getRandomizedBoard funckijata imase gresno ime , odnono faleshe karakterot d :D

```
def getRandomizedBoard():
    # Get a list of every possible shape in every possible color.
    icons = []
    for color in ALLCOLORS:
        for shape in ALLSHAPES:
            icons.append( (shape, color) )

    random.shuffle(icons) # randomize the order of the icons list
    numIconsUsed = int(BOARDWIDTH * BOARDHEIGHT / 2) # calculate how many icons are needed
    icons = icons[:numIconsUsed] * 2 # make two of each
    random.shuffle(icons)
```

splitIntoGroups funkcijata ne vrakase nikakov statement , odnosno ne mozeshe da se povika vo startGameAnimation funkcijata

```
def splitIntoGroupsOf(groupSize, theList):
    # splits a list into a list of lists, where the inner lists have at
    # most groupSize number of items.
    result = []
    for i in range(0, len(theList), groupSize):
        result.append(theList[i:i + groupSize])
    return result
```

falese get box at pixel funkcijata za x I y kordinatite kade shto ni e postaven mausot

```
                mouseClicked = True

        boxx, boxy = getBoxAtPixel(mousex, mousey)
        if boxx != None and boxy != None:
            # The mouse is currently over a box.
            if not revealedBoxes[boxx][boxy]:
                drawHighlightBox(boxx, boxy)
            if not revealedBoxes[boxx][boxy] and mouseClicked:
                revealBoxesAnimation(mainBoard, [(boxx, boxy)])
                revealedBoxes[boxx][boxy] = True # set the box as "revealed"
```

falese display update funkcijata koja shto ni ovozmozuva osvezuvanje na ekranot

```
                startGameAnimation(mainBoard)
            firstSelection = None # reset firstSelection variable

        # Redraw the screen and wait a clock tick.
        pygame.display.update() # falese display update funkcijata koja shto ni ovozmozuva osvezuvanje na ekranot
        FPSCLOCK.tick(FPS)
```

ne bea vo zagrada boxsize i gapsize , pa bi dobile drugi kordinati

```
def leftTopCoordsOfBox(boxx, boxy):
    # Convert board coordinates to pixel coordinates
    left = boxx * (BOXSIZE + GAPSIZE) + XMARGIN
    top = boxy * (BOXSIZE + GAPSIZE) + YMARGIN ne bea vo zagrada boxsize i gapsize , pa bi dobile drugi kordinati
    return (left, top)
```

1/4 treba da dobieme od golemina na box-ot , ne bese mnozeno po 0.25 so toa so bi ja imale cela

```
# 1/4 treba da dobieme od golemina na box-ot , ne bese mnozeno po 0.25 so toa so bi ja imale cela
def drawIcon(shape, color, boxx, boxy):
    quarter = int(BOXSIZE) * 0.25 # syntactic sugar #
    half = int(BOXSIZE * 0.5)  # syntactic sugar
```

1 vo 0 , 0 vo 1 , shape vo color

```
def getShapeAndColor(board, boxx, boxy):
    # shape value for x, y spot is stored in board[x][y][0]
    # color value for x, y spot is stored in board[x][y][1]
    return board[boxx][boxy][0], board[boxx][boxy][1]# 1 vo 0 , 0 vo 1 , shape vo color
```

falea ushte 2 boi koi se potrebni za animacijata koja menja 4 boi , i flagot da se postavi na true za da se menuvaat boite

```
def gameWonAnimation(board):
    # flash the background color when the player has won
    coveredBoxes = generateRevealedBoxesData(True)
    color1 = LIGHTBGCOLOR
    color2 = BGCOLOR
    color3 = CYAN
    color4 = PURPLE
    flag = 1

    #falea ushte 2 boi koi se potrebni za animacijata koja menja 4 boi , i flagot da se postavi na true za da se menuvaat boite
```

animacija za pobeda na igrata i promenite na bojata so pomos na flagot nedostasuvaa

```
    for i in range(13):
        if flag == 1:
            color1, color2 = color2, color1# swap colors
            flag = 0
        if flag == 0:
            color1, color2 = color3, color4
            flag = 1
        DISPLAYSURF.fill(color1)
        drawBoard(board, coveredBoxes)
        pygame.display.update()
        pygame.time.wait(300)

        #animacija za pobeda na igrata i promenite na bojata so pomos na flagot nedostasuvaa
```

dodadena opcijata za izbor na slika + promena vo brojot na boi pred da se prikazhe slikata + 60FPS od 30FPS

```
color5 = GREEN
color6 = BLUE
flag = 1

winPicture = pygame.image.load('1.png')  # birame koja slika da se vcita
winX = 10  # kordinati na X
winY = 10  # kordinati na Y

#falea ushte 2 boi koi se potrebni za animacijata koja menja 4 boi , i flagot da se postavi na true za da se menuvaat boite

for i in range(13):
    if flag == 1:
        color1, color2 = color2, color1 # swap colors
        color5, color6 = color1, color3
        flag = 0
    if flag == 0:
        color1, color2 = color3, color4
        color1, color3, color6, color5
        flag = 1
    DISPLAYSURF.fill(color1)
    drawBoard(board, coveredBoxes)
    pygame.display.update()
    pygame.time.wait(300)

    #animacija za pobeda na igrata i promenite na bojata so pomos na flagot nedostasuvaa

DISPLAYSURF.blit(winPicture, (winX, winY))  # funkcija za prikaz na slikata , odkoga ke se naprai cela win animacija ke izleze slikata.

pygame.display.update()  # osvezhuvanje na ekranot so poslednite promeni
```