

T1A3

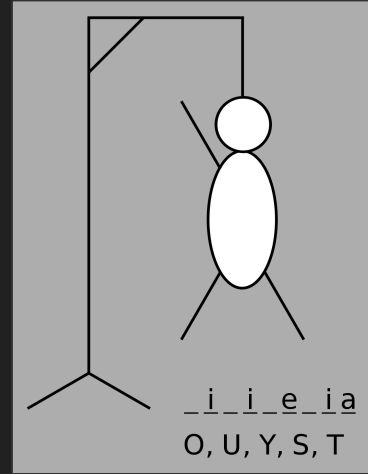
Coder Academy Terminal App
Assignment

Raphael Florea

Hangman application introduction

Features:

- Play hangman against computer, where computer randomly chooses word to be guessed with 10 available wrong guesses
- Print out picture of hangman getting closer to being hanged each failed guess
- After one round of guessing word, next round involves word but is longer by one letter (eg round 1: 'cats', round 2: 'giant')
- Multiplayer feature that involves scoring system, to determine who can get best score
- Hint system where if player is running out of guesses, computer gives a hint for a single letter



Walkthrough of terminal - 1st bash script

```
Marianas-MacBook-Pro:src ralphfloRafrea$ ls
__pycache__          index.sh              testing.py
file_opener.sh        main.py               wordlist.txt
hangman_classes.py    run_game.py          wordlist_builder.py
hangman_gameloop.py   run_game_cheats.py
Marianas-MacBook-Pro:src ralphfloRafrea$ bash index.sh
Python 3 is installed. Able to run program.
Enter 't' to run test, enter 'c' to run program with cheats.
Enter anything else to run program normally.
```

First checks Python 3 is installed,
then loads next bash script.

Otherwise, doesn't load program.

```
$ index.sh
1  #!/bin/bash
2
3  #Check if Python 3 is installed, exit if not
4  if [[ -x "$(command -v python3)" ]]; then
5      echo "Python 3 is installed. Able to run program."
6      bash file_opener.sh
7  else
8      echo "Python 3 is not installed. Install before running."
```

Walkthrough of terminal - 2nd bash script

```
Marianas-MacBook-Pro:src ralphfloRafrea$ bash index.sh
Python 3 is installed. Able to run program.
Enter 't' to run test, enter 'c' to run program with cheats.
Enter anything else to run program normally.
x
```

```
-----
WELCOME TO HANGMAN!
```

```
-----
Press '3' to quit.
```

```
Enter '1' for singleplayer, enter '2' for multiplayer: █
```

Play normally, given no special inputs

Creates, makes, deactivates,
hen removes virtual environment

```
$ file_opener.sh
```

```
1  #!/bin/bash
```

```
2  #Create virtual environment, then activate
```

```
3  python3 -m venv hangman_venv
```

```
4  source hangman_venv/bin/activate
```

```
5  #Have option to either run test, or run program
```

```
6  echo "Enter 't' to run test, enter 'c' to run program with cheats."
```

```
7  echo "Enter anything else to run program normally."
```

```
8  read input
```

```
9  if [[ "$input" == "t" ]] ; then
```

```
10     # Install pytest, then run program
```

```
11     pip install pytest
```

```
12     pytest testing.py
```

```
13 elif [[ "$input" == "c" ]] ; then
```

```
14     #Run program file that enables cheats, no dependencies needed to be installed
```

```
15     python3 run_game_cheats.py
```

```
16 else
```

```
17     #Run program, no dependencies needed to be installed
```

```
18     python3 run_game.py
```

```
19 fi
```


```
20 #Deactivate then remove virtual environment after finishing
```

```
21 deactivate
```

```
22 rm -r hangman_venv
```

Walkthrough of terminal - Open game with cheats

Enter 'c' for game with cheats




```
Marianas-MacBook-Pro:src rafloRafrea$ bash index.sh
Python 3 is installed. Able to run program.
Enter 't' to run test, enter 'c' to run program with cheats.
Enter anything else to run program normally.
c
```

```
WELCOME TO HANGMAN (with cheats)!
```

```
-----
Press '3' to quit.
```

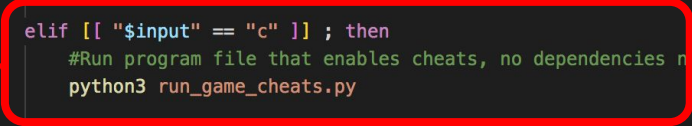
```
Enter '1' for singleplayer, enter '2' for multiplayer:
```



Play game with cheats

Cheats = Show word to be guessed

```
$ file_opener.sh
1  #!/bin/bash
2
3  #Create virtual environment, then activate
4  python3 -m venv hangman_venv
5  source hangman_venv/bin/activate
6
7  #Have option to either run test, or run program
8  echo "Enter 't' to run test, enter 'c' to run program with cheats."
9  echo "Enter anything else to run program normally."
10 read input
11 if [[ "$input" == "t" ]] ; then
12
13     # Install pytest, then run program
14     pip install pytest
15     pytest testing.py
16
17 elif [[ "$input" == "c" ]] ; then
18     #Run program file that enables cheats, no dependencies needed to be installed
19     python3 run_game_cheats.py
20
21 else
22     #Run program, no dependencies needed to be installed
23     python3 run_game.py
24 fi
25
26 #Deactivate then remove virtual environment after finishing
27 deactivate
28 rm -r hangman_venv
```



Loading (automatic) tests

Enter 't' start for test to activate

10 tests

- 5 tests to test scoring system
- 5 to test letter guess system

```
Python 3 is installed. Able to run program
Enter 't' to run test, enter 'c' to run program with cheats.
Enter anything else to run program normally.
t
Collecting pytest
  Using cached pytest-7.3.1-py3-none-any.whl (320 kB)
Collecting iniconfig
  Using cached iniconfig-2.0.0-py3-none-any.whl (5.9 kB)
Collecting packaging
  Using cached packaging-23.1-py3-none-any.whl (48 kB)
Collecting pluggy<2.0,>=0.12
  Using cached pluggy-1.0.0-py2.py3-none-any.whl (13 kB)
Installing collected packages: pluggy, packaging, iniconfig, pytest
Successfully installed iniconfig-2.0.0 packaging-23.1 pluggy-1.0.0 pytest-7.3.1

[notice] A new release of pip available: 22.3.1 -> 23.1.2
[notice] To update, run: pip install --upgrade pip
===== test session starts =====
platform darwin -- Python 3.11.3, pytest-7.3.1, pluggy-1.0.0
rootdir: /Users/ralphfloRafrea/Desktop/Coder Academy Stuff/assignments/T1A3/Raph
aelFlorea_T1A3/src
collected 10 items

testing.py ..... [100%]

===== 10 passed in 0.03s =====
Marianas-MacBook-Pro:src ralphfloRafrea$
```

Walkthrough of game - single-player

Play game with cheats for demo

Cheats = Show word to be guessed

```
-----  
WELCOME TO HANGMAN (with cheats!)  
-----
```

```
Press '3' to quit.  
Enter '1' for singleplayer, enter '2' for multiplayer: 1  
You have chosen singleplayer.
```

```
*****
```

```
ROUND 1
```

```
GET READY TO GUESS WORD!
```

```
3
```

```
2
```

```
1
```

```
GO!
```

```
KONG  
Press '3' to quit.  
Guess a letter: 
```

Walkthrough of game - single-player (continued)

When guessing letters wrong,

Hangman image starts to be build,
Guesses left also reduces

```
KONG
Press '3' to quit.
Guess a letter: y
That is not correct!
```

```
----
```

```
-----
```

```
Incorrect guesses: ['Y']
```

```
Guesses remaining: 9
```

```
Press '3' to quit.
```

```
Guess a letter: e
```

```
That is not correct!
```

```
----
```

```
-----
```

```
Incorrect guesses: ['Y', 'E']
```

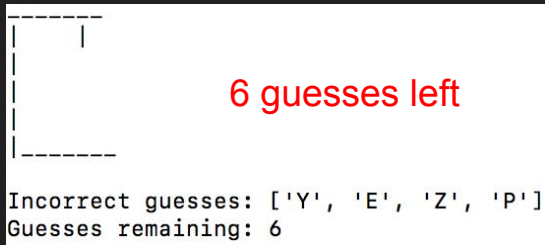
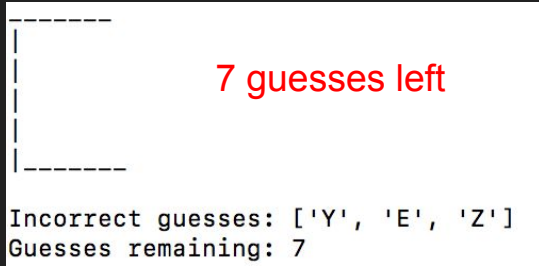
```
Guesses remaining: 8
```

```
Press '3' to quit.
```

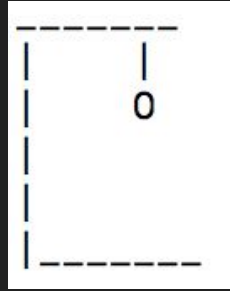
```
Guess a letter: █
```


Walkthrough of game - single-player (bad guesses)

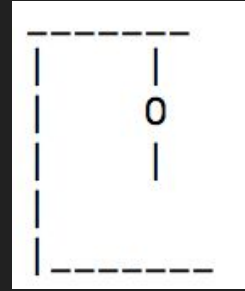
Image of hangman continually builds up



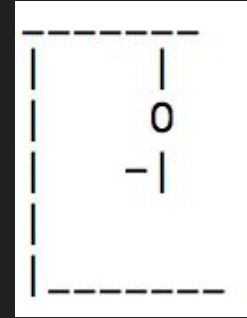
5 left



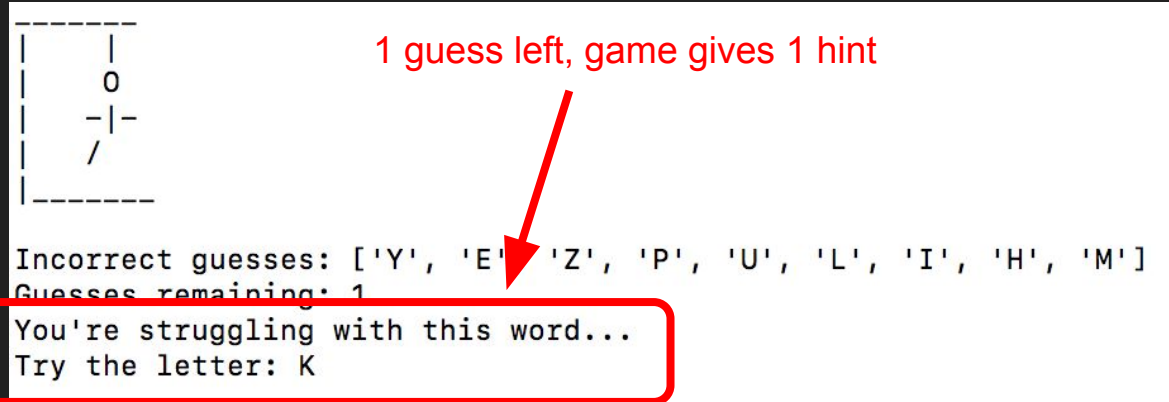
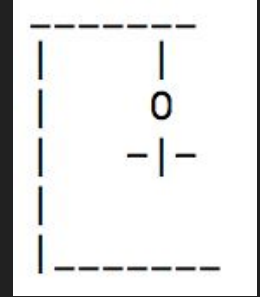
4 left



3 left



2 left



Walkthrough of game - single-player (FAILURE!)

```

|
| 0
| -|-
| / \
|
-----

Incorrect guesses: ['Y', 'E', 'Z', 'P', 'U', 'L', 'I', 'H', 'M', 'D']
Guesses remaining: 0
FAILURE!

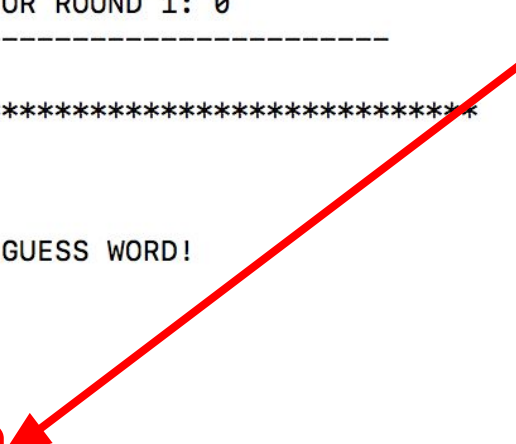
Correct word was: KONG
-----
Score for Round: 0
Streak: 0
TOTAL SCORE FOR ROUND 1: 0
-----
```

Game shows correct word

Score is 0 by default,
Streak set to 0

Walkthrough of game - single-player (2nd round)


```
-----  
Score for Round: 0  
Streak: 0  
TOTAL SCORE FOR ROUND 1: 0  
-----  
  
*****  
  
ROUND 2  
  
GET READY TO GUESS WORD!  
3  
2  
1  
GO!  
  
LOVES  
Press '3' to quit.  
Guess a letter: █
```



Word now 5 letters for next round.
Each round has word of length one letter more than previous round.

Round 1: 4 letters
Round 2: 5 letters
Round 3: 6 letters

Round 10: 13 letters



```
ROUND 3  
  
GET READY TO GUESS WORD!  
3  
2  
1  
GO!  
  
BEHALF  
Press '3' to quit.  
Guess a letter: █
```

Walkthrough of game - single-player (good guesses)

```
LOVES
Press '3' to quit.
Guess a letter: l
That is a correct letter!
```

L----

Everytime letter is guessed correctly,
it shows up in word display,
and number of guesses doesn't reduce

```
Incorrect guesses: []
Guesses remaining: 10
Press '3' to quit.
Guess a letter: o
That is a correct letter!
```

Here guesses stay at 10

LO---

Walkthrough of game - single-player (SUCCESS!)

LOVES

Word fully guess

Incorrect guesses: []
Guesses remaining: 10
GUESSED WORD CORRECTLY!

Able to get some score from round

Score for Round: 3

Streak: 1

Streak increases by 1

TOTAL SCORE FOR ROUND 2: 3

Walkthrough of game - single-player (end-game)

Note that for round 10,
word has 13 letters

ROUND 10

GET READY TO GUESS WORD!
3
2
1
GO!

EXTRAORDINARY

Press '+' to quit.
Guess a letter: e
That is a correct letter!

At end of game, receive final score,
get goodbye message

Score for Round: 36
Streak: 9
TOTAL SCORE FOR ROUND 10: 562

OVERALL SCORE: 562

Thanks for playing! Bye bye!

Walkthrough of game - multiplayer

```
-----  
WELCOME TO HANGMAN (with cheats!)  
-----
```

```
Press '3' to quit.  
Enter '1' for singleplayer, enter '2' for multiplayer: 2  
You have chosen multiplayer.
```

```
*****
```

```
ROUND 1 FOR PLAYER 1
```

```
GUESSED WORD CORRECTLY!
```

```
*****
```

```
ROUND 1 FOR PLAYER 2
```

```
GET READY TO GUESS WORD!
```

```
3  
2  
1  
GO!
```

```
-----  
GUESSED WORD CORRECTLY!  
-----
```

```
Player 1 score for Round: 2  
Player 1 streak: 1  
PLAYER 1 TOTAL SCORE FOR ROUND 1: 2  
-----
```

```
Player 2 score for Round: 1  
Player 2 streak: 1  
PLAYER 2 TOTAL SCORE FOR ROUND 1: 1  
-----
```

```
*****
```

First, player 1 has go

Then, player 2 has go

Then, score summary,

Then next round,
and same cycle happens

```
*****
```

```
ROUND 2 FOR PLAYER 1
```

```
GET READY TO GUESS WORD!
```

```
3  
2  
1  
GO!
```

Walkthrough of game - end of game

GUESSED WORD CORRECTLY!

Player 1 score for Round: 32

Player 1 streak: 2

PLAYER 1 TOTAL SCORE FOR ROUND 10: 466

Player 2 score for Round: 150

Player 2 streak: 10

PLAYER 2 TOTAL SCORE FOR ROUND 10: 874

OVERALL SCORE FOR PLAYER 1: 466

OVERALL SCORE FOR PLAYER 2: 874


PLAYER 2 WINS!

Player with highest score
is winner

If both players have
same score, there is tie

Scoring system

Failed round,
No score,
Streak 0



Player 1 score for Round: 0
Player 1 streak: 0

PLAYER 1 TOTAL SCORE FOR ROUND 8: 434

Player 2 score for Round: 128
Player 2 streak: 8

PLAYER 2 TOTAL SCORE FOR ROUND 8: 643

Scoring system explanation:

1. Score for a round is firstly determined by time to guess the entire word, using the formula below (note, the score is rounded to the nearest integer):

$\text{score for round} = 100 / \text{seconds}$



2. Another factor in scoring is 'streaks'. Streaks occur when a player has successfully guessed words consecutive times. For the 1st round, if the player gets a word correct, they get a streak of 1, then if they get the next word, they're on a streak of 2, then 3 etc. until they fail to guess a word. Then the streak goes back to 0, and they have to start building up the streak again. This streak is then accounted for in the final formula:

$\text{score for round} = \text{streak} * (100 / \text{seconds})$



Total score so far

Score = $8 \times (100 / 6.25 \text{ seconds})$

Score = 8×16

Score = 128

Scoring system (streaks demo)

GUESSED WORD CORRECTLY!

Score for Round: 45

Streak: 5

TOTAL SCORE FOR ROUND 5: 277

When failure, streak goes to 0, no matter what streak was before

Even if next round is successful, because streak needs to start again, score is dramatically reduced

FAILURE!

Correct word was: NORTHEAST

Score for Round: 0

Streak: 0

TOTAL SCORE FOR ROUND 6: 277

GUESSED WORD CORRECTLY!

Score for Round: 12

Streak: 1

TOTAL SCORE FOR ROUND 7: 289

Other functionality

```
Enter '1' for singleplayer, enter '2' for multiplayer: 6

Not a valid number.
Press '3' to quit.
Enter '1' for singleplayer, enter '2' for multiplayer: blah

Not a whole number.
Press '3' to quit.
Enter '1' for singleplayer, enter '2' for multiplayer: █
```

```
Guess a letter:

That's not a valid guess.
Guesses remaining: 10
Press '3' to quit.
Guess a letter: el

That's not a valid guess.
Guesses remaining: 10
Press '3' to quit.
Guess a letter: ]

That's not a valid guess.
```

Program handling
bad responses

Pressing '3' quits from
anywhere

Also asks for input again if put in
same letter

```
Guess a letter: e
That is a correct letter!
```

```
E__E
```

```
Incorrect guesses: []
Guesses remaining: 10
Press '3' to quit.
```

```
Guess a letter: e
You have already guessed that letter.
```

```
Guess a letter: e

You have already guessed that letter.
Guesses remaining: 10
Press '3' to quit.
Guess a letter: 3
Ended program early.

Thanks for playing! Bye bye!
```

Live game demonstration...

Program logic - creating hangman classes

- Create hangman class, where each instance will have 6 lines of characters to print out picture of hangman
- Create initial hangman instance, which is final form of hangman (game-over hangman)
- Create every other instance based on previous instance, by changing small part of picture each time

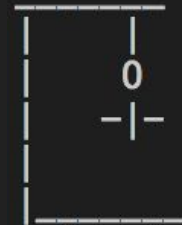
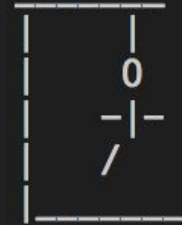
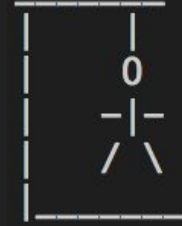
Examples:

```
stages = stage_builder(stages_initialiser())
```

```
stages[0].printer()
```

```
stages[1].printer()
```

```
stages[2].printer()
```



Program logic - creating grand list

- Read text file, list of 10,000 English (safe) words
- Put words into 2D list, where each list in 'grand_list' has words of certain length

Examples:

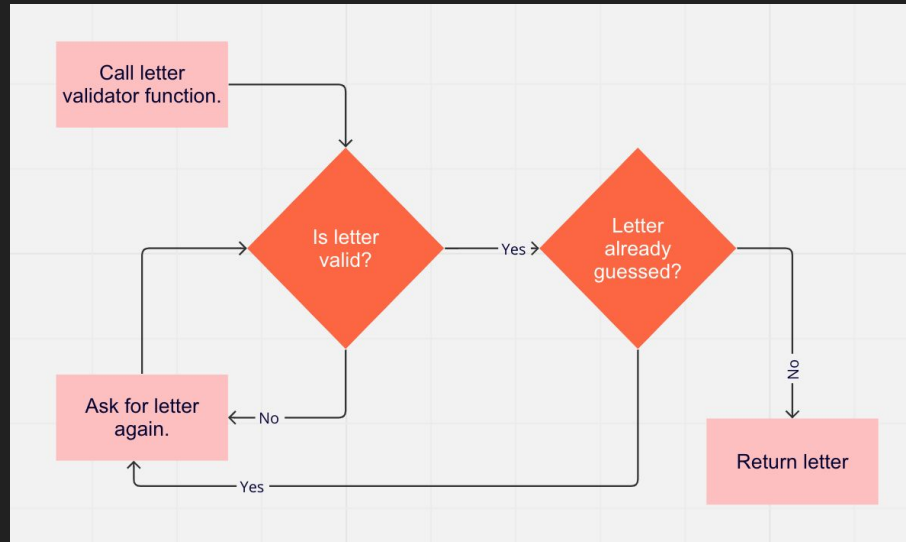
```
grand_list = grand_list_builder(wordlist_opener())  
  
print(grand_list[4 -1][:10])  
print("\n")  
print(grand_list[10 -1][:10])
```

```
['that', 'this', 'with', 'from', 'your', 'have', 'more', 'will', 'home', 'page']
```

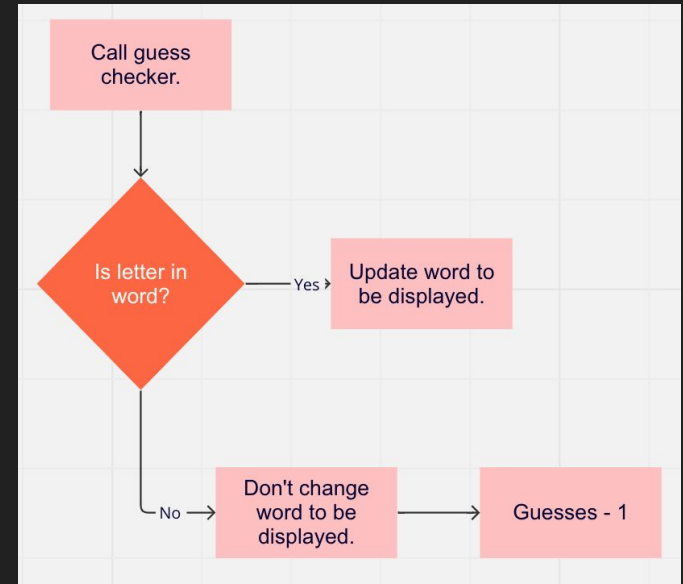
```
['university', 'management', 'technology', 'government', 'department', 'categories', 'conditions', 'experience', 'activities', 'additional']
```

Program logic - single round of hangman, check letter

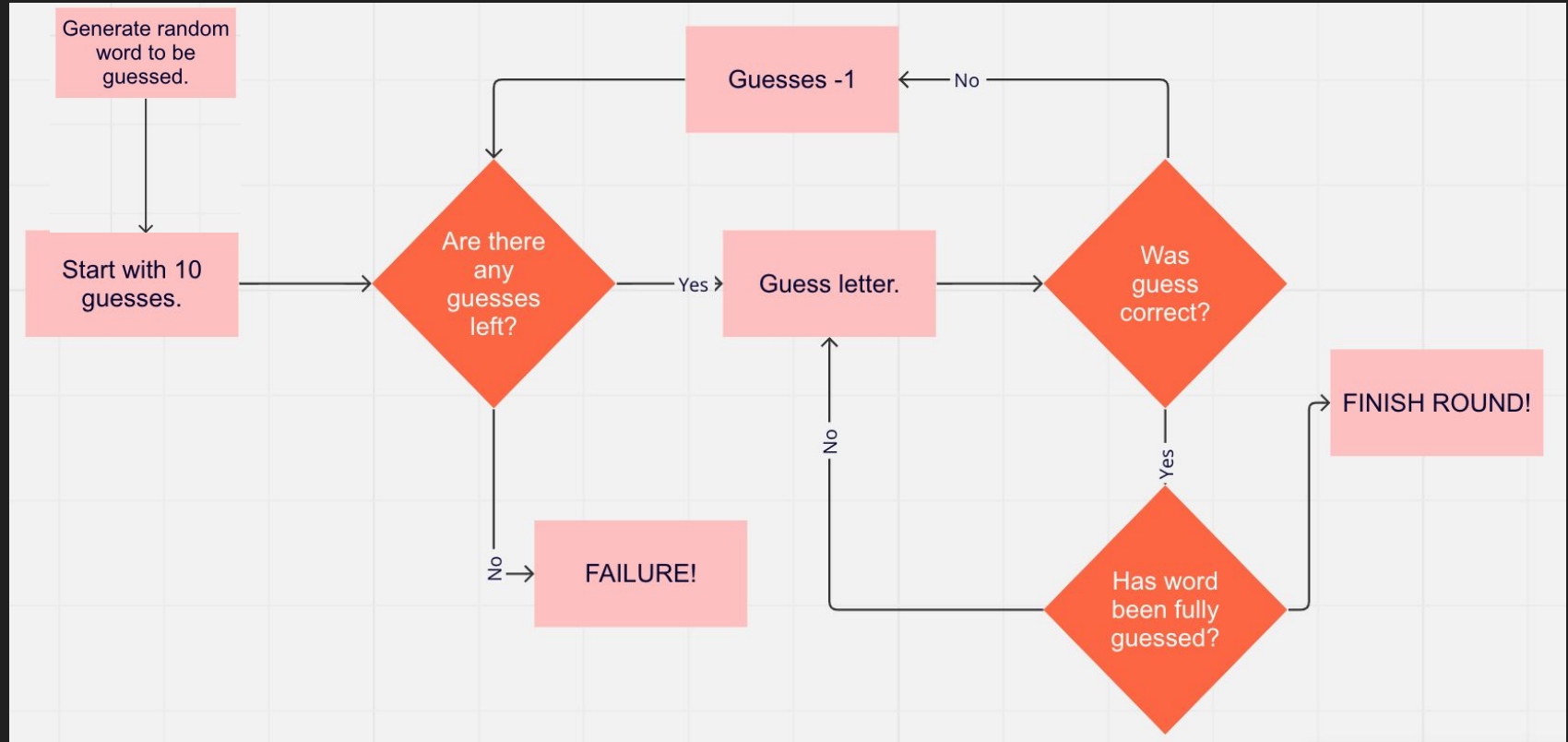
First get letter, check if input is valid



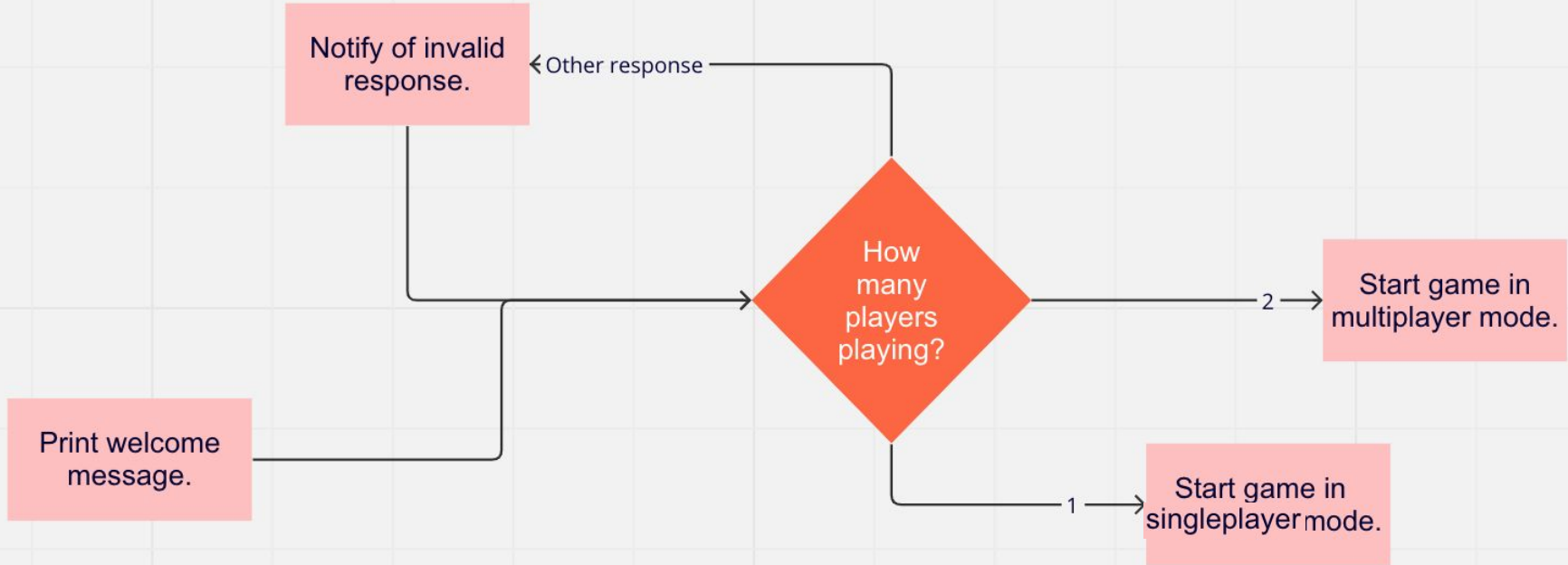
Then check if letter in word, make necessary updates



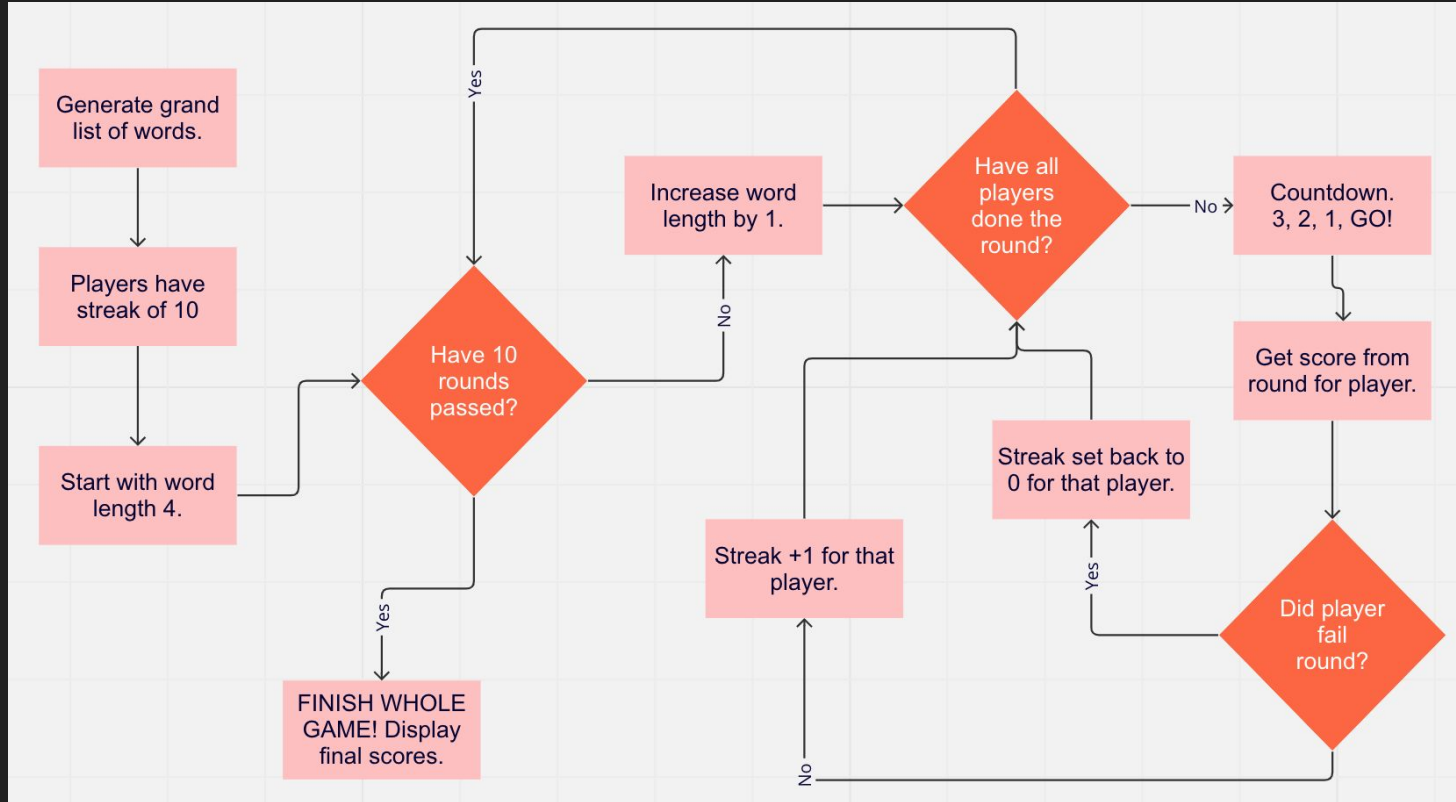
Program logic - entire hangman round



Program logic - game start



Program logic - main game



Challenges and ethical issues

- Finding a good word list, that had clean words that were also easy to guess
- Learning bash syntax, after becoming so familiar with Python
- Figuring out how to put in automatic tests, due to the fact the program is very input-oriented and contains rng too
- Keeping the presentation short!

T1A3

Thanks for watching!