```
import pygame
pygame.font.init()
Window = pygame.display.set_mode((500, 500))
pygame.display.set_caption("SUDOKU GAME by DataFlair")
x = 0
z = 0
diff = 500 / 9
value= 0
defaultgrid =[
      [0, 0, 4, 0, 6, 0, 0, 0, 5],
      [7, 8, 0, 4, 0, 0, 0, 2, 0],
      [0, 0, 2, 6, 0, 1, 0, 7, 8],
      [6, 1, 0, 0, 7, 5, 0, 0, 9],
      [0, 0, 7, 5, 4, 0, 0, 6, 1],
      [0, 0, 1, 7, 5, 0, 9, 3, 0],
      [0, 7, 0, 3, 0, 0, 0, 1, 0],
      [0, 4, 0, 2, 0, 6, 0, 0, 7],
      [0, 2, 0, 0, 0, 7, 4, 0, 0],
   ]


font = pygame.font.SysFont("comicsans", 27)
font1 = pygame.font.SysFont("comicsans", 19)
def cord(pos):
   global x
   x = pos[0]//diff
   global z
   z = pos[1]//diff

def highlightbox():
   for k in range(2):
      pygame.draw.line(Window, (0, 0, 0), (x * diff-3, (z + k)*diff), (x * diff + diff + 3, (z + k)*diff), 7)
      pygame.draw.line(Window, (0, 0, 0), ( (x + k)* diff, z * diff ), ((x + k) * diff, z * diff + diff), 7)

def drawlines():
   for i in range (9):
      for j in range (9):
         if defaultgrid[i][j]!= 0:
            pygame.draw.rect(Window, (255, 255, 0), (i * diff, j * diff, diff + 1, diff + 1))
            text1 = font.render(str(defaultgrid[i][j]), 1, (0, 0, 0))
            Window.blit(text1, (i * diff + 15, j * diff + 15))
   for l in range(10):
      if l % 3 == 0 :
         thick = 7
      else:
         thick = 1
      pygame.draw.line(Window, (0, 0, 0), (0, l * diff), (500, l * diff), thick)
      pygame.draw.line(Window, (0, 0, 0), (l * diff, 0), (l * diff, 500), thick)


def fillvalue(value):
   text1 = font.render(str(value), 1, (0, 0, 0))
   Window.blit(text1, (x * diff + 15, z * diff + 15))


def raiseerror():
   text1 = font.render("wrong!", 1, (0, 0, 0))
   Window.blit(text1, (20, 570))
def raiseerror1():
   text1 = font.render("wrong ! enter a valid key for the game", 1, (0, 0, 0))
   Window.blit(text1, (20, 570))

def validvalue(m, k, l, value):
   for it in range(9):
      if m[k][it]== value:
```

```
            return False
        if m[it][l]== value:
            return False
    it = k//3
    jt = l//3
    for k in range(it * 3, it * 3 + 3):
        for l in range (jt * 3, jt * 3 + 3):
            if m[k][l]== value:
                return False
    return True
def solvegame(defaultgrid, i, j):

    while defaultgrid[i][j]!= 0:
        if i<8:
            i+= 1
        elif i == 8 and j<8:
            i = 0
            j+= 1
        elif i == 8 and j == 8:
            return True
    pygame.event.pump()
    for it in range(1, 10):
        if validvalue(defaultgrid, i, j, it)== True:
            defaultgrid[i][j]= it
            global x, z
            x = i
            z = j
            Window.fill((255, 255, 255))
            drawlines()
            highlightbox()
            pygame.display.update()
            pygame.time.delay(20)
            if solvegame(defaultgrid, i, j)== 1:
                return True
            else:
                defaultgrid[i][j]= 0
            Window.fill((0,0,0))

            drawlines()
            highlightbox()
            pygame.display.update()
            pygame.time.delay(50)
    return False
def gameresult():
    text1 = font.render("game finished", 1, (0, 0, 0))
    Window.blit(text1, (20, 570))
flag=True
flag1 = 0
flag2 = 0
rs = 0
error = 0
while flag:
    Window.fill((255,182,193))
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            flag = False
        if event.type == pygame.MOUSEBUTTONDOWN:
            flag1 = 1
            pos = pygame.mouse.get_pos()
            cord(pos)
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                x-= 1
                flag1 = 1
            if event.key == pygame.K_RIGHT:
```

```
            x+= 1
            flag1 = 1
        if event.key == pygame.K_UP:
            y-= 1
            flag1 = 1
        if event.key == pygame.K_DOWN:
            y+= 1
            flag1 = 1
        if event.key == pygame.K_1:
            value = 1
        if event.key == pygame.K_2:
            value = 2
        if event.key == pygame.K_3:
            value = 3
        if event.key == pygame.K_4:
            value = 4
        if event.key == pygame.K_5:
            value = 5
        if event.key == pygame.K_6:
            value = 6
        if event.key == pygame.K_7:
            value = 7
        if event.key == pygame.K_8:
            value = 8
        if event.key == pygame.K_9:
            value = 9
        if event.key == pygame.K_RETURN:
            flag2 = 1
        if event.key == pygame.K_r:
            rs = 0
            error = 0
            flag2 = 0
            defaultgrid=[
            [0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0],
            [0, 0, 0, 0, 0, 0, 0, 0, 0]
            ]
        if event.key == pygame.K_d:
            rs = 0
            error = 0
            flag2 = 0
            defaultgrid  =[
                [0, 0, 4, 0, 6, 0, 0, 0, 5],
                [7, 8, 0, 4, 0, 0, 0, 2, 0],
                [0, 0, 2, 6, 0, 1, 0, 7, 8],
                [6, 1, 0, 0, 7, 5, 0, 0, 9],
                [0, 0, 7, 5, 4, 0, 0, 6, 1],
                [0, 0, 1, 7, 5, 0, 9, 3, 0],
                [0, 7, 0, 3, 0, 0, 0, 1, 0],
                [0, 4, 0, 2, 0, 6, 0, 0, 7],
                [0, 2, 0, 0, 0, 7, 4, 0, 0],
            ]
    if flag2 == 1:
        if solvegame(defaultgrid , 0, 0)== False:
            error = 1
        else:
            rs = 1
        flag2 = 0
    if value != 0:
        fillvalue(value)
```

```
        if validvalue(defaultgrid , int(x), int(z), value)== True:
            defaultgrid[int(x)][int(z)]= value
            flag1 = 0
        else:
            defaultgrid[int(x)][int(z)]= 0
            raiseerror1()
        value = 0

    if error == 1:
        raiseerror()
    if rs == 1:
        gameresult()
    drawlines()
    if flag1 == 1:
        highlightbox()
    pygame.display.update()

pygame.quit()
```