barriers
- Motivation to win
- A program to help win this game



Codenames: Original

board

grid

Codenames: mid-game

The Word Board

How the original photo looks:

The colour scale is **0-255**, respectively **black-white**

Any pixel above **10** was set equal to **255**

Board post filtering

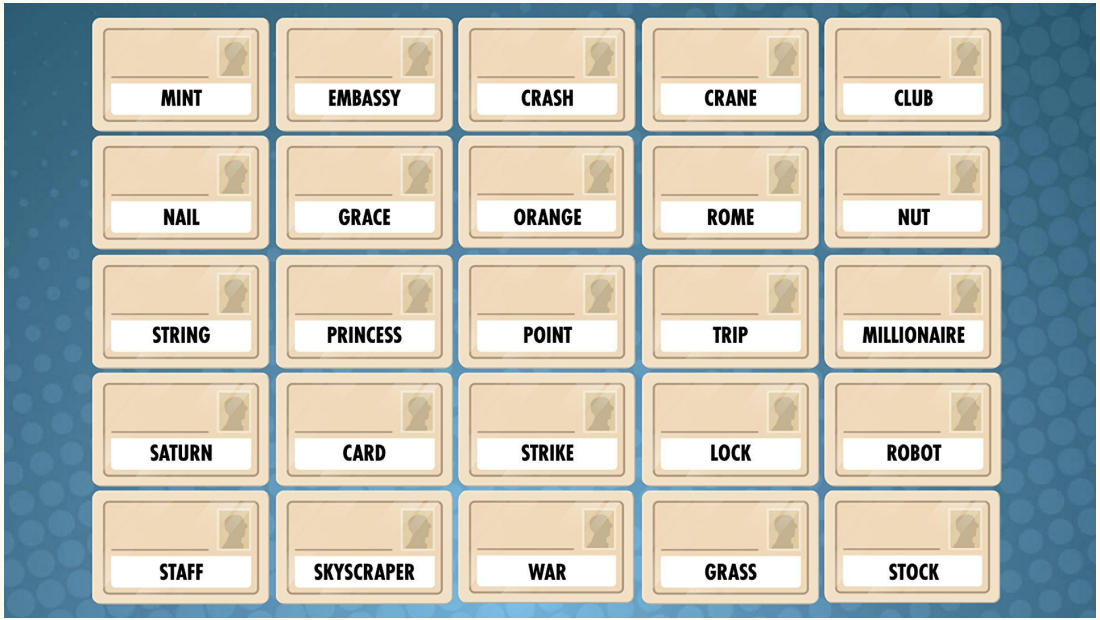
Words are hard to read

Some words are either missing or have missing spots

My coffee table markings and the tile borders are easier to see than the words

High resolution board

How the original photo looks:



The colour scale is **0-255**, respectively **black-white**

Any pixel above **10** was set equal to **255**

High resolution board post filtering

Easy to read words for both the human and the computer

Pytesseract read these without a single issue and output them in the direction of columns

Quick formatting of the output string and I had a matrix of words

Word classification grid

The colours on the grid represent what the teams have to guess or avoid:

- **Blue and red** are each for two teams
- **Neutral** doesn't result in loss but terminates the ongoing turn
- **Black** means that the game is automatically lost

OpenCV

- Initial release on 2000 by Intel
- A library aimed at real-time computer vision
- Written for C languages

- Python has a version **BUT** it is unofficial:
  - No documentation
  - How does it even work?
  - Most examples are by people figuring out this library themselves

OpenCV .matchTemplate

Matching the coin onto the image

Algorithm:

1. Read in both the template and the actual images
2. Use the function to get the likelihoods
3. Where the likelihood is above a certain percentage - draw a border

What I got...

Given threshold for both photos: 0.4

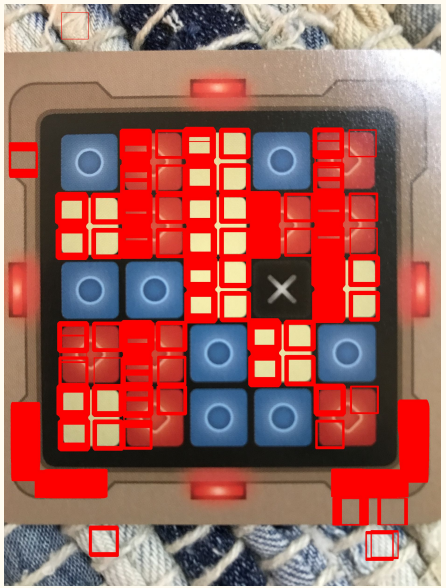
High-res :

- All blues are identified
- The problem seems to be in the border size

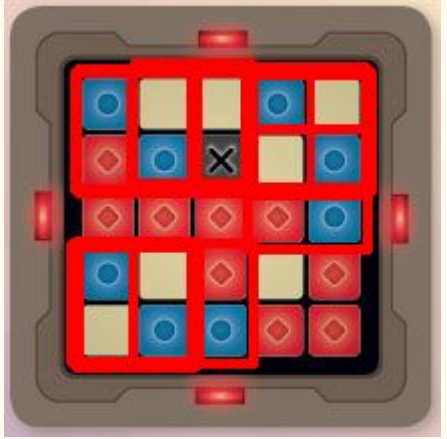
Low-res :

- Both red and neutral are identified, together with some of the border and my rug
- The borders are a quarter of the size of the actual image

Searching for red on low-res



Searching for blue on high-res



Word Vectors/Building the assumptions

- User input: how many words would they like to combine?

- Program:
  - Finds all the combinations of n words using count-permutations
  - Applies the model to each combination
  - Finds the top 3 clues
  - Using the threshold filters out the combinations that are too distant
  - Outputs the results: the word combination and the clues

Results

okay

```
The word combination is:
['crash', 'crane']
The clues are:
['profil', 'crooks', 'dreyer']
```

```
The word combination is:
['orange', 'grass']
The clues are:
['colored', 'striped', 'agate']
```

```
The word combination is:
['card', 'stock']
The clues are:
['cards', 'debit', 'vendor']
```

meh

```
The word combination is:
['string', 'grass', 'nut']
The clues are:
['gut', 'comb', 'metallic']
```

```
The word combination is:
['card', 'club', 'stock']
The clues are:
['cards']
```

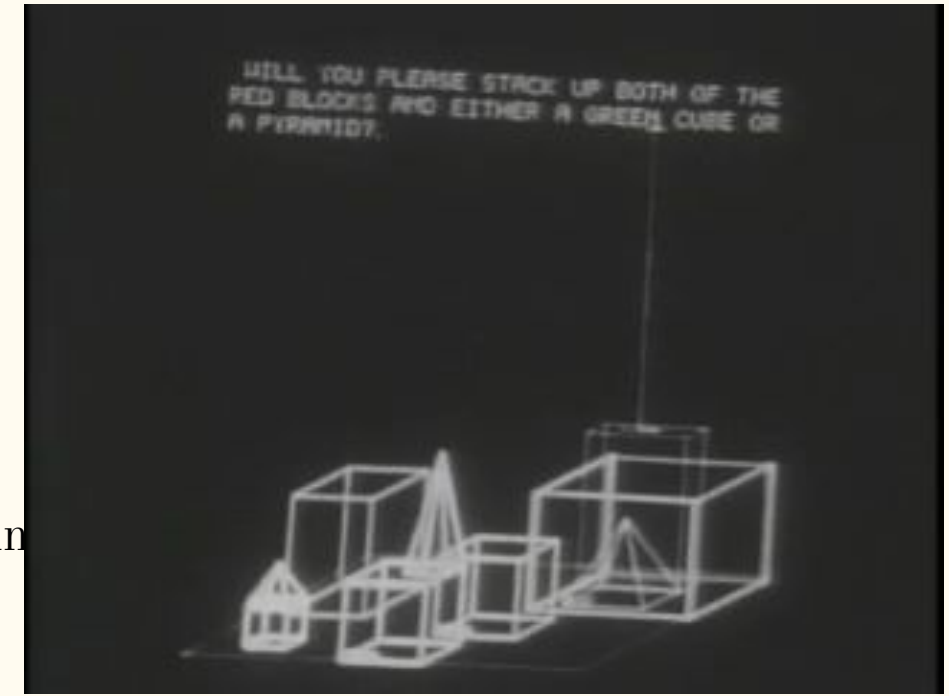
The best result

```
The word combination is:
['skyscraper', 'robot']
The clues are:
['shrdlu', 'runequest', 'humanoid']
```

SHRDLU

- 1968-1970 NLP program
- Computer program that builds blocks by understanding sentences

- One of the first AI programs responsible to the sparked interest in this field



Thank for your attention