PYTHON PROJECT

HANGMAN 2.0

SPECIFICATIONS OF THE GAME:

Hangman2.0 demands the following specifications:
>>mp3 compatible system.

>>keyboard and mouse

>>python 2.5 and above required.

>>Pygame installed in python.

>>Good word power(Optional).

FEATURES OF THE GAME:

Hangman 2.0 has the following features:

Tab feature to continue to the next screen in the game.

The word to be guessed is not displayed on the screen.(word given by player1) but topic is displayed.

Back space feature ,Quit feature

Pictorial display analogus to losing chance in the game (player2 guessing a wrong letter)

Button interactions.

Displaying blanks equal to the number of letters in the word but and filling up the blanks corresponding to matching inputs.Displaying the letters guessed by player2.

Preventing loss of chance to player2 due to guessing the letter aldready guessed before.

THE GAME:

Hangman 2.0's gaming part :

Tab feature(Continuining to next screen/page) is used by closing the current screen and opening(displaying) a new one without any time gap.

Backspace feature has been duplicated by replacing the word with the same word excluding the last letter and blacking the former one.

Button interactions and keyboard interactions have been provided using pygame library which tracks the mouse clicks and key typings and action matching the event is performed.

Two functions printdashes_ and input_Guess have been used to create the hangman2.0's playable part.

In order to use these functions we first take input word from player 1 and store the
the word in a list containing the letters of the word and also create another list with the unique elements of the first list.The lengths of these lists are also stored.

Printdashes_ ( ) is used to print dashes on screen corresponding to the number of letters in the given word.
If the no.of letters is less than 13 all are printed in one line. Else the remaining blanks are printed in the next line.The x and y coordinates of these dashes are stored in two seperated lists.

Printdashes_( ) first checks if the length of the list(word list) is less than 13 or no.If the length is less than 13 we print the dashes with spaces corresponding to number of letters in the word.If length greater than 13 we find the remainder(r) and quotient(q) of division of length of the word by 13.We print (r) lines of 13 dashes and q dashes in the last line.We store the locations of the coordinates of the dashes in lists.

input_Guess() is used to show whether the letter guessed by player2 is correct or not .
If player2 guesse the correct letter it is dispalyed at the corresponding blank by using the coordinates of the dashes we stored initially.If a wrong letter is guessed then a picture is displayed indicating loss of chance. If player 2 fails to guess in five chances player1 is declared the winner else player2 is declared the winner.

i is incremented every time a letter is guessed.We do this to print the all the guessed letters on the screen.(location of the display of letters guessed is given as (400+20*i))

In this function we initialize two integers i,j to zero and three empty lists.Player2 is given five chances to guess.So we do this by using a while loop with j<5 condition.If player guesses a wrong letter then j is incremented by 1 and the loop starts over again.When the player guesses a letter if it is there in the word then we display it on screen using the location of the corresponding blank from above stored lists.Also the guessed letters are stored in new seperate lists(correct ones ,wrong ones).We check for the letters using another while loop which goes through the list in which letters of the given worrd are there.If letter is not found then we display a picture and increment j by 1.We check for win or loss by seeing if j becomes 5 and the new list where the correct letters guessed if not equal to the list containing the letters of the word then we say player1 wins if j<5 and len of list containing correct guessed letters equal to len of list containing the letters of the word we sasy player2 is the winner.

Also we prevent loss of chance because of guessing the same wrong letter twice by storing the unique lettersw guessed in a list and working with it.

DESIGN OF THE GAME:

Hangman 2.0 has the following design: It is a 2 player game.

The code for the entire program has been divided into 7 parts namely:

1)Play Hangman
2)Hangman1
3)Loss
4)Victory
5)Topic
6)About
7)WarningMessage

The pictures being used have been placed in a folder with the name "IMAGES"

The sounds being used in the game have been placed in a folder with the name "SOUNDS"
TAB key is pressed to continue to next screen(or also call next function.)

The game begins with the execution of Play Hangman .py python file.
This file contains three functions – message1,message2 and message3
We begin by calling the function message1
The function message1 displays the initial screen with the title of the game on it.
Continuining to the next function message2 it displays simple rules regarding the game.Then the third function message3 is defined and called which displays the prizes to the winner and loser .The previous screens are closed after the players decide to continue forward and new screens are displayed.After calling the last function in Play Hangman we continue forward by importing WarningMessage

WarningMessage has a function WarningM( ) which opens a small window with a warning message and two buttons on it. It displays a warning and asks the players to agree to the stated conditions.Mouse click interaction is provided to the two buttons displayed.One button has 'yes' on it and the other has 'no' on it.Players can proceed further only if they click on 'yes' button.

We track the position of the mouse pointer click for enabling mouse interaction to the buttons.When yes button is clicked we import Hangman1 python file which contains the main game.

In Hangman1 we take a word input from the first player which is not displayed on the screen to player2 .We do this by tracking the keys pressed on the keyboard.

Then we begin the game.We ask player1 to specify the topic the word is related to and move onto the guessing screen where the player2 makes guesses.Blamks equal to the no.of letters in the given word are shown on the screen at the rigth .Player2 has five chances to crak the word using the clue.The letters guessed are displayed on the lower part of the screen . If a correct letter is guessed then it is displayed on the corresponding blank and chance is retained.If a wrong letter is guessed then the player loses a chance and a picture analogus to losing a chance is displayed on a white canvas in the left of the screen.

The word given is displayed at ens of the play.

If player2 wins we import Loss and if player1 wins we inport Victory python files.
Loss and Python .py files have functions correspondingly which display the winners and also provide a button to quit the game.

If the button'quit' is pressed the About is imported which has a function AboutM( ) which displays the info about game development .This screen is shown for 5 sec and the n the game is closed.