

# Лабораториска вежба Squirrel – ПНВИ

Фросина Цветковска 191216

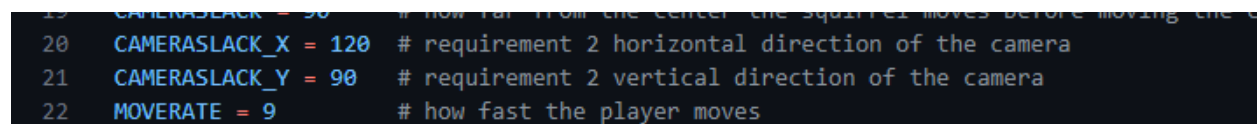
1. Increase the camera area of the game to 960x480 pixels (1 point).

Во ова барање на линија 10 ја зголемив вредноста на WINWIDTH на 960



2. Define two new constants for the CAMERASLACK constants for the horizontal and vertical direction appropriately (1 point).

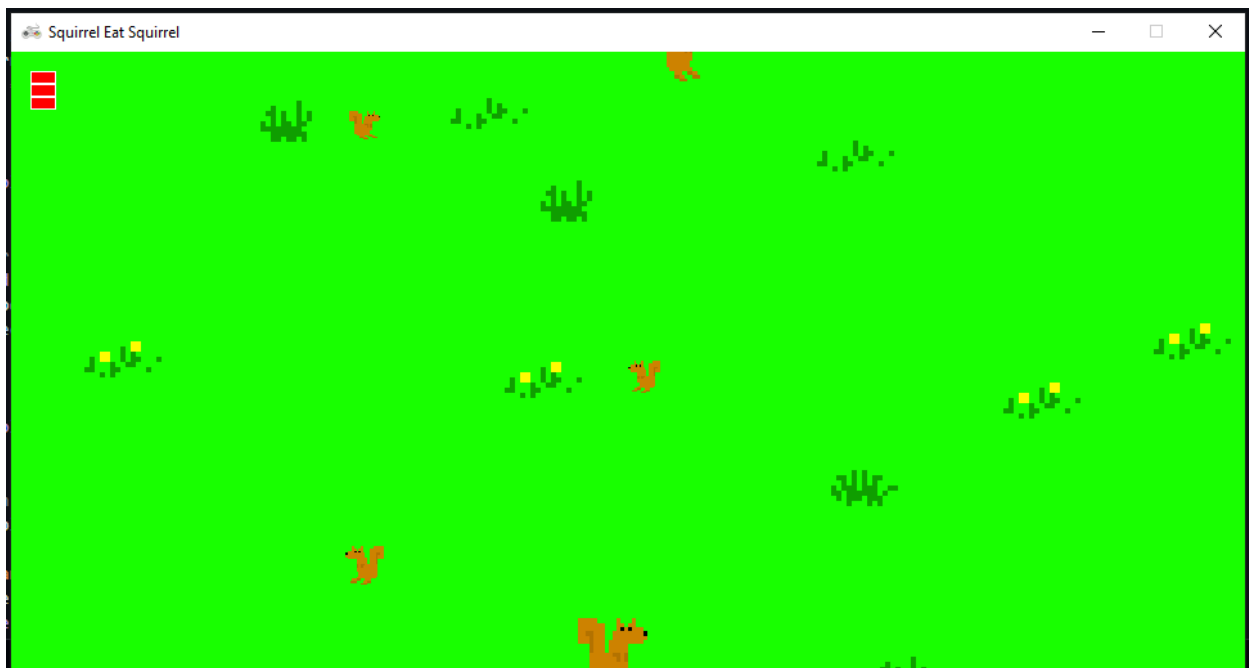
Со цел да имаме две различни вредности за CAMERASLACK креираме две глобални променливи CAMERASLACK\_X и CAMERASLACK\_Y (линии 20 и 21). Потоа за истите промени да се прикажат, во самиот main на линиите 182 – 189 ги менуваме CAMERASLACK со соодветните вредности за x и y кои дозволуваат промена на границите кога се следи играчот.



```

177
178 #requirement 2: change of CAMERASLACK to CAMERASLACK_X and CAMERASLACK_Y:
179 # adjust camerax and cameray if beyond the "camera slack"
180 playerCenterx = playerObj['x'] + int(playerObj['size'] / 2)
181 playerCentery = playerObj['y'] + int(playerObj['size'] / 2)
182 if (camerax + HALF_WINWIDTH) - playerCenterx > CAMERASLACK_X:
183     camerax = playerCenterx + CAMERASLACK_X - HALF_WINWIDTH
184 elif playerCenterx - (camerax + HALF_WINWIDTH) > CAMERASLACK_X:
185     camerax = playerCenterx - CAMERASLACK_X - HALF_WINWIDTH
186 if (cameray + HALF_WINHEIGHT) - playerCentery > CAMERASLACK_Y:
187     cameray = playerCentery + CAMERASLACK_Y - HALF_WINHEIGHT
188 elif playerCentery - (cameray + HALF_WINHEIGHT) > CAMERASLACK_Y:
189     cameray = playerCentery - CAMERASLACK_Y - HALF_WINHEIGHT
190

```



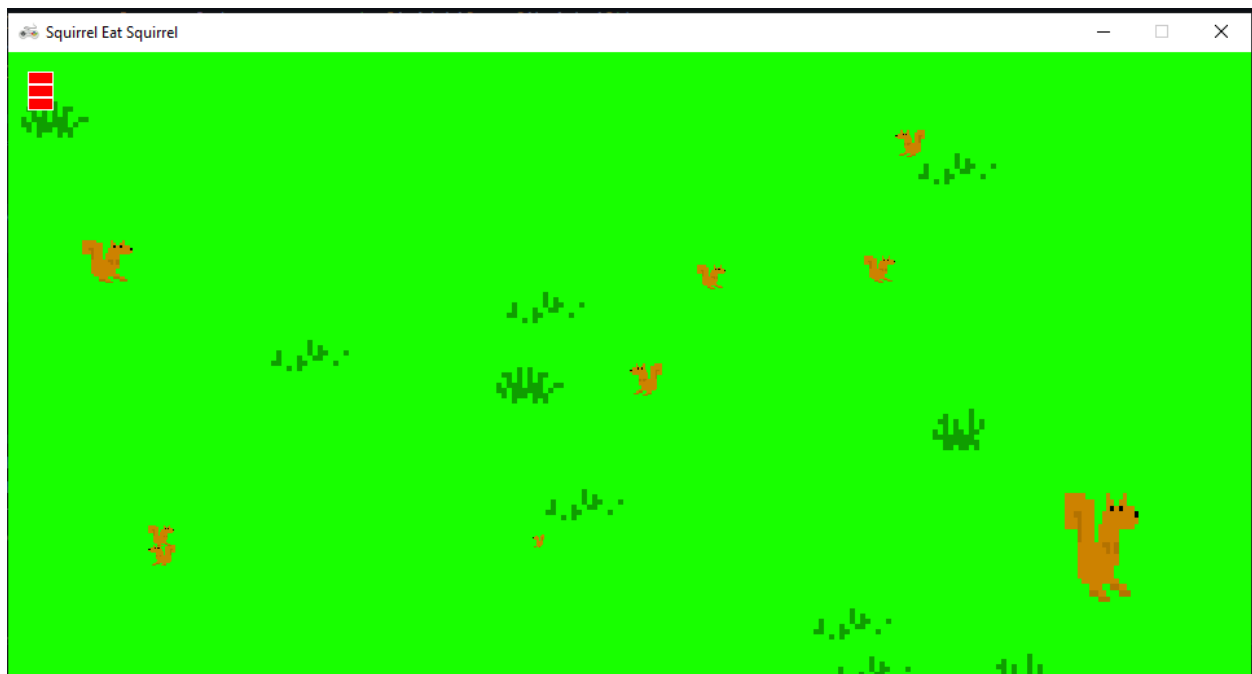
3. Introduce the possibility an enemy squirrels to bounce downwards (the squirrel could bounce only in one direction only upwards or only downwards). The direction is determined when creating an enemy squirrel. (3 points)

При креирање на нова верверичка со помош на методот `makeNewSquirrel()` секоја нова ќе се движи во насока доле (0) или горе (1) за случајна вредност меѓу 10 и 50 или -50 до -10 во зависност од насоката. (линии 405 -409 плус коментар на линија 403 со цел промена на насоките)

```

400     sq['surface'] = pygame.transform.scale(K_SQUIR_IMG, (sq['width'], sq['height']))
401     sq['bounce'] = 0
402     sq['bouncerate'] = random.randint(10, 18)
403     # sq['bounceheight'] = random.randint(10, 50)
404     #requirement 3:
405     direction = [0,1]
406     if random.choice(direction) == 1:
407         sq['bounceheight'] = random.randint(10,50)
408     else:
409         sq['bounceheight'] = random.randint(-50,-10)
410     return sq
411
412

```



4. Change the logic of the game. If the squirrel is hit by the larger animal it becomes smaller using the same logic as for the getting bigger at the current game. The game is over when the squirrel grows to the WINSIZE or LOSTSIZE. (5 points)

Со овие промени при удар со голема верверичка, играчот ќе се намали. Ако големината е под LOSTSIZE која што е на иницијална вредност 15 (линија 30) играчот ќе ја загуби играта. За полесно да се илустрира, црвените health bars се видливи и при удар не се намалува животот. (линии 327 – 330)

```

26 WINSIZE = 300      # how big the player needs to be to win
27 INVULNTIME = 2     # how long the player is invulnerable after being hit in seconds
28 GAMEOVERTIME = 4   # how long the "game over" text stays on the screen in seconds
29 MAXHEALTH = 3       # how much health the player starts with
30 LOSTSIZE = 15       # requirement 4: how big the player need to lose
31

```

```
325         invulnerableStartTime = time.time()
326         #requirement 4
327         playerObj['size'] -= int((sqObj['width'] * sqObj['height']**0.2) + 1)
328         if playerObj['size'] <= LOSTSIZE:
329             gameOverMode = True # turn on "game over mode"
330             gameOverStartTime = time.time()
331     else:
```



5. Which change will you make in the game in order to have more fun? Explain and add it! Comment all added or modified command with '# added or modified to provide functionality' (5 points)

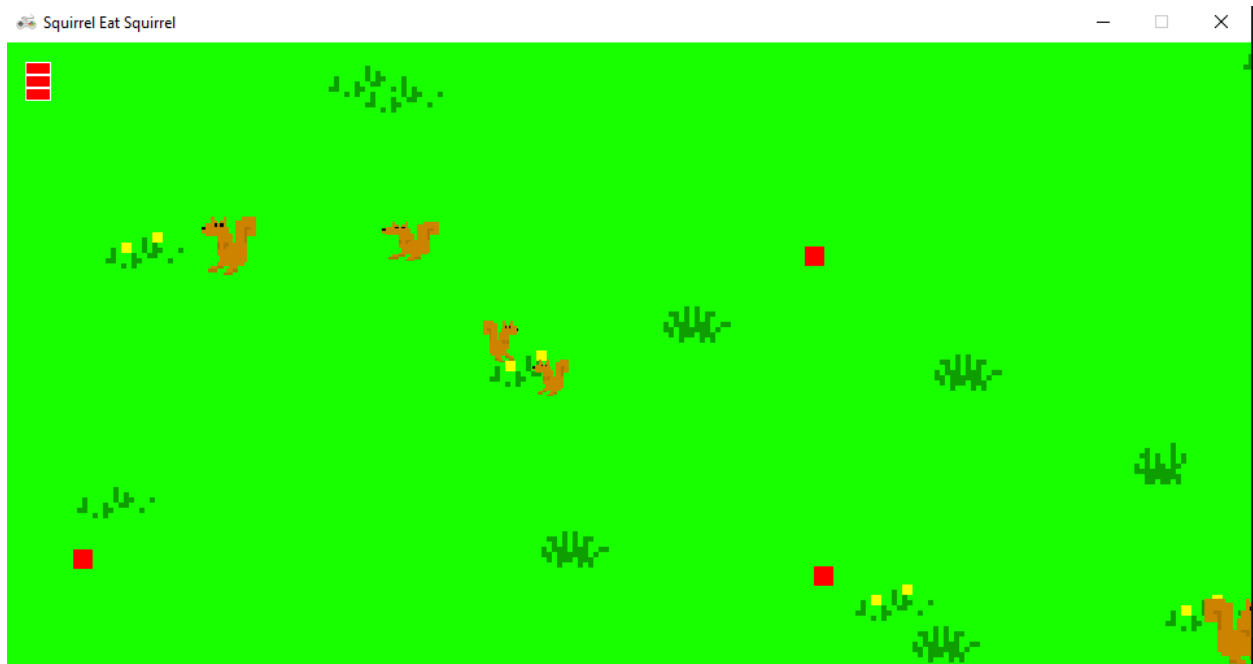
Во ова барање додаваме obstacles во самата игра. Креираме глобални променливи NUM\_OBSTACLES, MAX\_OBSTACLES и OBSTACLES\_SIZE (линии 37-39). Во main делот додаваме во playerObj нов објект за last\_obstacle\_time за да знаеме која последната препрека е креирана. Потоа правиме две промени, додавање на препрека на одредено време (192-198) и проверка дали играчот се судрил со една од препреките и го намалува health bar за 1 (линии 200-204). Во овој случај поради тоа што во 4тото барање при удар не се одзема живот само health bar ќе се намалува за 1. На крајот се исцртува самата препрека. (линии 242 - 244)

```
35 SQUIRRELMAXSPEED = 7 # fastest squirrel speed
36 DIRCHANGEFREQ = 2 # % chance of direction change per frame
37 NUM_OBSTACLES = 5 # requirement 5
38 MAX_OBSTACLES = 15 # requirement 5
39 OBSTACLES_SIZE = 15 # requirement 5
40 LEFT = 'left'
```

```
119 # stores the player object:
120 playerObj = {'surface': pygame.transform.scale(L_SQUIR_IMG, (STARTSIZE, STARTSIZE)),
121             'facing': LEFT,
122             'size': STARTSIZE,
123             'x': HALF_WINWIDTH,
124             'y': HALF_WINHEIGHT,
125             'bounce': 0,
126             'health': MAXHEALTH,
127             'last_obstacle_time': time.time()} #requirement 5
128
129 moveLeft = False
```

```
190
191 #requirement 5: add obstacle after a certain time
192 if time.time() - playerObj['last_obstacle_time'] > random.randint(5, 15):
193     for _ in range(min(NUM_OBSTACLES, MAX_OBSTACLES - len(obstacles))):
194         obstacle = {'x': random.randint(0, WINWIDTH - OBSTACLES_SIZE),
195                   'y': random.randint(0, WINHEIGHT - OBSTACLES_SIZE),
196                   'rect': pygame.Rect(0, 0, OBSTACLES_SIZE, OBSTACLES_SIZE)}
197         obstacles.append(obstacle)
198         playerObj['last_obstacle_time'] = time.time()
199 #requirement 5: check if player collides with any obstacle
200 for obstacle in obstacles:
201     if playerObj['rect'].colliderect(obstacle['rect']):
202         playerObj['health'] -= 1
203         obstacles.remove(obstacle)
204         break
205
```

```
239 drawHealthMeter(playerObj['health'])
240
241 #requirement 5: draw the obstacle
242 for obstacle in obstacles:
243     obstacle['rect'] = pygame.Rect(obstacle['x'] - camerax, obstacle['y'] - camery, OBSTACLES_SIZE, OBSTACLES_SIZE)
244     pygame.draw.rect(DISPLAYSURF, RED, obstacle['rect'])
245
246 for event in pygame.event.get(): # event handling loop
```



All changes should be done in the same source file and at the end should present the single game.