Game: Hacking Game

Version:V2

Description

When the game starts, a 800 pixels wide by 800 pixels high window and with the title , instructions are displayed, explaining how the game works. Words are then displayed in ALL CAPS. Each are on separate lines. The user is then prompted for their guess at the

password, **number of guesses remaining displayed.** If the word is correct, a congratulation message is displayed. **If the number of guesses now equals 0 and** the guess is incorrect, there is a condolence message displayed. If the user still has more guesses, the program checks to see if any letters are correct, and in the correct location, and displays the number of correct letters. **The program then prompts the user to try again, until the user is out of guesses.** When the game is over, a prompt for the enter key to end the game is displayed in the bottom left corner. When pressed, the window closes and the game ends. (All other writing starts in the top left, and goes down. No words are erased, and neither are the previous guesses. All text is 24pt font. White text on a black background.)

Functional tests:

1. Start the program
   1. Does window open with the right dimensions?
   2. Are the instructions displayed, followed by the words?
   3. Is it in the top left corner?
   4. Is the text white with a black background?
   5. Is there a prompt asking the user for their guess?(With correct font size and colour)
   6. Does the game wait until the enter key is pressed?
2. Press the enter key after inputting their guess
   1. Does the program tell the user if they are correct in their guess or not?(With correct font size and colour)
   2. Is there a prompt for the enter key to end the game?(With correct font size and colour)
3. Press enter
   1. Does the game end?
   2. Does the window close?

Algorithm

main program

create window (B)

**Display instructions(B)**

**Display words(B)**

Prompt for input(L)

**Check answer(B)**

**Display result(B)**

Prompt for enter(L)

end game (B)

block create window

initialize graphics library (L)

open window with title 'Window title', with size 500 by 400 (L)

Fill background with black(L)

block Display instructions

**Create an instruction list**

**Get string height**

**Compute initial location**

**For each instruction in instruction list**

**Display the instruction**

**Increment the y position by the string height**

Block Display words

**Create an word list**

**Get string height**

**Compute initial location**

**For each word in word list**

**Display the word**

**Increment the y position by the string height**

Block Check answer

**Player is incorrect.**

**while the player has more guesses**

**If the answer is correct**

**Player is right**

**No more guesses, because they’ve won**

**Else**

**Prompt for new guess(L)**

**Decrease the number of guesses**

Block display result

**If player is correct**

**Display congratulations(L)**

**Else**

**Display condolence(L)**

block end game

close window (L)

import uaio

import pygame

def main():

# create window

pygame.init()

surfaceX=800

surfaceY=800

surface = pygame.display.set\_mode((surfaceX, surfaceY), 0, 0)

pygame.display.set\_caption('Hacker!!!')

# display instructions

textX=0;

textY=0;

instructions = ['A group of possible passwords will be displayed.','You must guess the password. You have at most 4 guesses.','If you are incorrect you will be told how many letters in','your guess were exactly the correct location of the password.']

height=uaio.get\_height('Test.')

for instruction in instructions:

uaio.draw\_string(instruction, surface, (textX, textY))

textY = textY + height

# display words

words = ['PROVIDE', 'SETTING', 'CANTINA', 'CUTTING', 'HUNTERS', 'SURVIVE', 'HEARING', 'HUNTING', 'REALIZE', 'NOTHING', 'OVERLAP', 'FINDING', 'PUTTING']

answer = 'SURVIVE'

for word in words:

uaio.draw\_string(word, surface, (textX, textY))

textY = textY + height

#check answer

isPlayerCorrect = False

numberGuesses = 4

guess = uaio.input\_string('Enter Password (' + str(numberGuesses) + ' guesses remaining)>', surface, (textX, textY))

textY = textY + height

while numberGuesses > 0:

if guess == answer:

isPlayerCorrect = True

numberGuesses = 0

else:

numberGuesses = numberGuesses - 1

if numberGuesses > 0:

uaio.draw\_string('Password Incorrect', surface, (textX, textY))

textY = textY + height

guess = uaio.input\_string('Enter Password (' + str(numberGuesses) + ' guesses remaining)>', surface, (textX, textY))

textY = textY + height

#display result

if isPlayerCorrect:

uaio.draw\_string('User login successful', surface, (textX, textY))

else:

uaio.draw\_string('User login unsuccessful', surface, (textX, textY))

textY = surfaceY - height

uaio.input\_string('Press enter to exit', surface, (textX, textY))

# end game

pygame.quit()

main()