```
# 15-112 Fundamentals of Programming and Computer Science: Final Project
# Name: Elizabeth Marella
# Andrew ID: emarella
# File Created: Nov 7, 2017
# Last Updated: Nov 26, 2017

# this code requres the pygame, random, and my own module, solutioncheck

# Sources:

#  Base code:
#   Pygame base template for opening a window
#   Sample Python/Pygame Programs
#   Simpson College Computer Science
#   http://programarcadegames.com/
#   http://simpson.edu/computer-science/
#   Explanation video: http://youtu.be/vRB_983kUMc
#  Button Functions: https://www.youtube.com/watch?v=kK4xhHr1QeQ
#  Background: http://www.wallpapereast.com/wallpaper-pattern/page/3

import pygame
from solutioncheck import *
import random

# Define some colors
black = (0, 0, 0)
gray = (150,150,150)
white= (255, 255, 255)
green = (0, 255, 0)
red = (255, 0, 0)

# initialize pygame
pygame.init()
pygame.display.init()

#set the background and size
defimg = pygame.image.load("menubackground.jpg")
defimg = pygame.transform.scale(defimg,(550,610))
rect = defimg.get_rect()

# Set the width and height of the screen and caption
size = (550, 610)
screen = pygame.display.set_mode(size)
pygame.display.set_caption("Sudoku")
```

```
# declare other variables
done = False
freeOps = False
timedOps = False
inst = False
scores = False
easy = False
medium = False
hard = False
slct = False
easyHigh = "0"
mediumHigh = "0"
hardHigh = "0"

# these functions draw the button boxes with a message
def text_objects(text,font):
    textSurface = font.render(text,True,black)
    return textSurface,textSurface.get_rect()

def button(msg,x,y,w,h,ic,ac):
    mouse = pygame.mouse.get_pos()
    if x + w > mouse[0] > x and y + h > mouse[1] > y:
        pygame.draw.rect(screen,ac,(x,y,w,h))
    else:
        pygame.draw.rect(screen,ic,(x,y,w,h))

    smallText = pygame.font.Font("freesansbold.ttf",25)
    textSurf, textRect = text_objects(msg, smallText)
    textRect.center = ((x + (w/2)),(y + (h/2)))
    screen.blit(textSurf,textRect)
    return msg


def options(cell,puzzle):
    # this function takes a tuple, consisting of row and column for the chosen cell, and a puzzle
    # goes through each row, column, and 3x3 subgrid that the cell is in and removes any digits
    # in those areas
    # returns a list of the remaining valid possibilities for a cell

    nums = ["1","2","3","4","5","6","7","8","9"]
    for i in range(9):

        # goes through the row
        if puzzle[cell[0]][i+1][1] in nums:
            nums.remove(puzzle[cell[0]][i+1][1])

        # goes through the column
        if puzzle[i+1][cell[1]][1] in nums:
            nums.remove(puzzle[i+1][cell[1]][1])

    quadRow = (cell[0]-1)/3
    quadCol = (cell[1]-1)/3

    # goes through the 3x3 grid
    for i in range(1,4):

        for j in range(1,4):

            if puzzle[i+3*quadRow][j+3*quadCol][1] in nums:
                nums.remove(puzzle[i+3*quadRow][j+3*quadCol][1])

    return nums

def fillBoard(puzzle):

    # fills a 9x9 board with a valid sudoku puzzle solution
    successful = False

    while successful == False:

        # randomly fills in the first row
        for i in range(9):

            dig = random.choice(options((1,i+1),puzzle))

            puzzle[1][i+1][1] = dig

        # randomly fills in the first column
        for i in range(9):

            if options((i+1,1),puzzle) != []:

                dig = random.choice(options((i+1,1),puzzle))

                puzzle[i+1][1][1] = dig
    # declare the list to store all the possibilites, but make it not empty so the next while loop begins running
    poss = [1]

    # check the number of possibilites for each empty cell and randomly fills in the cell with the lowest number of possibilities
    # until the grid has been filled, the function repeats this process until it finds a valid solution
    while poss != []:

      poss = []

      for i in range(9):

        for j in range(9):

          remaining = options((i+1,j+1),puzzle)

          if len(remaining) != 0:

            if puzzle[i+1][j+1][1] == "":

              poss.append([(i+1,j+1),len(remaining)])

      poss.sort(key=lambda x: x[1])

      if poss != []:

        dig = random.choice(options((poss[0][0][0],poss[0][0][1]),puzzle))

        puzzle[poss[0][0][0]][poss[0][0][1]][1] = dig

    # checks to see if valid
    if solCheck(puzzle,1):
      successful = True

    # resets the puzzle
    else:
      for i in range(9):
        for j in range(9):
          puzzle[i+1][j+1][1] = ""

  return puzzle
```

```
def setGame(puzzle,difficulty):
   # based on the difficulty level, this function removes a certain number of cells to create a
playing board

   # ranges for the different difficulty levels
   if difficulty == "easy":
      fill = [4,5,6]

   elif difficulty == "medium":
      fill = [3,4,5]

   elif difficulty == "hard":
      fill = [2,3,4]

   # go through each 3x3 grid and randomly select a certain number of boxes to display
   for i in range(3):

      for j in range(3):

         boxes = random.choice(fill)

         for b in range(boxes):

            row = random.randint(1,3)
            col = random.randint(1,3)

            puzzle[row+3*i][col+3*j][0] = puzzle[row+3*i][col+3*j][1]
            puzzle[row+3*i][col+3*j][2] = white

   return puzzle

def generatePuzzle(difficulty):
   # creates a ready to play sudoku puzzle

   # data for the puzzle is stored using an array, where the first element for each box is the
displayed element
   # the second element is a possible solution, and the final element indicates whether or not
the box can be
   # affected by user input or not
   puzzle = [ 1, [ 1, ["","",gray],["","",gray],["","",gray],["","",gray],["","",gray],
          ["","",gray],["","",gray],["","",gray],["","",gray] ],
        [ 2, ["","",gray],["","",gray],["","",gray],["","",gray],["","",gray],
          ["","",gray],["","",gray],["","",gray],["","",gray] ],
        [ 3, ["","",gray],["","",gray],["","",gray],["","",gray],["","",gray],
          ["","",gray],["","",gray],["","",gray],["","",gray] ],
        [ 4, ["","",gray],["","",gray],["","",gray],["","",gray],["","",gray],
          ["","",gray],["","",gray],["","",gray],["","",gray] ],
        [ 5, ["","",gray],["","",gray],["","",gray],["","",gray],["","",gray],
          ["","",gray],["","",gray],["","",gray],["","",gray] ],
        [ 6, ["","",gray],["","",gray],["","",gray],["","",gray],["","",gray],
          ["","",gray],["","",gray],["","",gray],["","",gray] ],
        [ 7, ["","",gray],["","",gray],["","",gray],["","",gray],["","",gray],
          ["","",gray],["","",gray],["","",gray],["","",gray] ],
        [ 8, ["","",gray],["","",gray],["","",gray],["","",gray],["","",gray],
          ["","",gray],["","",gray],["","",gray],["","",gray] ],
        [ 9, ["","",gray],["","",gray],["","",gray],["","",gray],["","",gray],
          ["","",gray],["","",gray],["","",gray],["","",gray] ] ]

   solution = fillBoard(puzzle)
   gameBoard = setGame(solution,difficulty)

   return gameBoard

# Used to manage how fast the screen updates
clock = pygame.time.Clock()
clock1 = pygame.time.Clock()
clock2 = pygame.time.Clock()
clock3 = pygame.time.Clock()

# -------- Main Program Loop -----------
while not done:
   # --- Main event loop
   for event in pygame.event.get():
      if event.type == pygame.QUIT:
         done = True
      mouse = pygame.mouse.get_pos()
      click = pygame.mouse.get_pressed()

      # when the exit button is pressed
      if 200+150 > mouse[0] > 200 and 300+50 > mouse[1] > 300 and click[0] == 1:
         done = True

      # when the free play button is pressed
      if 100+150 > mouse[0] > 100 and 160+50 > mouse[1] > 160 and click[0] == 1:
         freeOps = True
         while freeOps:
            for event in pygame.event.get():
```

```
               if event.type == pygame.QUIT:
                  freeOps = False
                  done = True
               mouse = pygame.mouse.get_pos()
               click = pygame.mouse.get_pressed()

               # when back key is pressed
               if 200+150 > mouse[0] > 200 and 370+50 > mouse[1] > 370 and click[0] == 1:
                  freeOps = False

               # when easy button is pressed
               if 200+150 > mouse[0] > 200 and 160+50 > mouse[1] > 160 and click[0] == 1:
                  easy = True

                  puzzle = generatePuzzle("easy")

                  while easy:
                     for event in pygame.event.get():
                        if event.type == pygame.QUIT:
                           easy = False
                           freeOps = False
                           done = True
                        mouse = pygame.mouse.get_pos()
                        x = mouse[0]
                        y = mouse[1]
                        click = pygame.mouse.get_pressed()
                        one = pygame.key.get_pressed()[pygame.K_1]
                        two = pygame.key.get_pressed()[pygame.K_2]
                        three = pygame.key.get_pressed()[pygame.K_3]
                        four = pygame.key.get_pressed()[pygame.K_4]
                        five = pygame.key.get_pressed()[pygame.K_5]
                        six = pygame.key.get_pressed()[pygame.K_6]
                        seven = pygame.key.get_pressed()[pygame.K_7]
                        eight = pygame.key.get_pressed()[pygame.K_8]
                        nine = pygame.key.get_pressed()[pygame.K_9]
                        delete = pygame.key.get_pressed()[pygame.K_BACKSPACE]
                        hint = pygame.key.get_pressed()[pygame.K_h]

                        # when back key is pressed
                        if 90+150 > x > 90 and 550+50 > y > 550 and click[0] == 1:
                           easy = False
                        # when play again button is pressed
                        if 310+150 > x > 310 and 550+50 > y > 550 and click[0] == 1 and
solCheck(puzzle):
```

```
                    easy = False

                # if a number key, backspace, or h is pressed then take the appropriate action
for the
                # box selected
                if one == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                    puzzle[(y-50)/55+1][(x-30)/55+1][0] = "1"

                elif two == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                    puzzle[(y-50)/55+1][(x-30)/55+1][0] = "2"

                elif three == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                    puzzle[(y-50)/55+1][(x-30)/55+1][0] = "3"

                elif four == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                    puzzle[(y-50)/55+1][(x-30)/55+1][0] = "4"

                elif five == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                    puzzle[(y-50)/55+1][(x-30)/55+1][0] = "5"

                elif six == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                    puzzle[(y-50)/55+1][(x-30)/55+1][0] = "6"

                elif seven == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                    puzzle[(y-50)/55+1][(x-30)/55+1][0] = "7"

                elif eight == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                    puzzle[(y-50)/55+1][(x-30)/55+1][0] = "8"

                elif nine == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                    puzzle[(y-50)/55+1][(x-30)/55+1][0] = "9"

                elif delete == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                    puzzle[(y-50)/55+1][(x-30)/55+1][0] = ""

                elif hint == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                    puzzle[(y-50)/55+1][(x-30)/55+1][0] = puzzle[(y-50)/55+1][(x-30)/55+1][1]

            screen.blit(defimg,rect)

            # display these if the game has been won
            midText = pygame.font.Font("freesansbold.ttf",20)
            if solCheck(puzzle):
                textSurf, textRect = text_objects("Congratulations! You won!", midText)
```

```
    textRect.center = (size[0]/2,25)
    screen.blit(textSurf, textRect)
    button("Play Again",310,550,150,50,white,gray)

# display these if the game is still continuing
else:
    textSurf, textRect = text_objects("Easy", midText)
    textRect.center = (size[0]/2,25)
    screen.blit(textSurf, textRect)

# display 9x9 grid
# ROW 1
button(puzzle[1][1][0],30,50,50,50,white,puzzle[1][1][2])
button(puzzle[1][2][0],85,50,50,50,white,puzzle[1][2][2])
button(puzzle[1][3][0],140,50,50,50,white,puzzle[1][3][2])
button(puzzle[1][4][0],195,50,50,50,white,puzzle[1][4][2])
button(puzzle[1][5][0],250,50,50,50,white,puzzle[1][5][2])
button(puzzle[1][6][0],305,50,50,50,white,puzzle[1][6][2])
button(puzzle[1][7][0],360,50,50,50,white,puzzle[1][7][2])
button(puzzle[1][8][0],415,50,50,50,white,puzzle[1][8][2])
button(puzzle[1][9][0],470,50,50,50,white,puzzle[1][9][2])
# ROW 2
button(puzzle[2][1][0],30,105,50,50,white,puzzle[2][1][2])
button(puzzle[2][2][0],85,105,50,50,white,puzzle[2][2][2])
button(puzzle[2][3][0],140,105,50,50,white,puzzle[2][3][2])
button(puzzle[2][4][0],195,105,50,50,white,puzzle[2][4][2])
button(puzzle[2][5][0],250,105,50,50,white,puzzle[2][5][2])
button(puzzle[2][6][0],305,105,50,50,white,puzzle[2][6][2])
button(puzzle[2][7][0],360,105,50,50,white,puzzle[2][7][2])
button(puzzle[2][8][0],415,105,50,50,white,puzzle[2][8][2])
button(puzzle[2][9][0],470,105,50,50,white,puzzle[2][9][2])
# ROW 3
button(puzzle[3][1][0],30,160,50,50,white,puzzle[3][1][2])
button(puzzle[3][2][0],85,160,50,50,white,puzzle[3][2][2])
button(puzzle[3][3][0],140,160,50,50,white,puzzle[3][3][2])
button(puzzle[3][4][0],195,160,50,50,white,puzzle[3][4][2])
button(puzzle[3][5][0],250,160,50,50,white,puzzle[3][5][2])
button(puzzle[3][6][0],305,160,50,50,white,puzzle[3][6][2])
button(puzzle[3][7][0],360,160,50,50,white,puzzle[3][7][2])
button(puzzle[3][8][0],415,160,50,50,white,puzzle[3][8][2])
button(puzzle[3][9][0],470,160,50,50,white,puzzle[3][9][2])
# ROW 4
button(puzzle[4][1][0],30,215,50,50,white,puzzle[4][1][2])
button(puzzle[4][2][0],85,215,50,50,white,puzzle[4][2][2])
button(puzzle[4][3][0],140,215,50,50,white,puzzle[4][3][2])
button(puzzle[4][4][0],195,215,50,50,white,puzzle[4][4][2])
button(puzzle[4][5][0],250,215,50,50,white,puzzle[4][5][2])
button(puzzle[4][6][0],305,215,50,50,white,puzzle[4][6][2])
button(puzzle[4][7][0],360,215,50,50,white,puzzle[4][7][2])
button(puzzle[4][8][0],415,215,50,50,white,puzzle[4][8][2])
button(puzzle[4][9][0],470,215,50,50,white,puzzle[4][9][2])
# ROW 5
button(puzzle[5][1][0],30,270,50,50,white,puzzle[5][1][2])
button(puzzle[5][2][0],85,270,50,50,white,puzzle[5][2][2])
button(puzzle[5][3][0],140,270,50,50,white,puzzle[5][3][2])
button(puzzle[5][4][0],195,270,50,50,white,puzzle[5][4][2])
button(puzzle[5][5][0],250,270,50,50,white,puzzle[5][5][2])
button(puzzle[5][6][0],305,270,50,50,white,puzzle[5][6][2])
button(puzzle[5][7][0],360,270,50,50,white,puzzle[5][7][2])
button(puzzle[5][8][0],415,270,50,50,white,puzzle[5][8][2])
button(puzzle[5][9][0],470,270,50,50,white,puzzle[5][9][2])
# ROW 6
button(puzzle[6][1][0],30,325,50,50,white,puzzle[6][1][2])
button(puzzle[6][2][0],85,325,50,50,white,puzzle[6][2][2])
button(puzzle[6][3][0],140,325,50,50,white,puzzle[6][3][2])
button(puzzle[6][4][0],195,325,50,50,white,puzzle[6][4][2])
button(puzzle[6][5][0],250,325,50,50,white,puzzle[6][5][2])
button(puzzle[6][6][0],305,325,50,50,white,puzzle[6][6][2])
button(puzzle[6][7][0],360,325,50,50,white,puzzle[6][7][2])
button(puzzle[6][8][0],415,325,50,50,white,puzzle[6][8][2])
button(puzzle[6][9][0],470,325,50,50,white,puzzle[6][9][2])
# ROW 7
button(puzzle[7][1][0],30,380,50,50,white,puzzle[7][1][2])
button(puzzle[7][2][0],85,380,50,50,white,puzzle[7][2][2])
button(puzzle[7][3][0],140,380,50,50,white,puzzle[7][3][2])
button(puzzle[7][4][0],195,380,50,50,white,puzzle[7][4][2])
button(puzzle[7][5][0],250,380,50,50,white,puzzle[7][5][2])
button(puzzle[7][6][0],305,380,50,50,white,puzzle[7][6][2])
button(puzzle[7][7][0],360,380,50,50,white,puzzle[7][7][2])
button(puzzle[7][8][0],415,380,50,50,white,puzzle[7][8][2])
button(puzzle[7][9][0],470,380,50,50,white,puzzle[7][9][2])
# ROW 8
button(puzzle[8][1][0],30,435,50,50,white,puzzle[8][1][2])
button(puzzle[8][2][0],85,435,50,50,white,puzzle[8][2][2])
button(puzzle[8][3][0],140,435,50,50,white,puzzle[8][3][2])
button(puzzle[8][4][0],195,435,50,50,white,puzzle[8][4][2])
button(puzzle[8][5][0],250,435,50,50,white,puzzle[8][5][2])
button(puzzle[8][6][0],305,435,50,50,white,puzzle[8][6][2])
```

```
      button(puzzle[8][7][0],360,435,50,50,white,puzzle[8][7][2])
      button(puzzle[8][8][0],415,435,50,50,white,puzzle[8][8][2])
      button(puzzle[8][9][0],470,435,50,50,white,puzzle[8][9][2])
      # ROW 9
      button(puzzle[9][1][0],30,490,50,50,white,puzzle[9][1][2])
      button(puzzle[9][2][0],85,490,50,50,white,puzzle[9][2][2])
      button(puzzle[9][3][0],140,490,50,50,white,puzzle[9][3][2])
      button(puzzle[9][4][0],195,490,50,50,white,puzzle[9][4][2])
      button(puzzle[9][5][0],250,490,50,50,white,puzzle[9][5][2])
      button(puzzle[9][6][0],305,490,50,50,white,puzzle[9][6][2])
      button(puzzle[9][7][0],360,490,50,50,white,puzzle[9][7][2])
      button(puzzle[9][8][0],415,490,50,50,white,puzzle[9][8][2])
      button(puzzle[9][9][0],470,490,50,50,white,puzzle[9][9][2])

      pygame.draw.line(screen,black,(192,50),(192,540),5)
      pygame.draw.line(screen,black,(357,50),(357,540),5)
      pygame.draw.line(screen,black,(30,212),(520,212),5)
      pygame.draw.line(screen,black,(30,377),(520,377),5)

      button("Back",90,550,150,50,white,gray)

      pygame.display.flip()
      clock.tick(60)

# when medium button is pressed
if 200+150 > mouse[0] > 200 and 230+50 > mouse[1] > 230 and click[0] == 1:
   medium = True

   puzzle = generatePuzzle("medium")

   while medium:
      for event in pygame.event.get():
         if event.type == pygame.QUIT:
            medium = False
            freeOps = False
            done = True
         mouse = pygame.mouse.get_pos()
         x = mouse[0]
         y = mouse[1]
         click = pygame.mouse.get_pressed()
         one = pygame.key.get_pressed()[pygame.K_1]
         two = pygame.key.get_pressed()[pygame.K_2]
         three = pygame.key.get_pressed()[pygame.K_3]
         four = pygame.key.get_pressed()[pygame.K_4]
```

```
                        five = pygame.key.get_pressed()[pygame.K_5]
                        six = pygame.key.get_pressed()[pygame.K_6]
                        seven = pygame.key.get_pressed()[pygame.K_7]
                        eight = pygame.key.get_pressed()[pygame.K_8]
                        nine = pygame.key.get_pressed()[pygame.K_9]
                        delete = pygame.key.get_pressed()[pygame.K_BACKSPACE]
                        hint = pygame.key.get_pressed()[pygame.K_h]

                        # when back key is pressed
                        if 90+150 > mouse[0] > 90 and 550+50 > mouse[1] > 550 and click[0] == 1:
                           medium = False

                        # when play again button is pressed
                        if 310+150 > x > 310 and 550+50 > y > 550 and click[0] == 1 and
solCheck(puzzle):
                           medium = False

                        # if a number key, backspace, or h is pressed then take the appropriate action
for the
                        # box selected
                        if one == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                           puzzle[(y-50)/55+1][(x-30)/55+1][0] = "1"

                        elif two == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                           puzzle[(y-50)/55+1][(x-30)/55+1][0] = "2"

                        elif three == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                           puzzle[(y-50)/55+1][(x-30)/55+1][0] = "3"

                        elif four == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                           puzzle[(y-50)/55+1][(x-30)/55+1][0] = "4"

                        elif five == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                           puzzle[(y-50)/55+1][(x-30)/55+1][0] = "5"

                        elif six == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                           puzzle[(y-50)/55+1][(x-30)/55+1][0] = "6"

                        elif seven == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                           puzzle[(y-50)/55+1][(x-30)/55+1][0] = "7"

                        elif eight == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                           puzzle[(y-50)/55+1][(x-30)/55+1][0] = "8"
```

```
    elif nine == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "9"

    elif delete == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = ""

    elif hint == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = puzzle[(y-50)/55+1][(x-30)/55+1][1]

screen.blit(defimg,rect)

# display these if the game has been won
midText = pygame.font.Font("freesansbold.ttf",20)
if solCheck(puzzle):
    textSurf, textRect = text_objects("Congratulations! You won!", midText)
    textRect.center = (size[0]/2,25)
    screen.blit(textSurf, textRect)
    button("Play Again",310,550,150,50,white,gray)

# display these if the game is still continuing
else:
    textSurf, textRect = text_objects("Medium", midText)
    textRect.center = (size[0]/2,25)
    screen.blit(textSurf, textRect)

# display box code goes here
# ROW 1
button(puzzle[1][1][0],30,50,50,50,white,puzzle[1][1][2])
button(puzzle[1][2][0],85,50,50,50,white,puzzle[1][2][2])
button(puzzle[1][3][0],140,50,50,50,white,puzzle[1][3][2])
button(puzzle[1][4][0],195,50,50,50,white,puzzle[1][4][2])
button(puzzle[1][5][0],250,50,50,50,white,puzzle[1][5][2])
button(puzzle[1][6][0],305,50,50,50,white,puzzle[1][6][2])
button(puzzle[1][7][0],360,50,50,50,white,puzzle[1][7][2])
button(puzzle[1][8][0],415,50,50,50,white,puzzle[1][8][2])
button(puzzle[1][9][0],470,50,50,50,white,puzzle[1][9][2])
# ROW 2
button(puzzle[2][1][0],30,105,50,50,white,puzzle[2][1][2])
button(puzzle[2][2][0],85,105,50,50,white,puzzle[2][2][2])
button(puzzle[2][3][0],140,105,50,50,white,puzzle[2][3][2])
button(puzzle[2][4][0],195,105,50,50,white,puzzle[2][4][2])
button(puzzle[2][5][0],250,105,50,50,white,puzzle[2][5][2])
button(puzzle[2][6][0],305,105,50,50,white,puzzle[2][6][2])
button(puzzle[2][7][0],360,105,50,50,white,puzzle[2][7][2])
button(puzzle[2][8][0],415,105,50,50,white,puzzle[2][8][2])
button(puzzle[2][9][0],470,105,50,50,white,puzzle[2][9][2])
# ROW 3
button(puzzle[3][1][0],30,160,50,50,white,puzzle[3][1][2])
button(puzzle[3][2][0],85,160,50,50,white,puzzle[3][2][2])
button(puzzle[3][3][0],140,160,50,50,white,puzzle[3][3][2])
button(puzzle[3][4][0],195,160,50,50,white,puzzle[3][4][2])
button(puzzle[3][5][0],250,160,50,50,white,puzzle[3][5][2])
button(puzzle[3][6][0],305,160,50,50,white,puzzle[3][6][2])
button(puzzle[3][7][0],360,160,50,50,white,puzzle[3][7][2])
button(puzzle[3][8][0],415,160,50,50,white,puzzle[3][8][2])
button(puzzle[3][9][0],470,160,50,50,white,puzzle[3][9][2])
# ROW 4
button(puzzle[4][1][0],30,215,50,50,white,puzzle[4][1][2])
button(puzzle[4][2][0],85,215,50,50,white,puzzle[4][2][2])
button(puzzle[4][3][0],140,215,50,50,white,puzzle[4][3][2])
button(puzzle[4][4][0],195,215,50,50,white,puzzle[4][4][2])
button(puzzle[4][5][0],250,215,50,50,white,puzzle[4][5][2])
button(puzzle[4][6][0],305,215,50,50,white,puzzle[4][6][2])
button(puzzle[4][7][0],360,215,50,50,white,puzzle[4][7][2])
button(puzzle[4][8][0],415,215,50,50,white,puzzle[4][8][2])
button(puzzle[4][9][0],470,215,50,50,white,puzzle[4][9][2])
# ROW 5
button(puzzle[5][1][0],30,270,50,50,white,puzzle[5][1][2])
button(puzzle[5][2][0],85,270,50,50,white,puzzle[5][2][2])
button(puzzle[5][3][0],140,270,50,50,white,puzzle[5][3][2])
button(puzzle[5][4][0],195,270,50,50,white,puzzle[5][4][2])
button(puzzle[5][5][0],250,270,50,50,white,puzzle[5][5][2])
button(puzzle[5][6][0],305,270,50,50,white,puzzle[5][6][2])
button(puzzle[5][7][0],360,270,50,50,white,puzzle[5][7][2])
button(puzzle[5][8][0],415,270,50,50,white,puzzle[5][8][2])
button(puzzle[5][9][0],470,270,50,50,white,puzzle[5][9][2])
# ROW 6
button(puzzle[6][1][0],30,325,50,50,white,puzzle[6][1][2])
button(puzzle[6][2][0],85,325,50,50,white,puzzle[6][2][2])
button(puzzle[6][3][0],140,325,50,50,white,puzzle[6][3][2])
button(puzzle[6][4][0],195,325,50,50,white,puzzle[6][4][2])
button(puzzle[6][5][0],250,325,50,50,white,puzzle[6][5][2])
button(puzzle[6][6][0],305,325,50,50,white,puzzle[6][6][2])
button(puzzle[6][7][0],360,325,50,50,white,puzzle[6][7][2])
button(puzzle[6][8][0],415,325,50,50,white,puzzle[6][8][2])
button(puzzle[6][9][0],470,325,50,50,white,puzzle[6][9][2])
# ROW 7
button(puzzle[7][1][0],30,380,50,50,white,puzzle[7][1][2])
```

```
        button(puzzle[7][2][0],85,380,50,50,white,puzzle[7][2][2])
        button(puzzle[7][3][0],140,380,50,50,white,puzzle[7][3][2])
        button(puzzle[7][4][0],195,380,50,50,white,puzzle[7][4][2])
        button(puzzle[7][5][0],250,380,50,50,white,puzzle[7][5][2])
        button(puzzle[7][6][0],305,380,50,50,white,puzzle[7][6][2])
        button(puzzle[7][7][0],360,380,50,50,white,puzzle[7][7][2])
        button(puzzle[7][8][0],415,380,50,50,white,puzzle[7][8][2])
        button(puzzle[7][9][0],470,380,50,50,white,puzzle[7][9][2])
        # ROW 8
        button(puzzle[8][1][0],30,435,50,50,white,puzzle[8][1][2])
        button(puzzle[8][2][0],85,435,50,50,white,puzzle[8][2][2])
        button(puzzle[8][3][0],140,435,50,50,white,puzzle[8][3][2])
        button(puzzle[8][4][0],195,435,50,50,white,puzzle[8][4][2])
        button(puzzle[8][5][0],250,435,50,50,white,puzzle[8][5][2])
        button(puzzle[8][6][0],305,435,50,50,white,puzzle[8][6][2])
        button(puzzle[8][7][0],360,435,50,50,white,puzzle[8][7][2])
        button(puzzle[8][8][0],415,435,50,50,white,puzzle[8][8][2])
        button(puzzle[8][9][0],470,435,50,50,white,puzzle[8][9][2])
        # ROW 9
        button(puzzle[9][1][0],30,490,50,50,white,puzzle[9][1][2])
        button(puzzle[9][2][0],85,490,50,50,white,puzzle[9][2][2])
        button(puzzle[9][3][0],140,490,50,50,white,puzzle[9][3][2])
        button(puzzle[9][4][0],195,490,50,50,white,puzzle[9][4][2])
        button(puzzle[9][5][0],250,490,50,50,white,puzzle[9][5][2])
        button(puzzle[9][6][0],305,490,50,50,white,puzzle[9][6][2])
        button(puzzle[9][7][0],360,490,50,50,white,puzzle[9][7][2])
        button(puzzle[9][8][0],415,490,50,50,white,puzzle[9][8][2])
        button(puzzle[9][9][0],470,490,50,50,white,puzzle[9][9][2])

        pygame.draw.line(screen,black,(192,50),(192,540),5)
        pygame.draw.line(screen,black,(357,50),(357,540),5)
        pygame.draw.line(screen,black,(30,212),(520,212),5)
        pygame.draw.line(screen,black,(30,377),(520,377),5)

        button("Back",90,550,150,50,white,gray)

        pygame.display.flip()
        clock.tick(60)

# when hard button is pressed
if 200+150 > mouse[0] > 200 and 300+50 > mouse[1] > 300 and click[0] == 1:
    hard = True

    puzzle = generatePuzzle("hard")
```

```
                        while hard:
                            for event in pygame.event.get():
                                if event.type == pygame.QUIT:
                                    hard = False
                                    freeOps = False
                                    done = True
                                mouse = pygame.mouse.get_pos()
                                x = mouse[0]
                                y = mouse[1]
                                click = pygame.mouse.get_pressed()
                                one = pygame.key.get_pressed()[pygame.K_1]
                                two = pygame.key.get_pressed()[pygame.K_2]
                                three = pygame.key.get_pressed()[pygame.K_3]
                                four = pygame.key.get_pressed()[pygame.K_4]
                                five = pygame.key.get_pressed()[pygame.K_5]
                                six = pygame.key.get_pressed()[pygame.K_6]
                                seven = pygame.key.get_pressed()[pygame.K_7]
                                eight = pygame.key.get_pressed()[pygame.K_8]
                                nine = pygame.key.get_pressed()[pygame.K_9]
                                delete = pygame.key.get_pressed()[pygame.K_BACKSPACE]
                                hint = pygame.key.get_pressed()[pygame.K_h]

                                # when back key is pressed
                                if 90+150 > mouse[0] > 90 and 550+50 > mouse[1] > 550 and click[0] == 1:
                                    hard = False

                                # when play again button is pressed
                                if 310+150 > x > 310 and 550+50 > y > 550 and click[0] == 1 and
solCheck(puzzle):
                                    hard = False

                                # if a number key, backspace, or h is pressed then take the appropriate action
for the
                                # box selected
                                if one == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                                    puzzle[(y-50)/55+1][(x-30)/55+1][0] = "1"

                                elif two == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                                    puzzle[(y-50)/55+1][(x-30)/55+1][0] = "2"

                                elif three == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                                    puzzle[(y-50)/55+1][(x-30)/55+1][0] = "3"
```

```
    elif four == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "4"

    elif five == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "5"

    elif six == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "6"

    elif seven == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "7"

    elif eight == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "8"

    elif nine == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "9"

    elif delete == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = ""

    elif hint == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = puzzle[(y-50)/55+1][(x-30)/55+1][1]

screen.blit(defimg,rect)

# display these if the game has been won
midText = pygame.font.Font("freesansbold.ttf",20)
if solCheck(puzzle):
    textSurf, textRect = text_objects("Congratulations! You won!", midText)
    textRect.center = (size[0]/2,25)
    screen.blit(textSurf, textRect)
    button("Play Again",310,550,150,50,white,gray)

# display these if the game has been won
else:
    textSurf, textRect = text_objects("Hard", midText)
    textRect.center = (size[0]/2,25)
    screen.blit(textSurf, textRect)

# display box code goes here
# ROW 1
button(puzzle[1][1][0],30,50,50,50,white,puzzle[1][1][2])
button(puzzle[1][2][0],85,50,50,50,white,puzzle[1][2][2])
button(puzzle[1][3][0],140,50,50,50,white,puzzle[1][3][2])
button(puzzle[1][4][0],195,50,50,50,white,puzzle[1][4][2])
button(puzzle[1][5][0],250,50,50,50,white,puzzle[1][5][2])
button(puzzle[1][6][0],305,50,50,50,white,puzzle[1][6][2])
button(puzzle[1][7][0],360,50,50,50,white,puzzle[1][7][2])
button(puzzle[1][8][0],415,50,50,50,white,puzzle[1][8][2])
button(puzzle[1][9][0],470,50,50,50,white,puzzle[1][9][2])
# ROW 2
button(puzzle[2][1][0],30,105,50,50,white,puzzle[2][1][2])
button(puzzle[2][2][0],85,105,50,50,white,puzzle[2][2][2])
button(puzzle[2][3][0],140,105,50,50,white,puzzle[2][3][2])
button(puzzle[2][4][0],195,105,50,50,white,puzzle[2][4][2])
button(puzzle[2][5][0],250,105,50,50,white,puzzle[2][5][2])
button(puzzle[2][6][0],305,105,50,50,white,puzzle[2][6][2])
button(puzzle[2][7][0],360,105,50,50,white,puzzle[2][7][2])
button(puzzle[2][8][0],415,105,50,50,white,puzzle[2][8][2])
button(puzzle[2][9][0],470,105,50,50,white,puzzle[2][9][2])
# ROW 3
button(puzzle[3][1][0],30,160,50,50,white,puzzle[3][1][2])
button(puzzle[3][2][0],85,160,50,50,white,puzzle[3][2][2])
button(puzzle[3][3][0],140,160,50,50,white,puzzle[3][3][2])
button(puzzle[3][4][0],195,160,50,50,white,puzzle[3][4][2])
button(puzzle[3][5][0],250,160,50,50,white,puzzle[3][5][2])
button(puzzle[3][6][0],305,160,50,50,white,puzzle[3][6][2])
button(puzzle[3][7][0],360,160,50,50,white,puzzle[3][7][2])
button(puzzle[3][8][0],415,160,50,50,white,puzzle[3][8][2])
button(puzzle[3][9][0],470,160,50,50,white,puzzle[3][9][2])
# ROW 4
button(puzzle[4][1][0],30,215,50,50,white,puzzle[4][1][2])
button(puzzle[4][2][0],85,215,50,50,white,puzzle[4][2][2])
button(puzzle[4][3][0],140,215,50,50,white,puzzle[4][3][2])
button(puzzle[4][4][0],195,215,50,50,white,puzzle[4][4][2])
button(puzzle[4][5][0],250,215,50,50,white,puzzle[4][5][2])
button(puzzle[4][6][0],305,215,50,50,white,puzzle[4][6][2])
button(puzzle[4][7][0],360,215,50,50,white,puzzle[4][7][2])
button(puzzle[4][8][0],415,215,50,50,white,puzzle[4][8][2])
button(puzzle[4][9][0],470,215,50,50,white,puzzle[4][9][2])
# ROW 5
button(puzzle[5][1][0],30,270,50,50,white,puzzle[5][1][2])
button(puzzle[5][2][0],85,270,50,50,white,puzzle[5][2][2])
button(puzzle[5][3][0],140,270,50,50,white,puzzle[5][3][2])
button(puzzle[5][4][0],195,270,50,50,white,puzzle[5][4][2])
button(puzzle[5][5][0],250,270,50,50,white,puzzle[5][5][2])
button(puzzle[5][6][0],305,270,50,50,white,puzzle[5][6][2])
button(puzzle[5][7][0],360,270,50,50,white,puzzle[5][7][2])
button(puzzle[5][8][0],415,270,50,50,white,puzzle[5][8][2])
button(puzzle[5][9][0],470,270,50,50,white,puzzle[5][9][2])
# ROW 6
button(puzzle[6][1][0],30,325,50,50,white,puzzle[6][1][2])
button(puzzle[6][2][0],85,325,50,50,white,puzzle[6][2][2])
button(puzzle[6][3][0],140,325,50,50,white,puzzle[6][3][2])
button(puzzle[6][4][0],195,325,50,50,white,puzzle[6][4][2])
button(puzzle[6][5][0],250,325,50,50,white,puzzle[6][5][2])
button(puzzle[6][6][0],305,325,50,50,white,puzzle[6][6][2])
button(puzzle[6][7][0],360,325,50,50,white,puzzle[6][7][2])
button(puzzle[6][8][0],415,325,50,50,white,puzzle[6][8][2])
button(puzzle[6][9][0],470,325,50,50,white,puzzle[6][9][2])
# ROW 7
button(puzzle[7][1][0],30,380,50,50,white,puzzle[7][1][2])
button(puzzle[7][2][0],85,380,50,50,white,puzzle[7][2][2])
button(puzzle[7][3][0],140,380,50,50,white,puzzle[7][3][2])
button(puzzle[7][4][0],195,380,50,50,white,puzzle[7][4][2])
button(puzzle[7][5][0],250,380,50,50,white,puzzle[7][5][2])
button(puzzle[7][6][0],305,380,50,50,white,puzzle[7][6][2])
button(puzzle[7][7][0],360,380,50,50,white,puzzle[7][7][2])
button(puzzle[7][8][0],415,380,50,50,white,puzzle[7][8][2])
button(puzzle[7][9][0],470,380,50,50,white,puzzle[7][9][2])
# ROW 8
button(puzzle[8][1][0],30,435,50,50,white,puzzle[8][1][2])
button(puzzle[8][2][0],85,435,50,50,white,puzzle[8][2][2])
button(puzzle[8][3][0],140,435,50,50,white,puzzle[8][3][2])
button(puzzle[8][4][0],195,435,50,50,white,puzzle[8][4][2])
button(puzzle[8][5][0],250,435,50,50,white,puzzle[8][5][2])
button(puzzle[8][6][0],305,435,50,50,white,puzzle[8][6][2])
button(puzzle[8][7][0],360,435,50,50,white,puzzle[8][7][2])
button(puzzle[8][8][0],415,435,50,50,white,puzzle[8][8][2])
button(puzzle[8][9][0],470,435,50,50,white,puzzle[8][9][2])
# ROW 9
button(puzzle[9][1][0],30,490,50,50,white,puzzle[9][1][2])
button(puzzle[9][2][0],85,490,50,50,white,puzzle[9][2][2])
button(puzzle[9][3][0],140,490,50,50,white,puzzle[9][3][2])
button(puzzle[9][4][0],195,490,50,50,white,puzzle[9][4][2])
button(puzzle[9][5][0],250,490,50,50,white,puzzle[9][5][2])
button(puzzle[9][6][0],305,490,50,50,white,puzzle[9][6][2])
button(puzzle[9][7][0],360,490,50,50,white,puzzle[9][7][2])
button(puzzle[9][8][0],415,490,50,50,white,puzzle[9][8][2])
button(puzzle[9][9][0],470,490,50,50,white,puzzle[9][9][2])
```

```
               pygame.draw.line(screen,black,(192,50),(192,540),5)
               pygame.draw.line(screen,black,(357,50),(357,540),5)
               pygame.draw.line(screen,black,(30,212),(520,212),5)
               pygame.draw.line(screen,black,(30,377),(520,377),5)

               button("Back",90,550,150,50,white,gray)

               pygame.display.flip()
               clock.tick(50)

      screen.blit(defimg,rect)
      midText = pygame.font.Font("freesansbold.ttf",60)
      textSurf, textRect = text_objects("Sudoku", midText)
      textRect.center = (size[0]/2,(size[1]/2)-200)
      screen.blit(textSurf, textRect)

      button("Easy",200,160,150,50,white,gray)
      button("Medium",200,230,150,50,white,gray)
      button("Hard",200,300,150,50,white,gray)
      button("Back",200,370,150,50,white,gray)

      pygame.display.flip()
      clock.tick(60)

# when the timed play button is pressed
if 100+150 > mouse[0] > 100 and 230+50 > mouse[1] > 230 and click[0] == 1:
   timedOps = True
   while timedOps:
      for event in pygame.event.get():
         if event.type == pygame.QUIT:
            timedOps = False
            done = True
         mouse = pygame.mouse.get_pos()
         click = pygame.mouse.get_pressed()

         # when back key is pressed
         if 200+150 > mouse[0] > 200 and 370+50 > mouse[1] > 370 and click[0] == 1:
            timedOps = False

         # when easy button is pressed
         if 200+150 > mouse[0] > 200 and 160+50 > mouse[1] > 160 and click[0] == 1:
            easy = True

            puzzle = generatePuzzle("easy")
            time = pygame.time.get_ticks()

            while easy:
               for event in pygame.event.get():
                  if event.type == pygame.QUIT:
                     easy = False
                     timedOps = False
                     done = True
                  mouse = pygame.mouse.get_pos()
                  x = mouse[0]
                  y = mouse[1]
                  click = pygame.mouse.get_pressed()
                  one = pygame.key.get_pressed()[pygame.K_1]
                  two = pygame.key.get_pressed()[pygame.K_2]
                  three = pygame.key.get_pressed()[pygame.K_3]
                  four = pygame.key.get_pressed()[pygame.K_4]
                  five = pygame.key.get_pressed()[pygame.K_5]
                  six = pygame.key.get_pressed()[pygame.K_6]
                  seven = pygame.key.get_pressed()[pygame.K_7]
                  eight = pygame.key.get_pressed()[pygame.K_8]
                  nine = pygame.key.get_pressed()[pygame.K_9]
                  delete = pygame.key.get_pressed()[pygame.K_BACKSPACE]
                  hint = pygame.key.get_pressed()[pygame.K_h]

                  # when back key is pressed
                  if 90+150 > mouse[0] > 90 and 550+50 > mouse[1] > 550 and click[0] == 1:
                     easy = False

                  # when play again button is pressed
                  if 310+150 > x > 310 and 550+50 > y > 550 and click[0] == 1 and
solCheck(puzzle):
                     easy = False

                  # if a number key, backspace, or h is pressed then take the appropriate action
for the
                  # box selected
                  if one == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                     puzzle[(y-50)/55+1][(x-30)/55+1][0] = "1"

                  elif two == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                     puzzle[(y-50)/55+1][(x-30)/55+1][0] = "2"

                  elif three == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                     puzzle[(y-50)/55+1][(x-30)/55+1][0] = "3"
```

```
    elif four == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "4"

    elif five == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "5"

    elif six == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "6"

    elif seven == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "7"

    elif eight == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "8"

    elif nine == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "9"

    elif delete == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = ""

    elif hint == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = puzzle[(y-50)/55+1][(x-30)/55+1][1]

screen.blit(defimg,rect)

# display these if the game has been won
midText = pygame.font.Font("freesansbold.ttf",20)
if solCheck(puzzle):
    textSurf, textRect = text_objects("Congratulations! You won!", midText)
    textRect.center = (size[0]/2,25)
    screen.blit(textSurf, textRect)
    button("Play Again",310,550,150,50,white,gray)
    button(str(timer/1000),245,550,60,50,white,white)
    if timer/1000 < int(easyHigh):
        easyHigh = str(timer/1000)

# display these if the game is still continuing
else:
    textSurf, textRect = text_objects("Easy", midText)
    textRect.center = (size[0]/2,25)
    screen.blit(textSurf, textRect)
    timer = (pygame.time.get_ticks())-time
    button(str(timer/1000),245,550,60,50,white,white)

# display box code goes here
# ROW 1
button(puzzle[1][1][0],30,50,50,50,white,puzzle[1][1][2])
button(puzzle[1][2][0],85,50,50,50,white,puzzle[1][2][2])
button(puzzle[1][3][0],140,50,50,50,white,puzzle[1][3][2])
button(puzzle[1][4][0],195,50,50,50,white,puzzle[1][4][2])
button(puzzle[1][5][0],250,50,50,50,white,puzzle[1][5][2])
button(puzzle[1][6][0],305,50,50,50,white,puzzle[1][6][2])
button(puzzle[1][7][0],360,50,50,50,white,puzzle[1][7][2])
button(puzzle[1][8][0],415,50,50,50,white,puzzle[1][8][2])
button(puzzle[1][9][0],470,50,50,50,white,puzzle[1][9][2])
# ROW 2
button(puzzle[2][1][0],30,105,50,50,white,puzzle[2][1][2])
button(puzzle[2][2][0],85,105,50,50,white,puzzle[2][2][2])
button(puzzle[2][3][0],140,105,50,50,white,puzzle[2][3][2])
button(puzzle[2][4][0],195,105,50,50,white,puzzle[2][4][2])
button(puzzle[2][5][0],250,105,50,50,white,puzzle[2][5][2])
button(puzzle[2][6][0],305,105,50,50,white,puzzle[2][6][2])
button(puzzle[2][7][0],360,105,50,50,white,puzzle[2][7][2])
button(puzzle[2][8][0],415,105,50,50,white,puzzle[2][8][2])
button(puzzle[2][9][0],470,105,50,50,white,puzzle[2][9][2])
# ROW 3
button(puzzle[3][1][0],30,160,50,50,white,puzzle[3][1][2])
button(puzzle[3][2][0],85,160,50,50,white,puzzle[3][2][2])
button(puzzle[3][3][0],140,160,50,50,white,puzzle[3][3][2])
button(puzzle[3][4][0],195,160,50,50,white,puzzle[3][4][2])
button(puzzle[3][5][0],250,160,50,50,white,puzzle[3][5][2])
button(puzzle[3][6][0],305,160,50,50,white,puzzle[3][6][2])
button(puzzle[3][7][0],360,160,50,50,white,puzzle[3][7][2])
button(puzzle[3][8][0],415,160,50,50,white,puzzle[3][8][2])
button(puzzle[3][9][0],470,160,50,50,white,puzzle[3][9][2])
# ROW 4
button(puzzle[4][1][0],30,215,50,50,white,puzzle[4][1][2])
button(puzzle[4][2][0],85,215,50,50,white,puzzle[4][2][2])
button(puzzle[4][3][0],140,215,50,50,white,puzzle[4][3][2])
button(puzzle[4][4][0],195,215,50,50,white,puzzle[4][4][2])
button(puzzle[4][5][0],250,215,50,50,white,puzzle[4][5][2])
button(puzzle[4][6][0],305,215,50,50,white,puzzle[4][6][2])
button(puzzle[4][7][0],360,215,50,50,white,puzzle[4][7][2])
button(puzzle[4][8][0],415,215,50,50,white,puzzle[4][8][2])
button(puzzle[4][9][0],470,215,50,50,white,puzzle[4][9][2])
# ROW 5
```

```
button(puzzle[5][1][0],30,270,50,50,white,puzzle[5][1][2])
button(puzzle[5][2][0],85,270,50,50,white,puzzle[5][2][2])
button(puzzle[5][3][0],140,270,50,50,white,puzzle[5][3][2])
button(puzzle[5][4][0],195,270,50,50,white,puzzle[5][4][2])
button(puzzle[5][5][0],250,270,50,50,white,puzzle[5][5][2])
button(puzzle[5][6][0],305,270,50,50,white,puzzle[5][6][2])
button(puzzle[5][7][0],360,270,50,50,white,puzzle[5][7][2])
button(puzzle[5][8][0],415,270,50,50,white,puzzle[5][8][2])
button(puzzle[5][9][0],470,270,50,50,white,puzzle[5][9][2])
# ROW 6
button(puzzle[6][1][0],30,325,50,50,white,puzzle[6][1][2])
button(puzzle[6][2][0],85,325,50,50,white,puzzle[6][2][2])
button(puzzle[6][3][0],140,325,50,50,white,puzzle[6][3][2])
button(puzzle[6][4][0],195,325,50,50,white,puzzle[6][4][2])
button(puzzle[6][5][0],250,325,50,50,white,puzzle[6][5][2])
button(puzzle[6][6][0],305,325,50,50,white,puzzle[6][6][2])
button(puzzle[6][7][0],360,325,50,50,white,puzzle[6][7][2])
button(puzzle[6][8][0],415,325,50,50,white,puzzle[6][8][2])
button(puzzle[6][9][0],470,325,50,50,white,puzzle[6][9][2])
# ROW 7
button(puzzle[7][1][0],30,380,50,50,white,puzzle[7][1][2])
button(puzzle[7][2][0],85,380,50,50,white,puzzle[7][2][2])
button(puzzle[7][3][0],140,380,50,50,white,puzzle[7][3][2])
button(puzzle[7][4][0],195,380,50,50,white,puzzle[7][4][2])
button(puzzle[7][5][0],250,380,50,50,white,puzzle[7][5][2])
button(puzzle[7][6][0],305,380,50,50,white,puzzle[7][6][2])
button(puzzle[7][7][0],360,380,50,50,white,puzzle[7][7][2])
button(puzzle[7][8][0],415,380,50,50,white,puzzle[7][8][2])
button(puzzle[7][9][0],470,380,50,50,white,puzzle[7][9][2])
# ROW 8
button(puzzle[8][1][0],30,435,50,50,white,puzzle[8][1][2])
button(puzzle[8][2][0],85,435,50,50,white,puzzle[8][2][2])
button(puzzle[8][3][0],140,435,50,50,white,puzzle[8][3][2])
button(puzzle[8][4][0],195,435,50,50,white,puzzle[8][4][2])
button(puzzle[8][5][0],250,435,50,50,white,puzzle[8][5][2])
button(puzzle[8][6][0],305,435,50,50,white,puzzle[8][6][2])
button(puzzle[8][7][0],360,435,50,50,white,puzzle[8][7][2])
button(puzzle[8][8][0],415,435,50,50,white,puzzle[8][8][2])
button(puzzle[8][9][0],470,435,50,50,white,puzzle[8][9][2])
# ROW 9
button(puzzle[9][1][0],30,490,50,50,white,puzzle[9][1][2])
button(puzzle[9][2][0],85,490,50,50,white,puzzle[9][2][2])
button(puzzle[9][3][0],140,490,50,50,white,puzzle[9][3][2])
button(puzzle[9][4][0],195,490,50,50,white,puzzle[9][4][2])
```

```
        button(puzzle[9][5][0],250,490,50,50,white,puzzle[9][5][2])
        button(puzzle[9][6][0],305,490,50,50,white,puzzle[9][6][2])
        button(puzzle[9][7][0],360,490,50,50,white,puzzle[9][7][2])
        button(puzzle[9][8][0],415,490,50,50,white,puzzle[9][8][2])
        button(puzzle[9][9][0],470,490,50,50,white,puzzle[9][9][2])

        pygame.draw.line(screen,black,(192,50),(192,540),5)
        pygame.draw.line(screen,black,(357,50),(357,540),5)
        pygame.draw.line(screen,black,(30,212),(520,212),5)
        pygame.draw.line(screen,black,(30,377),(520,377),5)

        button("Back",90,550,150,50,white,gray)

        pygame.display.flip()
        clock1.tick(60)

# when medium button is pressed
if 200+150 > mouse[0] > 200 and 230+50 > mouse[1] > 230 and click[0] == 1:
    medium = True

    puzzle = generatePuzzle("medium")
    time = pygame.time.get_ticks()

    while medium:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                medium = False
                timedOps = False
                done = True
            mouse = pygame.mouse.get_pos()
            x = mouse[0]
            y = mouse[1]
            click = pygame.mouse.get_pressed()
            one = pygame.key.get_pressed()[pygame.K_1]
            two = pygame.key.get_pressed()[pygame.K_2]
            three = pygame.key.get_pressed()[pygame.K_3]
            four = pygame.key.get_pressed()[pygame.K_4]
            five = pygame.key.get_pressed()[pygame.K_5]
            six = pygame.key.get_pressed()[pygame.K_6]
            seven = pygame.key.get_pressed()[pygame.K_7]
            eight = pygame.key.get_pressed()[pygame.K_8]
            nine = pygame.key.get_pressed()[pygame.K_9]
            delete = pygame.key.get_pressed()[pygame.K_BACKSPACE]
            hint = pygame.key.get_pressed()[pygame.K_h]
```

```
                    # when back key is pressed
                    if 90+150 > mouse[0] > 90 and 550+50 > mouse[1] > 550 and click[0] == 1:
                        medium = False

                    # when play again button is pressed
                    if 310+150 > x > 310 and 550+50 > y > 550 and click[0] == 1 and
solCheck(puzzle):
                        medium = False

                    # if a number key, backspace, or h is pressed then take the appropriate action
for the
                    # box selected
                    if one == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "1"

                    elif two == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "2"

                    elif three == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "3"

                    elif four == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "4"

                    elif five == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "5"

                    elif six == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "6"

                    elif seven == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "7"

                    elif eight == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "8"

                    elif nine == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "9"

                    elif delete == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                        puzzle[(y-50)/55+1][(x-30)/55+1][0] = ""

                    elif hint == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
```

```
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = puzzle[(y-50)/55+1][(x-30)/55+1][1]

screen.blit(defimg,rect)

midText = pygame.font.Font("freesansbold.ttf",20)

# display these if the game has been won
if solCheck(puzzle):
    textSurf, textRect = text_objects("Congratulations! You won!", midText)
    textRect.center = (size[0]/2,25)
    screen.blit(textSurf, textRect)
    button("Play Again",310,550,150,50,white,gray)
    button(str(timer/1000),245,550,60,50,white,white)
    if timer/1000 < int(mediumHigh):
        mediumHigh = str(timer/1000)

# display these if the game has been won
else:
    textSurf, textRect = text_objects("Medium", midText)
    textRect.center = (size[0]/2,25)
    screen.blit(textSurf, textRect)
    timer = (pygame.time.get_ticks()) - time
    button(str(timer/1000),245,550,60,50,white,white)

# display box code goes here
# ROW 1
button(puzzle[1][1][0],30,50,50,50,white,puzzle[1][1][2])
button(puzzle[1][2][0],85,50,50,50,white,puzzle[1][2][2])
button(puzzle[1][3][0],140,50,50,50,white,puzzle[1][3][2])
button(puzzle[1][4][0],195,50,50,50,white,puzzle[1][4][2])
button(puzzle[1][5][0],250,50,50,50,white,puzzle[1][5][2])
button(puzzle[1][6][0],305,50,50,50,white,puzzle[1][6][2])
button(puzzle[1][7][0],360,50,50,50,white,puzzle[1][7][2])
button(puzzle[1][8][0],415,50,50,50,white,puzzle[1][8][2])
button(puzzle[1][9][0],470,50,50,50,white,puzzle[1][9][2])
# ROW 2
button(puzzle[2][1][0],30,105,50,50,white,puzzle[2][1][2])
button(puzzle[2][2][0],85,105,50,50,white,puzzle[2][2][2])
button(puzzle[2][3][0],140,105,50,50,white,puzzle[2][3][2])
button(puzzle[2][4][0],195,105,50,50,white,puzzle[2][4][2])
button(puzzle[2][5][0],250,105,50,50,white,puzzle[2][5][2])
button(puzzle[2][6][0],305,105,50,50,white,puzzle[2][6][2])
button(puzzle[2][7][0],360,105,50,50,white,puzzle[2][7][2])
button(puzzle[2][8][0],415,105,50,50,white,puzzle[2][8][2])
button(puzzle[2][9][0],470,105,50,50,white,puzzle[2][9][2])
# ROW 3
button(puzzle[3][1][0],30,160,50,50,white,puzzle[3][1][2])
button(puzzle[3][2][0],85,160,50,50,white,puzzle[3][2][2])
button(puzzle[3][3][0],140,160,50,50,white,puzzle[3][3][2])
button(puzzle[3][4][0],195,160,50,50,white,puzzle[3][4][2])
button(puzzle[3][5][0],250,160,50,50,white,puzzle[3][5][2])
button(puzzle[3][6][0],305,160,50,50,white,puzzle[3][6][2])
button(puzzle[3][7][0],360,160,50,50,white,puzzle[3][7][2])
button(puzzle[3][8][0],415,160,50,50,white,puzzle[3][8][2])
button(puzzle[3][9][0],470,160,50,50,white,puzzle[3][9][2])
# ROW 4
button(puzzle[4][1][0],30,215,50,50,white,puzzle[4][1][2])
button(puzzle[4][2][0],85,215,50,50,white,puzzle[4][2][2])
button(puzzle[4][3][0],140,215,50,50,white,puzzle[4][3][2])
button(puzzle[4][4][0],195,215,50,50,white,puzzle[4][4][2])
button(puzzle[4][5][0],250,215,50,50,white,puzzle[4][5][2])
button(puzzle[4][6][0],305,215,50,50,white,puzzle[4][6][2])
button(puzzle[4][7][0],360,215,50,50,white,puzzle[4][7][2])
button(puzzle[4][8][0],415,215,50,50,white,puzzle[4][8][2])
button(puzzle[4][9][0],470,215,50,50,white,puzzle[4][9][2])
# ROW 5
button(puzzle[5][1][0],30,270,50,50,white,puzzle[5][1][2])
button(puzzle[5][2][0],85,270,50,50,white,puzzle[5][2][2])
button(puzzle[5][3][0],140,270,50,50,white,puzzle[5][3][2])
button(puzzle[5][4][0],195,270,50,50,white,puzzle[5][4][2])
button(puzzle[5][5][0],250,270,50,50,white,puzzle[5][5][2])
button(puzzle[5][6][0],305,270,50,50,white,puzzle[5][6][2])
button(puzzle[5][7][0],360,270,50,50,white,puzzle[5][7][2])
button(puzzle[5][8][0],415,270,50,50,white,puzzle[5][8][2])
button(puzzle[5][9][0],470,270,50,50,white,puzzle[5][9][2])
# ROW 6
button(puzzle[6][1][0],30,325,50,50,white,puzzle[6][1][2])
button(puzzle[6][2][0],85,325,50,50,white,puzzle[6][2][2])
button(puzzle[6][3][0],140,325,50,50,white,puzzle[6][3][2])
button(puzzle[6][4][0],195,325,50,50,white,puzzle[6][4][2])
button(puzzle[6][5][0],250,325,50,50,white,puzzle[6][5][2])
button(puzzle[6][6][0],305,325,50,50,white,puzzle[6][6][2])
button(puzzle[6][7][0],360,325,50,50,white,puzzle[6][7][2])
button(puzzle[6][8][0],415,325,50,50,white,puzzle[6][8][2])
button(puzzle[6][9][0],470,325,50,50,white,puzzle[6][9][2])
# ROW 7
button(puzzle[7][1][0],30,380,50,50,white,puzzle[7][1][2])
button(puzzle[7][2][0],85,380,50,50,white,puzzle[7][2][2])
```

```
        button(puzzle[7][3][0],140,380,50,50,white,puzzle[7][3][2])
        button(puzzle[7][4][0],195,380,50,50,white,puzzle[7][4][2])
        button(puzzle[7][5][0],250,380,50,50,white,puzzle[7][5][2])
        button(puzzle[7][6][0],305,380,50,50,white,puzzle[7][6][2])
        button(puzzle[7][7][0],360,380,50,50,white,puzzle[7][7][2])
        button(puzzle[7][8][0],415,380,50,50,white,puzzle[7][8][2])
        button(puzzle[7][9][0],470,380,50,50,white,puzzle[7][9][2])
        # ROW 8
        button(puzzle[8][1][0],30,435,50,50,white,puzzle[8][1][2])
        button(puzzle[8][2][0],85,435,50,50,white,puzzle[8][2][2])
        button(puzzle[8][3][0],140,435,50,50,white,puzzle[8][3][2])
        button(puzzle[8][4][0],195,435,50,50,white,puzzle[8][4][2])
        button(puzzle[8][5][0],250,435,50,50,white,puzzle[8][5][2])
        button(puzzle[8][6][0],305,435,50,50,white,puzzle[8][6][2])
        button(puzzle[8][7][0],360,435,50,50,white,puzzle[8][7][2])
        button(puzzle[8][8][0],415,435,50,50,white,puzzle[8][8][2])
        button(puzzle[8][9][0],470,435,50,50,white,puzzle[8][9][2])
        # ROW 9
        button(puzzle[9][1][0],30,490,50,50,white,puzzle[9][1][2])
        button(puzzle[9][2][0],85,490,50,50,white,puzzle[9][2][2])
        button(puzzle[9][3][0],140,490,50,50,white,puzzle[9][3][2])
        button(puzzle[9][4][0],195,490,50,50,white,puzzle[9][4][2])
        button(puzzle[9][5][0],250,490,50,50,white,puzzle[9][5][2])
        button(puzzle[9][6][0],305,490,50,50,white,puzzle[9][6][2])
        button(puzzle[9][7][0],360,490,50,50,white,puzzle[9][7][2])
        button(puzzle[9][8][0],415,490,50,50,white,puzzle[9][8][2])
        button(puzzle[9][9][0],470,490,50,50,white,puzzle[9][9][2])

        pygame.draw.line(screen,black,(192,50),(192,540),5)
        pygame.draw.line(screen,black,(357,50),(357,540),5)
        pygame.draw.line(screen,black,(30,212),(520,212),5)
        pygame.draw.line(screen,black,(30,377),(520,377),5)

        button("Back",90,550,150,50,white,gray)

        pygame.display.flip()
        clock2.tick(60)

# when hard button is pressed
if 200+150 > mouse[0] > 200 and 300+50 > mouse[1] > 300 and click[0] == 1:
    hard = True

    puzzle = generatePuzzle("hard")
    time = pygame.time.get_ticks()
```

```
            while hard:
              for event in pygame.event.get():
                if event.type == pygame.QUIT:
                  hard = False
                  timedOps = False
                  done = True
                mouse = pygame.mouse.get_pos()
                x = mouse[0]
                y = mouse[1]
                click = pygame.mouse.get_pressed()
                one = pygame.key.get_pressed()[pygame.K_1]
                two = pygame.key.get_pressed()[pygame.K_2]
                three = pygame.key.get_pressed()[pygame.K_3]
                four = pygame.key.get_pressed()[pygame.K_4]
                five = pygame.key.get_pressed()[pygame.K_5]
                six = pygame.key.get_pressed()[pygame.K_6]
                seven = pygame.key.get_pressed()[pygame.K_7]
                eight = pygame.key.get_pressed()[pygame.K_8]
                nine = pygame.key.get_pressed()[pygame.K_9]
                delete = pygame.key.get_pressed()[pygame.K_BACKSPACE]
                hint = pygame.key.get_pressed()[pygame.K_h]

                # when back key is pressed
                if 90+150 > mouse[0] > 90 and 550+50 > mouse[1] > 550 and click[0] == 1:
                  hard = False

                # when play again button is pressed
                if 310+150 > x > 310 and 550+50 > y > 550 and click[0] == 1 and
solCheck(puzzle):
                  hard = False

                # if a number key, backspace, or h is pressed then take the appropriate action
for the
                # box selected
                if one == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                  puzzle[(y-50)/55+1][(x-30)/55+1][0] = "1"

                elif two == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                  puzzle[(y-50)/55+1][(x-30)/55+1][0] = "2"

                elif three == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
                  puzzle[(y-50)/55+1][(x-30)/55+1][0] = "3"
```

```
    elif four == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "4"

    elif five == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "5"

    elif six == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "6"

    elif seven == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "7"

    elif eight == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "8"

    elif nine == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = "9"

    elif delete == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = ""

    elif hint == 1 and puzzle[(y-50)/55+1][(x-30)/55+1][2] != white:
        puzzle[(y-50)/55+1][(x-30)/55+1][0] = puzzle[(y-50)/55+1][(x-30)/55+1][1]

screen.blit(defimg,rect)

midText = pygame.font.Font("freesansbold.ttf",20)

# display these if the game has been won
if solCheck(puzzle):
    textSurf, textRect = text_objects("Congratulations! You won!", midText)
    textRect.center = (size[0]/2,25)
    screen.blit(textSurf, textRect)
    button("Play Again",310,550,150,50,white,gray)
    button(str(timer/1000),245,550,60,50,white,white)
    if timer/1000 < int(hardHigh):
        hardHigh = str(timer/1000)
# display these if the game is still continuing
else:
    textSurf, textRect = text_objects("Hard", midText)
    textRect.center = (size[0]/2,25)
    screen.blit(textSurf, textRect)
    timer = (pygame.time.get_ticks())-time
    button(str(timer/1000),245,550,60,50,white,white)
```

```
# display box code goes here
# ROW 1
button(puzzle[1][1][0],30,50,50,50,white,puzzle[1][1][2])
button(puzzle[1][2][0],85,50,50,50,white,puzzle[1][2][2])
button(puzzle[1][3][0],140,50,50,50,white,puzzle[1][3][2])
button(puzzle[1][4][0],195,50,50,50,white,puzzle[1][4][2])
button(puzzle[1][5][0],250,50,50,50,white,puzzle[1][5][2])
button(puzzle[1][6][0],305,50,50,50,white,puzzle[1][6][2])
button(puzzle[1][7][0],360,50,50,50,white,puzzle[1][7][2])
button(puzzle[1][8][0],415,50,50,50,white,puzzle[1][8][2])
button(puzzle[1][9][0],470,50,50,50,white,puzzle[1][9][2])
# ROW 2
button(puzzle[2][1][0],30,105,50,50,white,puzzle[2][1][2])
button(puzzle[2][2][0],85,105,50,50,white,puzzle[2][2][2])
button(puzzle[2][3][0],140,105,50,50,white,puzzle[2][3][2])
button(puzzle[2][4][0],195,105,50,50,white,puzzle[2][4][2])
button(puzzle[2][5][0],250,105,50,50,white,puzzle[2][5][2])
button(puzzle[2][6][0],305,105,50,50,white,puzzle[2][6][2])
button(puzzle[2][7][0],360,105,50,50,white,puzzle[2][7][2])
button(puzzle[2][8][0],415,105,50,50,white,puzzle[2][8][2])
button(puzzle[2][9][0],470,105,50,50,white,puzzle[2][9][2])
# ROW 3
button(puzzle[3][1][0],30,160,50,50,white,puzzle[3][1][2])
button(puzzle[3][2][0],85,160,50,50,white,puzzle[3][2][2])
button(puzzle[3][3][0],140,160,50,50,white,puzzle[3][3][2])
button(puzzle[3][4][0],195,160,50,50,white,puzzle[3][4][2])
button(puzzle[3][5][0],250,160,50,50,white,puzzle[3][5][2])
button(puzzle[3][6][0],305,160,50,50,white,puzzle[3][6][2])
button(puzzle[3][7][0],360,160,50,50,white,puzzle[3][7][2])
button(puzzle[3][8][0],415,160,50,50,white,puzzle[3][8][2])
button(puzzle[3][9][0],470,160,50,50,white,puzzle[3][9][2])
# ROW 4
button(puzzle[4][1][0],30,215,50,50,white,puzzle[4][1][2])
button(puzzle[4][2][0],85,215,50,50,white,puzzle[4][2][2])
button(puzzle[4][3][0],140,215,50,50,white,puzzle[4][3][2])
button(puzzle[4][4][0],195,215,50,50,white,puzzle[4][4][2])
button(puzzle[4][5][0],250,215,50,50,white,puzzle[4][5][2])
button(puzzle[4][6][0],305,215,50,50,white,puzzle[4][6][2])
button(puzzle[4][7][0],360,215,50,50,white,puzzle[4][7][2])
button(puzzle[4][8][0],415,215,50,50,white,puzzle[4][8][2])
button(puzzle[4][9][0],470,215,50,50,white,puzzle[4][9][2])
# ROW 5
button(puzzle[5][1][0],30,270,50,50,white,puzzle[5][1][2])
button(puzzle[5][2][0],85,270,50,50,white,puzzle[5][2][2])
button(puzzle[5][3][0],140,270,50,50,white,puzzle[5][3][2])
button(puzzle[5][4][0],195,270,50,50,white,puzzle[5][4][2])
button(puzzle[5][5][0],250,270,50,50,white,puzzle[5][5][2])
button(puzzle[5][6][0],305,270,50,50,white,puzzle[5][6][2])
button(puzzle[5][7][0],360,270,50,50,white,puzzle[5][7][2])
button(puzzle[5][8][0],415,270,50,50,white,puzzle[5][8][2])
button(puzzle[5][9][0],470,270,50,50,white,puzzle[5][9][2])
# ROW 6
button(puzzle[6][1][0],30,325,50,50,white,puzzle[6][1][2])
button(puzzle[6][2][0],85,325,50,50,white,puzzle[6][2][2])
button(puzzle[6][3][0],140,325,50,50,white,puzzle[6][3][2])
button(puzzle[6][4][0],195,325,50,50,white,puzzle[6][4][2])
button(puzzle[6][5][0],250,325,50,50,white,puzzle[6][5][2])
button(puzzle[6][6][0],305,325,50,50,white,puzzle[6][6][2])
button(puzzle[6][7][0],360,325,50,50,white,puzzle[6][7][2])
button(puzzle[6][8][0],415,325,50,50,white,puzzle[6][8][2])
button(puzzle[6][9][0],470,325,50,50,white,puzzle[6][9][2])
# ROW 7
button(puzzle[7][1][0],30,380,50,50,white,puzzle[7][1][2])
button(puzzle[7][2][0],85,380,50,50,white,puzzle[7][2][2])
button(puzzle[7][3][0],140,380,50,50,white,puzzle[7][3][2])
button(puzzle[7][4][0],195,380,50,50,white,puzzle[7][4][2])
button(puzzle[7][5][0],250,380,50,50,white,puzzle[7][5][2])
button(puzzle[7][6][0],305,380,50,50,white,puzzle[7][6][2])
button(puzzle[7][7][0],360,380,50,50,white,puzzle[7][7][2])
button(puzzle[7][8][0],415,380,50,50,white,puzzle[7][8][2])
button(puzzle[7][9][0],470,380,50,50,white,puzzle[7][9][2])
# ROW 8
button(puzzle[8][1][0],30,435,50,50,white,puzzle[8][1][2])
button(puzzle[8][2][0],85,435,50,50,white,puzzle[8][2][2])
button(puzzle[8][3][0],140,435,50,50,white,puzzle[8][3][2])
button(puzzle[8][4][0],195,435,50,50,white,puzzle[8][4][2])
button(puzzle[8][5][0],250,435,50,50,white,puzzle[8][5][2])
button(puzzle[8][6][0],305,435,50,50,white,puzzle[8][6][2])
button(puzzle[8][7][0],360,435,50,50,white,puzzle[8][7][2])
button(puzzle[8][8][0],415,435,50,50,white,puzzle[8][8][2])
button(puzzle[8][9][0],470,435,50,50,white,puzzle[8][9][2])
# ROW 9
button(puzzle[9][1][0],30,490,50,50,white,puzzle[9][1][2])
button(puzzle[9][2][0],85,490,50,50,white,puzzle[9][2][2])
button(puzzle[9][3][0],140,490,50,50,white,puzzle[9][3][2])
button(puzzle[9][4][0],195,490,50,50,white,puzzle[9][4][2])
button(puzzle[9][5][0],250,490,50,50,white,puzzle[9][5][2])
```

```
                    button(puzzle[9][6][0],305,490,50,50,white,puzzle[9][6][2])
                    button(puzzle[9][7][0],360,490,50,50,white,puzzle[9][7][2])
                    button(puzzle[9][8][0],415,490,50,50,white,puzzle[9][8][2])
                    button(puzzle[9][9][0],470,490,50,50,white,puzzle[9][9][2])

                    pygame.draw.line(screen,black,(192,50),(192,540),5)
                    pygame.draw.line(screen,black,(357,50),(357,540),5)
                    pygame.draw.line(screen,black,(30,212),(520,212),5)
                    pygame.draw.line(screen,black,(30,377),(520,377),5)

                    button("Back",90,550,150,50,white,gray)

                    pygame.display.flip()
                    clock.tick(50)

        # display text
        screen.blit(defimg,rect)
        midText = pygame.font.Font("freesansbold.ttf",60)
        textSurf, textRect = text_objects("Sudoku", midText)
        textRect.center = (size[0]/2,(size[1]/2)-200)
        screen.blit(textSurf, textRect)

        # menu options
        button("Easy",200,160,150,50,white,gray)
        button("Medium",200,230,150,50,white,gray)
        button("Hard",200,300,150,50,white,gray)
        button("Back",200,370,150,50,white,gray)

        pygame.display.flip()
        clock3.tick(60)

# when the instructions button is pressed
if 300+150 > mouse[0] > 300 and 160+50 > mouse[1] > 160 and click[0] == 1:
    inst = True
    while inst:

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                inst = False
                done = True
            mouse = pygame.mouse.get_pos()
            click = pygame.mouse.get_pressed()

            # when back key is pressed
```

```
          if 200+150 > mouse[0] > 200 and 450+50 > mouse[1] > 450 and click[0] == 1:
            inst = False

        screen.blit(defimg,rect)

        # display text
        midText = pygame.font.Font("freesansbold.ttf",60)
        textSurf, textRect = text_objects("Sudoku", midText)
        textRect.center = (size[0]/2,(size[1]/2)-200)
        screen.blit(textSurf, textRect)

        # list all the instructions
        midText = pygame.font.Font("freesansbold.ttf",15)
        textSurf, textRect = text_objects("Sudoku is a number puzzle where the 9x9 board is
divided",midText)
        textRect.center = (size[0]/2,(size[1]/2)-150)
        screen.blit(textSurf, textRect)
        textSurf, textRect = text_objects("into smaller 3x3 subgrids. The goal of the game is to
fill",midText)
        textRect.center = (size[0]/2,(size[1]/2)-120)
        screen.blit(textSurf, textRect)
        textSurf, textRect = text_objects("in the board so that the numbers 1-9 are all
represented in",midText)
        textRect.center = (size[0]/2,(size[1]/2)-90)
        screen.blit(textSurf, textRect)
        textSurf, textRect = text_objects("every row, column, and 3x3 grid, thus there can be
no repeated",midText)
        textRect.center = (size[0]/2,(size[1]/2)-60)
        screen.blit(textSurf, textRect)
        textSurf, textRect = text_objects("digits. Each game begins with a partially completed
board. In",midText)
        textRect.center = (size[0]/2,(size[1]/2)-30)
        screen.blit(textSurf, textRect)
        textSurf, textRect = text_objects("order to fill in the rest of the board, hover over the
chosen",midText)
        textRect.center = (size[0]/2,(size[1]/2))
        screen.blit(textSurf, textRect)
        textSurf, textRect = text_objects("box and enter the number that belongs there. If help
is needed,",midText)
        textRect.center = (size[0]/2,(size[1]/2)+30)
        screen.blit(textSurf, textRect)
        textSurf, textRect = text_objects("press h and the correct number for the chosen box
will be revealed.",midText)
        textRect.center = (size[0]/2,(size[1]/2)+60)
```

```
                screen.blit(textSurf, textRect)
                textSurf, textRect = text_objects("Once the puzzle has been correctly solved, the game
will",midText)
                textRect.center = (size[0]/2,(size[1]/2)+90)
                screen.blit(textSurf, textRect)
                textSurf, textRect = text_objects("congratulate you!", midText)
                textRect.center = (size[0]/2,(size[1]/2)+120)
                screen.blit(textSurf, textRect)

                button("Back",200,450,150,50,white,gray)

                pygame.display.flip()

                clock.tick(60)

        # when the high scores button is pressed
        if 300+150 > mouse[0] > 300 and 230+50 > mouse[1] > 230 and click[0] == 1:
            scores = True
            while scores:

                for event in pygame.event.get():
                    if event.type == pygame.QUIT:
                        scores = False
                        done = True
                    mouse = pygame.mouse.get_pos()
                    click = pygame.mouse.get_pressed()

                    # when back key is pressed
                    if 200+150 > mouse[0] > 200 and 450+50 > mouse[1] > 450 and click[0] == 1:
                        scores = False

                screen.blit(defimg,rect)

                # list all the high scores
                midText = pygame.font.Font("freesansbold.ttf",60)
                textSurf, textRect = text_objects("Sudoku", midText)
                textRect.center = (size[0]/2,(size[1]/2)-200)
                screen.blit(textSurf, textRect)
                midText = pygame.font.Font("freesansbold.ttf",50)
                textSurf, textRect = text_objects("High Scores", midText)
                textRect.center = (size[0]/2,(size[1]/2)-125)
                screen.blit(textSurf, textRect)
                midText = pygame.font.Font("freesansbold.ttf",50)
                textSurf, textRect = text_objects("Easy: "+easyHigh+" seconds", midText)
                textRect.center = (size[0]/2,(size[1]/2)-75)
                screen.blit(textSurf, textRect)
                midText = pygame.font.Font("freesansbold.ttf",50)
                textSurf, textRect = text_objects("Medium: "+mediumHigh+" seconds", midText)
                textRect.center = (size[0]/2,(size[1]/2)-25)
                screen.blit(textSurf, textRect)
                midText = pygame.font.Font("freesansbold.ttf",50)
                textSurf, textRect = text_objects("Hard: "+hardHigh+" seconds", midText)
                textRect.center = (size[0]/2,(size[1]/2)+25)
                screen.blit(textSurf, textRect)

                button("Back",200,450,150,50,white,gray)

                pygame.display.flip()

                clock.tick(60)

    # puts the background image
    screen.blit(defimg,rect)

    # display text
    midText = pygame.font.Font("freesansbold.ttf",60)
    textSurf, textRect = text_objects("Sudoku", midText)
    textRect.center = (size[0]/2,(size[1]/2)-200)
    screen.blit(textSurf, textRect)

    # main menu options
    button("Free Play",100,160,150,50,white,gray)
    button("Timed Play",100,230,150,50,white,gray)
    button("Instructions",300,160,150,50,white,gray)
    button("High Scores",300,230,150,50,white,gray)
    button("Exit",200,300,150,50,white,gray)


    # update the screen
    pygame.display.flip()

    clock.tick(60)

# Close the window and quit.
pygame.quit()
```