Лабораториска вежба 1 – Пнви

Фросина Цветковска 191216

1. Постојат 8 грешки во дадениот код. Обидете се да ги најдете, означете ги, коментирајте ги и потоа напишете го точниот код на тоа место со цел да добиете целосно функционална програма.

1. На линија 87 наместо boxx, boxy = mousex, mousey треба да биде:

boxx, boxy = getBoxAtPixel(mousex, mousey)

```
85                  mouseClicked = True
86              #Bug 1 промена на boxx, boxy = mousex
87              boxx, boxy = getBoxAtPixel(mousex, mousey)
88              if boxx != None and boxy != None:
89                  # The mouse is currently over a box.
```

2. На линија 126 додадено е pygame.display.update() цо цел да работи FPSCLOCK.tick(FPS)

```
123                     firstSelection = None # reset firstSelection variable
124             #Bug 2 Додадено pygame.display.update()
125             # Redraw the screen and wait a clock tick.
126             pygame.display.update()
127             FPSCLOCK.tick(FPS)
```

3. На линија 137 фалеше d во името на функцијата getRandomizedBoard() и со тоа сите повици кои даваа грешка во самиот код се поправени.

```
135
136     #Bug 3 додадено d во името на функцијата
137     def getRandomizedBoard():
138         # Get a list of every possible shape in every possible color.
```

4. Функцијата splitIntoGroupsOf(groupSize, theList) не го враќаше резултатот од поделбата и на линија 166 се додава return result

```
160     def splitIntoGroupsOf(groupSize, theList):
161         # splits a list into a list of lists, where the inner lists have at
162         # most groupSize number of items.
163         result = []
164         for i in range(0, len(theList), groupSize):
165             result.append(theList[i:i + groupSize])
166         return result #Bug  4 додавање на return result за враќање на резултатот
167
```

5. На линиите 171 и 172 недостасуваат загради меѓу BOXSIZE + GAPSIZE така што конвертирањето треба да се реализира на следниот начин:

left = boxx * (BOXSIZE + GAPSIZE) + XMARGIN

top = boxy * (BOXSIZE + GAPSIZE) + YMARGIN

```
168 def leftTopCoordsOfBox(boxx, boxy):
169     # Convert board coordinates to pixel coordinates
170     #Bug 5 додавање на загради помеѓу BOXSIZE + GAPSIZE
171     left = boxx * (BOXSIZE + GAPSIZE) + XMARGIN
172     top = boxy * (BOXSIZE + GAPSIZE) + YMARGIN
173     return (left, top)
174
```

6. Во функцијата drawIcon(shape, color, box, boxy) на линија 188 во делот за пресметување на quarter потребно е да се множи со BOXSIZE со 0.25

```
186 def drawIcon(shape, color, boxx, boxy):
187     #Bug 6 потребно е множење со 0.25 кај quarter
188     quarter = int(BOXSIZE * 0.25) # syntactic sugar
189     half =    int(BOXSIZE * 0.5)  # syntactic sugar
190
```

7.Во функцијата getShapeAndColor() на линија 212 потребно беше да се променат местата на вредностите на бојата и самите објекти.

Пред промените:

return board[boxx][boxy][1], board[boxx][boxy][0]

По промените:

return board[boxx][boxy][0], board[boxx][boxy][1]

```
208 def getShapeAndColor(board, boxx, boxy):
209     # shape value for x, y spot is stored in board[x][y][0]
210     # color value for x, y spot is stored in board[x][y][1]
211     #Bug 7 промена на местата на вредностите на бојата и на вредноста на објектите
212     return board[boxx][boxy][0], board[boxx][boxy][1]
213
```

8. Во функцијата drawBoxCovers() со цел анимациите да се прикажуваат со брзината која што сакаме, деловите:

pygame.display.update()

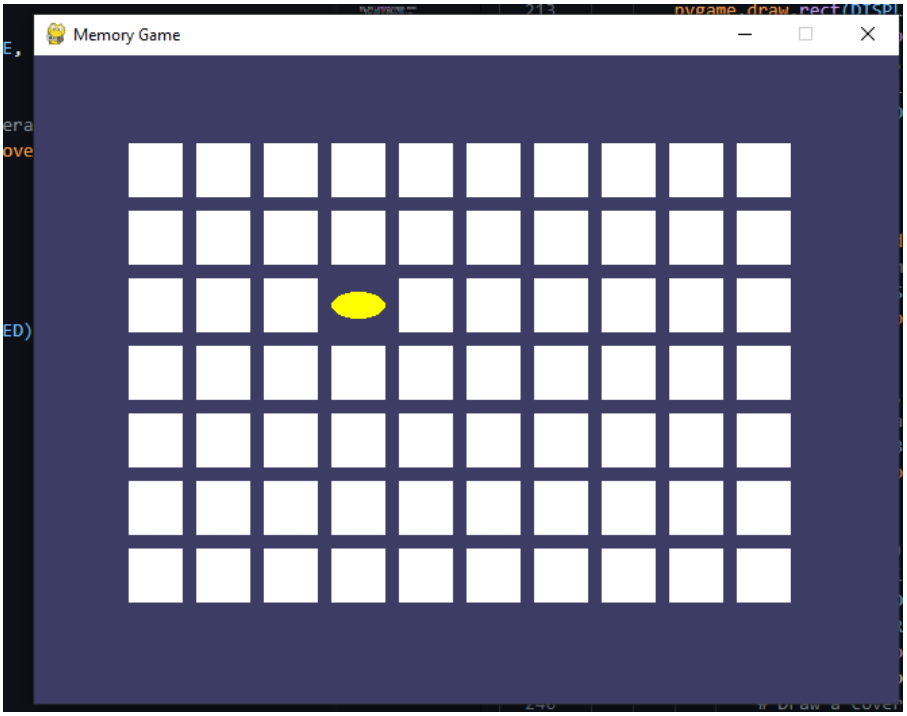
FPSCLOCK.tick(FPS)

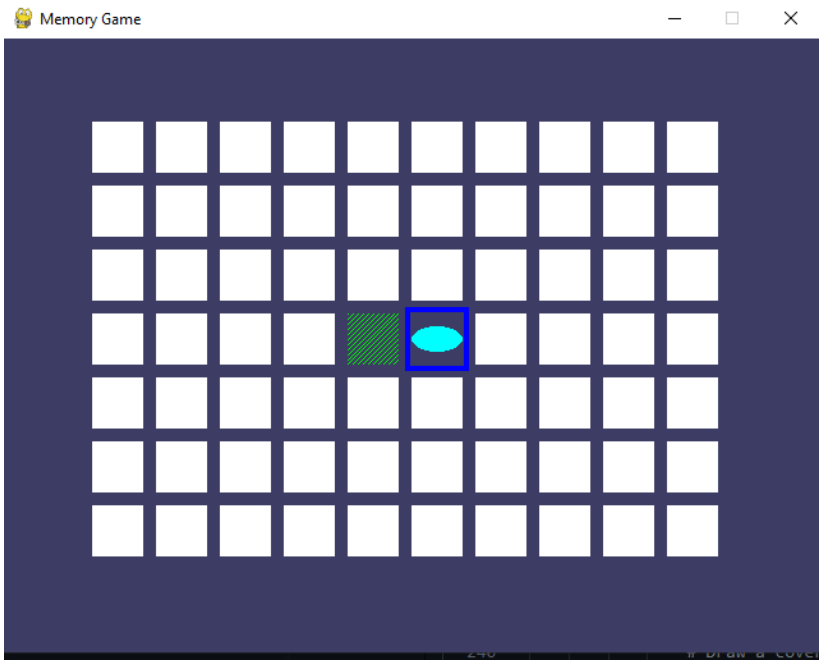
Кои се наоѓаат на линии 226 и 227 ги поместуваме еден tab во лево со цел истите да бидат надвор од самиот циклус и да се извршуваат на ниво на функција.

```
        if coverage > 0: # only draw the cover if there is an coverage
            pygame.draw.rect(DISPLAYSURF, BOXCOLOR, (left, top, coverage, BOXSIZE))
    #Bug 8 Поместување на лево за да се извршат на ниво на функција
    pygame.display.update()
    FPSCLOCK.tick(FPS)

```

Резултатите по поправките во кодот





2. При победа на играчот, наместо постоечката, креирајте анимација така што ќе вметнете слика по ваш избор и ќе обезбедите нејзино движење/трепкање/појавување-исчезнување или друг вид ефект што би се вклопил во сцената.

Во функцијата gameWonAnimation со цел да се илустрира движење на сликата, на линија 285 ја додаваме сликата која што сакаме да ја употребиме. Потоа во самиот for циклус, со цел да го имаме самиот ефект, го додаваме DISPLAYSURF.blit(image, (i*10, i)) на линија 287 и ги бришеме деловите кои ги користевме за додавање на нова позадина која што има само одредена боја

```
273  def gameWonAnimation(board):
274      # flash the background color when the player has won
275      coveredBoxes = generateRevealedBoxesData(True)
276      color1 = LIGHTBGCOLOR
277      color2 = BGCOLOR
278
279      image = pygame.image.load("winner.jpg")
280      for i in range(13):
281          DISPLAYSURF.blit(image, (i*10, i))
282          drawBoard(board, coveredBoxes)
283          pygame.display.update()
284          pygame.time.wait(300)
285
286
```

3. Направете промена по ваш избор, објаснете ја во коментар и имплементирајте ја.

Во функцијата coverBoxesAnimation додадов време на чекање пред извршување на анимациите pygame.time.wait(100) на линија 239 бидејќи самата игра многу брзо ги затвораше кутиите. Со тоа се овозможува пауза пред затворање на коцките. Бидејќи самата табла се состои од повеќе бои и објекти, на играчот ќе му се олесни запомнувањето на објектите и ќе може подобро да се фокусира на играта.

```
236  def coverBoxesAnimation(board, boxesToCover):
237      # Do the "box cover" animation.
238      #Барање број 3
239      pygame.time.wait(100)
240      for coverage in range(0, BOXSIZE + REVEALSPEED, REVEALSPEED):
241          drawBoxCovers(board, boxesToCover, coverage)
242
```