Hangman Game

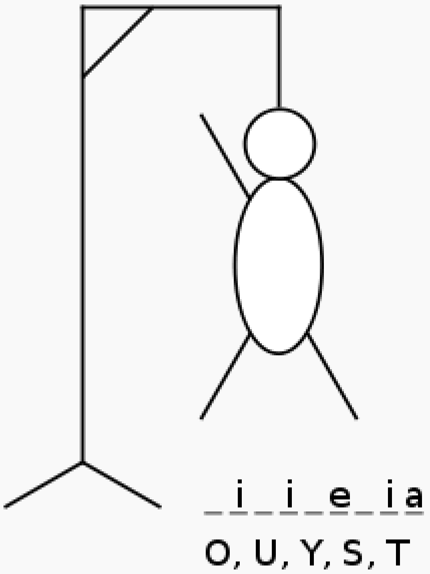
In this exercise, you'll implement the game logic of hangman and two user interfaces that reuse the same game logic.

Rules of the Game

Hangman is a two player game where one player selects a word and the other player attempts to guess the word. In our version of the game, the computer plays the part of the player that selects the word and tells the other player feedback about the guesses.

The word to guess is represented by a row of dashes, giving the number of letters, numbers and category. If the guessing player suggests a letter or number which occurs in the word, the other player writes it in all its correct positions. If the suggested letter or number does not occur in the word, the other player draws one element of the hanged man stick figure as a tally mark. The game is over when:

- The guessing player completes the word, or guesses the whole word correctly
- The other player completes the diagram



An example game in progress; the answer is Wikipedia.

Body Parts

The stick figure consists of the following body parts: head, body, left leg, right leg, left arm, right arm, left foot, right foot, left hand, right hand. This gives the player ten guesses before losing.

User Interfaces

Our customer wants two kinds of user interfaces for the game. The first is a “teletype” interface that prints out the entire state of the game after each of the player’s guesses in a streaming printout style. The second interface uses cursor addressing on the screen (such as curses or ncurses) to update the game state on the screen without redrawing the whole thing.

Exercise

Implement the game logic in a way that collaborates with the user interface and can be reused with either user interface.

Implement the user interfaces that collaborate with the game logic.

Remember that classes are cheap, so don’t be afraid to create classes that encapsulate single concepts in your design!

You may find it helpful to create a “driver” program that integrates your components together and allows you to manually test the gameplay. This can serve as an “acceptance test” of your game and ensure that the pieces you have created in isolation work well together.

You can use any character representation you like for the body parts. Our customer tells us that he’s going to hire an ASCII artist for the final version anyway.

Here is a representative example of the “teletype” style interface. Your interface needn’t display things exactly as this example shows, it is merely a guide.

SAMPLE RUN

```
HERE ARE THE LETTERS YOU USED:


--------

WHAT IS YOUR GUESS? E


SORRY, THAT LETTER ISN'T IN THE WORD.
FIRST, WE DRAW A HEAD
XXXXXXX
X     X
X    ---
X   (. .)
X    ---
X
X
X
X
X
X
X


HERE ARE THE LETTERS YOU USED:
E

--------

WHAT IS YOUR GUESS? A

-A---A--

WHAT IS YOUR GUESS FOR THE WORD?
WRONG.  TRY ANOTHER LETTER.

HERE ARE THE LETTERS YOU USED:
E,A

-A---A--

WHAT IS YOUR GUESS? R


SORRY, THAT LETTER ISN'T IN THE WORD.
NOW WE DRAW A BODY
XXXXXXX
X     X
X    ---
X   (. .)
X    ---
X     X
X     X
X     X
X     X
X
X
X


HERE ARE THE LETTERS YOU USED:
E,A,R

-A---A--

WHAT IS YOUR GUESS? O

-A-O-A--

WHAT IS YOUR GUESS FOR THE WORD?
WRONG.  TRY ANOTHER LETTER.

HERE ARE THE LETTERS YOU USED:
E,A,R,O

-A-O-A--

WHAT IS YOUR GUESS? T


SORRY, THAT LETTER ISN'T IN THE WORD.
NEXT WE DRAW AN ARM
XXXXXXX
X     X
X    ---
X \ (. .)
X  \ ---
X   \ X
X    \X
X     X
X     X
X
X
X
```