

HANGMAN GAME

(IN PYCHARM)

Introduction:

Hangman game is game based on Artificial intelligence which is used by the machine. Once we feed the information into the database then it's up to the machine to use the artificial intelligence and ask the questions by random choices.

Basically, Hangman game is the beginners level game to Sharpe the kids mind by asking quick and unique questions and to challenge them to solve these questions in limited time or maximum SEVEN tries otherwise u have failed to proceed to the next level.

Description:

In this section we described the game details and other important features of our project.

Levels:

There are total five levels of our game which are as:

- 1) FILL IN THE BLANKS.
- 2) GUESSING OBJECTS.
- 3) MATHEMATICAL QUESTIONS.
- 4) GUESSING PLACES.
- 5) MAZE

Age Restrictions:

There are no age restrictions for this game but it is preferred for the kids age between 4 to 16 years. So that their mind would become more sharper and stronger.

Game Details:

This game is consisting of Five basic levels.

First level is comprised of random choice words which is asked by the computer in terms of “Fill in the Blanks”. To fill up the blank every kid gets SEVEN TRIES to complete the word, in mean time if any kid failed to fill the blanks in first six tries then beside WORD there is a statue which is slowly going to the death mode and at last on SEVEN TRY if they pass then the game proceed to the next level otherwise statue will be hang up to the rope.

Second level is comprised of guessing the objects by adjusting their answer with given shapes to find out the correct answer. It is just an analytical reasoning type question. A phrase is given in term of question and a blank is also leave and some diagrams are shown in the choices and you have to select your answer from the diagrams as one of the choices. The selected right diagram has 80 to 90 % resemblance to the right answer.

Third level is comprised of mathematical operations. These are not as much tougher as the Math’s questions but simply comprise of ADD questions and some equation type questions. You have to be just careful or if you can know how to use the simple calculator. By using these gadgets, you may easily pass this level.

- Remember you have only Seven Chances to pass the first level, from there onwards any single mistake will make you out of the game, so in these remaining four levels you have only 1 chance to pass the level to remain alive in the game. In the remaining four levels we have decreased the choices to the minimum level because in every question there are four choices so that we guess there is no need to gave them any extra choices.

Fourth level is comprised of guessing the right place from the given options. In level 4, some important places pictures are shown to you and you have to pick the right one. You will only be succeeded if you have any interest in world geography. And I guess this level is one of the interesting levels that we have in this game.

Fifth level is just a MAZE. A maze map is shown in the last level and you have to pass the maze by choosing the right path. If you touch the lines or boundaries a warning is issued just as a life line and in the third try you will be considered as out of the game, so in case you have to pass the maze in less than three tries. Once you pass the maze your game is completed, you have passed all the levels successfully. And your game will be END.

Concepts:

Some concepts which are used in making of this game are:

Pygame graphics.

Linear searching.

Code encryption/decryption.

Loops.

Conditional statements.

Functions.

Random choices.

Time complexity.

Screen Shots of program.

Documentation of Project

```

1  import pygame,sys,random,time
2  import pygame.event as GAME_EVENT
3  import pygame.locals as GAME_LOCALS
4  pygame.init()
5  pygame.font.init()
6
7  #####
8
9  #window size
10 window_width = 1024
11 window_height = 720
12 font_w = window_width/2 - 250
13 font_h = window_height - 200
14 font_x = 300
15 font_y = 30
16 window = pygame.display.set_mode((window_width,window_height))
17 pygame.display.set_caption('Hangman')
18 start_image = pygame.image.load("start.jpg")
19 game_title = pygame.image.load("HANGMAN.PNG")
20 level_1 = pygame.image.load("1level.jpg")
21 level_2 = pygame.image.load("2level.jpg")
22 level_3 = pygame.image.load("3level.jpg")
23 level_4 = pygame.image.load("4level.jpg")
24 #level_5 = pygame.image.load("level5.jpg")
25 chance_1 = 0
26 count = 0 #to check whether word is complete or not
27 start = False
28 stop = False
29 sp = 50
30
31 #####
32
33 #####

```

```

34 st = ['Stage 1 || Press Space to Continue', 'stage 2 || Press Space to Continue',
35       'stage 3 || Press Space to Continue', 'stage 4 || Press Space to Continue',
36       'stage 5 || Press Space to Continue']
37 def stage(i):
38     global st
39     run = True
40     while run:
41         window.fill((255, 255, 255))
42         font = pygame.font.SysFont("arial", 40)
43         text = font.render(st[i], True, (0, 0, 0))
44         window.blit(text, (200, 300))
45         for event in GAME_EVENT.get():
46             if event.type == pygame.KEYDOWN:
47                 if event.key == pygame.K_SPACE:
48                     run = False
49             if event.type == GAME_LOCALS.QUIT:
50                 pygame.quit()
51                 sys.exit()
52         pygame.display.update()
53
54 #animated lines
55 def lines():
56     pygame.draw.lines(window, (0, 0, 0), False, [(150, 80), (150, 450)], 7)
57     pygame.draw.lines(window, (0, 0, 0), False, [(100, 450), (400, 450)], 10)
58     pygame.draw.lines(window, (0, 0, 0), False, [(150, 80), (300, 80)], 8)
59     pygame.draw.lines(window, (0, 0, 0), False, [(230, 80), (230, 150)], 6)
60
61 #####
62
63 word = ['algorithm', 'name', 'country', 'stream', 'mean',
64        'witch', 'dialogue', 'background', 'filter', 'products',
65        'aftershock', 'abolishment', 'copyright', 'computer', 'mobile']
66 txt = random.choice(word)
67 guess = []
68 for letters in txt:

```

Documentation of Project

```

69     guess += letters
70     for blanks in range(len(guess)):
71         guess += ['_']
72
73 def select_word():
74
75     global sp,txt,guess,count
76     l_txt = len(txt)
77     l_guess = len(guess)
78     for i in range(l_txt,l_guess):
79         font = pygame.font.SysFont("arial", 50)
80         text = font.render(guess[i], True, (0, 180,255))
81         window.blit(text, (font_w+sp,font_h))
82         sp += 50
83
84
85 #####
86
87 def restart_game():
88     font = pygame.font.SysFont("arial", 50)
89     text = font.render('GAME OVER. Press Space to restart...', True, (255, 0, 0))
90     window.blit(text, (window_width / 10, window_height / 2))
91
92 #####
93
94 def body(ch):
95     global chance_l,stop
96     if ch == 1:
97         pygame.draw.circle(window, (0,0,0), (230,170),30,0)
98     if ch == 2:
99         pygame.draw.lines(window, (0, 0, 0), False, [(230, 170), (230, 300)], 5)
100    if ch == 3:
101        pygame.draw.lines(window, (0, 0, 0), False, [(230, 220), (290, 250)], 5)
102    if ch == 4:
103        pygame.draw.lines(window, (0, 0, 0), False, [(230, 220), (170, 250)], 5)
104
105    pygame.draw.lines(window, (0, 0, 0), False, [(230, 300), (280, 350)], 5)
106    if ch == 6:
107        pygame.draw.lines(window, (0, 0, 0), False, [(230, 300), (180, 350)], 5)
108    if ch > 6:
109        stop = True
110        restart_game()
111
112 #####
113
114 def word_search(check1):
115     global chance_l, txt,guess,sp,count
116     status = False
117     num_0 = len(guess)
118     num = len(txt)
119     num_l = len(txt) # for print letter at right blank
120     sp = 50
121     for ltr_n in range(num):
122         if guess[ltr_n] == check1 and check1 not in guess[num:num_0]: #letters will not duplicate
123             status = True
124             guess[num_l] = check1
125             select_word()
126             count += 1
127             num_l += 1
128
129     if status == False and check1 not in guess[num:num_0]:
130         chance_l += 1
131         body(chance_l)
132
133 #####
134
135 def load_start_image():
136     run = True
137     while run:
138         window.blit(start_image, (0,0))

```

Documentation of Project

```

139     window.blit(game_title, (window_width/2 - 380, window_height/2 - 100))
140     font = pygame.font.SysFont("arial", 50)
141     text = font.render(' ', True, (255, 255, 255))
142     window.blit(text, (window_width / 10, window_height / 2))
143     for event in GAME_EVENT.get():
144         if event.type == GAME_LOCALS.QUIT:
145             pygame.quit()
146             sys.exit()
147         if event.type == pygame.KEYDOWN:
148             if event.key == pygame.K_SPACE:
149                 run = False
150             pygame.display.update()
151
152     #level 1 main loop
153     def level_1_fun():
154         global start, chance_1, stop, txt, guess
155         run = True
156         while run:
157             if start == False: # for game start
158                 window.blit(level_1, (0, 0))
159                 lines()
160                 select_word()
161                 start = True # game running
162             for event in GAME_EVENT.get():
163                 if event.type == pygame.KEYDOWN:
164                     if stop == False: # to restart the game if over
165                         if event.key == pygame.K_a:
166                             char = 'a'
167                             word_search(char)
168                         if event.key == pygame.K_b:
169                             char = 'b'
170                             word_search(char)
171                         if event.key == pygame.K_c:
172                             char = 'c'
173                             word_search(char)

```

```

174             if event.key == pygame.K_d:
175                 char = 'd'
176                 word_search(char)
177             if event.key == pygame.K_e:
178                 char = 'e'
179                 word_search(char)
180             if event.key == pygame.K_f:
181                 char = 'f'
182                 word_search(char)
183             if event.key == pygame.K_g:
184                 char = 'g'
185                 word_search(char)
186             if event.key == pygame.K_h:
187                 char = 'h'
188                 word_search(char)
189             if event.key == pygame.K_i:
190                 char = 'i'
191                 word_search(char)
192             if event.key == pygame.K_j:
193                 char = 'j'
194                 word_search(char)
195             if event.key == pygame.K_k:
196                 char = 'k'
197                 word_search(char)
198             if event.key == pygame.K_l:
199                 char = 'l'
200                 word_search(char)
201             if event.key == pygame.K_m:
202                 char = 'm'
203                 word_search(char)
204             if event.key == pygame.K_n:
205                 char = 'n'
206                 word_search(char)
207             if event.key == pygame.K_o:
208                 char = 'o'

```

Documentation of Project

```

211         char = 'p'
212         word_search(char)
213     if event.key == pygame.K_q:
214         char = 'q'
215         word_search(char)
216     if event.key == pygame.K_r:
217         char = 'r'
218         word_search(char)
219     if event.key == pygame.K_s:
220         char = 's'
221         word_search(char)
222     if event.key == pygame.K_t:
223         char = 't'
224         word_search(char)
225     if event.key == pygame.K_u:
226         char = 'u'
227         word_search(char)
228     if event.key == pygame.K_v:
229         char = 'v'
230         word_search(char)
231     if event.key == pygame.K_w:
232         char = 'w'
233         word_search(char)
234     if event.key == pygame.K_x:
235         char = 'x'
236         word_search(char)
237     if event.key == pygame.K_y:
238         char = 'y'
239         word_search(char)
240     if event.key == pygame.K_z:
241         char = 'z'
242         word_search(char)
243     if event.key == pygame.K_SPACE and chance_1 > 6: # letter won't print after game over
244         txt = random.choice(word)
245         guess = []

```

```

245         guess = []
246         for letters in txt:
247             guess += letters
248         for blanks in range(len(guess)):
249             guess += ['_']
250         stop = False
251         chance_1 = 0
252         start = False
253     if event.type == GAME_LOCALS.QUIT:
254         pygame.quit()
255         sys.exit()
256     if count == len(txt):
257         run = False
258
259     pygame.display.update()
260
261     #####
262     #####
263     #####
264
265

```