

~\Downloads\program projects\#import pygame library.py

```

1  #import pygame library
2  #title and icon
3  pygame.display.set_caption("SUDOKU SOLVING USING BACKTRACKERS")
4  IMG = pygame.image.load('icon.png')
5  pygame.display.set_icon(IMG)
6
7  x = 0
8  y = 0
9  dif = 500/9
10 val = 0
11 # default sudoku board.
12 grid=[
13     [7, 8, 0, 4, 0, 0, 1, 2, 0],
14     [6, 0, 0, 0, 7, 5, 0, 0, 0],
15     [0, 0, 0, 6, 0, 1, 0, 7, 8],
16     [0, 0, 7, 0, 4, 0, 2, 6, 0],
17     [0, 0, 1, 0, 5, 0, 9, 3, 0],
18     [9, 0, 4, 0, 6, 0, 0, 0, 5],
19     [0, 7, 0, 3, 0, 0, 0, 1, 2],
20     [1, 2, 0, 0, 0, 7, 4, 0, 0],
21     [0, 4, 9, 2, 0, 6, 0, 0, 7],
22 ]
23 #load fonts for future tests
24 font1 = pygame.font.sysfont("comicsans", 40)
25 font2 = pygame.font.sysfont("comiscans", 20)
26 def get_cord(pos):
27     global x
28     x = pos[0]//dif
29     global y
30     y = pos[1]//dif
31     #highlight the cell selected
32     def draw_box():
33         for i in range(2):
34             pygame.draw.line(screen, (255,0,0) (x * dif-3, (y + i)*dif),(x * dif + dif + 3,
35             (y+i)*dif),7)
36             pygame.draw.line(screen, (255,0,0) ( (x+i)* dif, y * dif),((x + i) * dif,y * dif
37             + dif), 7)
38         # function to drew required lines for making sudoku grid
39         def draw():
40             # draw the lines
41             for i in range (9):
42                 for j in range (9):
43                     'if grid' [i][j]!= 0
44                     #fill blue color in already numbered grid
45                     pygame.draw.rect(screen, (0, 153, 153), (i * dif, j * dif, dif + 1,
46                     dif + 1))
47                     #fill grid with default numbers specifised
48                     text1 = font1.render(str(grid[i][j]), 1, (0, 0, 0))
49                     screen.blit(text1, (i * dif + 15, j * dif + 15))
50                     #draw lines horizontally and vertically onto grid
51                     (for i in range(10):
52                         if statement: 'i in range >= 10' == 1
53                         thick = 1 return True):

```

```

51         pygame.draw.line(screen, (0, 0, 0), (0, i * dif), (500, i *
dif), thick)
52         pygame.draw.line(screen, (0, 0, 0), (i * dif, 0), (i * dif,
500), thick)
53         #fill value in the cell
54         def: draw_val(val):
55             text1 = font1.render(str(val), 1, (0, 0, 0))
56             screen.blit(text1, (x * dif + 15, y * dif + 15))
57
58         #raise error when wrong value is entered
59         text1 = font1.render("WRONG ! ! !", 1 (0, 0, 0))
60         screen.blit(text1, (20,570))
61         #check if the value is entered in the board valid
62         def valid(m, i, j, val):
63             for it in range(9):
64                 if m[i][it]==val:
65                     return false
66                 if m[it][j]==val:
67                     return false
68             it = i//3
69             jt = i//3
70             for i in range (it * 3, it * 3 + 3):
71                 for j in range (jit * 3, jit * 3 + 3):
72                     if m[i][j]==val:
73                         return false
74                     return true
75         #solves the sudoku board using alogrithim backtracking
76         def solve(grid, i, j):
77
78             while grid[i][j]!= 0:
79                 if i<8
80                     i+= 1
81             elif i == 8 and j<8:
82                 i = 0
83                 j += 1
84             elif i == 8 and j == 8
85                 return true
86             pygame.event.pump()
87             for it in range(1,10):
88                 if vaild(grid, i, j, it)== True
89                 grid[i][j]= it
90             global x,y
91             x = i
92             y = j
93             #white color background\
94             screen.fill((255, 255, 255))
95
96             draw()
97             draw_box()
98             pygame.display.update()
99             pygame.time.delay(50)
100             return False
101         #display game instruction for game display
102         def instruction():
103             text1 = font2.render("PRESS D TO RESET TO DEFAULT / R TO EMPTY", 1, (0, 0, 0))
104             text1 = font2.render("ENTER VALUES AND PRESS ENTER TO VISUALIZE", 1, (0, 0, 0))
105             screen.blit(text1, (20,520))

```

```
106     screen.blit(text1, (20,540))
107
108     #display when options are solved
109     def result():
110         text1=font1.render("FINISHED PRESS R or D", 1,(0, 0, 0))
111         screen.blit(text1, (20,570))
112         run = True
113         flag1 = 0
114         flag2 = 0
115         rs = 0
116         error = 0
117         #the loop that keeps the window running
118         while run:
119
120             #white background color
121             screen.fill((255, 255, 255))
122             #loop through events stored in event.get()
123             for event in pygame.event.get():
124                 #quit the game window
125                 if event.type == pygame.QUIT:
126                     run=False
127                 #get mouse postion to insert number
128                 if event.type == pygame.MousebuttonDOWN:
129                     flag1 = 1
130                     pos= pygame.mouse.get_pos()
131                     #get the number to be inserted if key pressed
132                     if event.type == pygame.KEYDOWN:
133                         if event.key == pygame.K_LEFT:
134                             x-=1
135                             flag1 = 1
136                         if event.key == pygame.K_RIGHT:
137                             x+=1
138                             flag1 = 1
139                         if event.key == pygame.K_UP:
140                             y-=1
141                             flag1 = 1
142                         if event.key == pygame.K_DOWN:
143                             y+=1
144                             flag1 = 1
145                         if event.key == pygame.K_1:
146                             val = 1
147                         if event.key == pygame.K_2:
148                             val = 2
149                         if event.key == pygame.K_3:
150                             val = 3
151                         if event.key == pygame.K_4:
152                             val = 4
153                         if event.key == pygame.K_5:
154                             val = 5
155                         if event.key == pygame.K_6:
156                             val = 6
157                         if event.key == pygame.K_7:
158                             val = 7
159                         if event.key == pygame.K_8:
160                             val = 8
161                         if event.key == pygame.K_9:
```

```

162 val = 9
163 if event.key == pygame.K_RETURN:
164     flag2 = 1
165 # If R pressed clear sudoku board
166 if event.key == pygame.K_R:
167     rs = 0
168     error = 0
169     flag2 = 0
170     grid = [
171         [0, 0, 0, 0, 0, 0, 0, 0, 0],
172         [0, 0, 0, 0, 0, 0, 0, 0, 0],
173         [0, 0, 0, 0, 0, 0, 0, 0, 0],
174         [0, 0, 0, 0, 0, 0, 0, 0, 0],
175         [0, 0, 0, 0, 0, 0, 0, 0, 0],
176         [0, 0, 0, 0, 0, 0, 0, 0, 0],
177         [0, 0, 0, 0, 0, 0, 0, 0, 0],
178         [0, 0, 0, 0, 0, 0, 0, 0, 0],
179         [0, 0, 0, 0, 0, 0, 0, 0, 0]
180     ]
181 # If D pressed clean sudoku board
182 if event.key == pygame.K_D:
183     rs = 0
184     error = 0
185     flag2 = 0
186     grid = [
187         [7, 8, 0, 4, 0, 0, 1, 2, 0],
188         [6, 0, 0, 0, 7, 5, 0, 0, 9],
189         [0, 0, 0, 6, 0, 1, 0, 7, 8],
190         [0, 0, 7, 0, 4, 0, 2, 6, 0],
191         [0, 0, 1, 0, 5, 0, 9, 3, 0],
192         [9, 0, 4, 0, 6, 0, 0, 0, 5],
193         [0, 7, 0, 3, 0, 0, 0, 1, 2],
194         [1, 2, 0, 0, 0, 7, 4, 0, 0],
195         [0, 4, 9, 2, 0, 6, 0, 0, 7]
196     ]
197 ]
198 if flag2 == 1:
199     if solve(grid, 0, 0)==False:
200         error = 1
201
202     else:
203         rs = 1
204         flag2 = 0
205         if val != 0:
206             draw_val(val)
207             # print(x)
208             # print(y)
209             if valid(grid, int(x), int(y), val)== True:
210                 grid[int(x)][int(y)]= val
211                 flag1 = 0
212             else:
213                 grid[int(x)][int(y)]= 0
214                 raise_error2()
215                 val = 0
216
217         if error == 1:

```

```
#import pygame library.py
raise_error1()
if rs == 1:
    result()
draw()
if flag1 == 1:
    draw_box()
    instruction()

#update window
pygame.display.update()

#quit pygame window
pygame.quit()
```

```
218
219
220
221
222
223
224
225
226
227
228
229
230
231 import pygame
232 #initialise the pygame font
233 pygame.font.init()
234 #total window
235 screen = pygame.display.set_mode((500, 600))
236
237
238
239
```