

Задача по ПНВИ

задача 1

“Wormy.py”

- **generateObstacles()** функција

Генерира случајни препреки на таблата(екранот).

Бројот на препреки е од 1 до 5% од вкупниот број на ќелии.

Проверува да не се појават на исто место со телото на црвчето.

```
# Generira prepreki od 1 do 5% od ukupni kelii
def generateObstacles(wormCoords): 1 usage  ▲ DimitarBogoski
    total_cells = CELLWIDTH * CELLHEIGHT
    max_obstacles = max(1, int(total_cells * 0.05))
    num_obstacles = random.randint( 1, max_obstacles)

    obstacles = []
    forbidden = set((seg['x'], seg['y']) for seg in wormCoords)

    while len(obstacles) < num_obstacles:
        x = random.randint( 0, CELLWIDTH - 1)
        y = random.randint( 0, CELLHEIGHT - 1)
        coord = (x, y)
        if coord in forbidden:
            continue
        if coord in [(o['x'], o['y']) for o in obstacles]:
            continue
        obstacles.append({'x': x, 'y': y})
    return obstacles
```

Цел: создава сива „барицада“ која црвчето не смее да ја допре.

- **isObstacleCell()** функција

Проверува дали координатите (x, y) се дел од некоја препрека.

Ако се враќа True, инаку False.

```
# Proverka na preklopuvacka kelija
def isObstacleCell(x, y): 2 usages  ▲ DimitarBogoski
    for obs in OBSTACLES:
        if obs['x'] == x and obs['y'] == y:
            return True
    return False
```

Цел: се користи при движење и при генерирање на јаболката,

за да не се појават објекти врз препрека.

- **willHitObstacle()** функција

Предвремено проверува дали ако црвчето се движи во дадена насока, следната ќелија ќе биде препрека.

```
# Предвремена проверка на колизија
def willHitObstacle(direction, headCoord): 4 usages ▾ DimitarBogojoski
    nextHead = getNextHead(direction, headCoord)
    return isObstacleCell(nextHead['x'], nextHead['y'])
```

Цел: спречува играчот да сврти кон препрека (на пр. не дозволува смена на насока кон ѕид).

- **getNextHead() функција**

Пресметува и враќа координати на следната позиција на главата на црвчето според моменталната насока.

```
# Предвремена проверка за главата
def getNextHead(direction, headCoord): 2 usages ▾ DimitarBogojoski
    if direction == UP:
        return {'x': headCoord['x'], 'y': headCoord['y'] - 1}
    elif direction == DOWN:
        return {'x': headCoord['x'], 'y': headCoord['y'] + 1}
    elif direction == LEFT:
        return {'x': headCoord['x'] - 1, 'y': headCoord['y']}
    elif direction == RIGHT:
        return {'x': headCoord['x'] + 1, 'y': headCoord['y']}
    return dict(headCoord)
```

Цел: централизира логика за движење — се користи и за проверки (дали има препрека) и за вистинско движење на црвчето.

- **drawObstacles() функција**

Ги црта препреките на екранот како сиви квадрати (DARKGRAY).

```
# Ги црта препреките
def drawObstacles(obstacles): 1 usage ▾ DimitarBogojoski
    for coord in obstacles:
        x = coord['x'] * CELLSIZE
        y = coord['y'] * CELLSIZE
        rect = pygame.Rect(x, y, CELLSIZE, CELLSIZE)
        pygame.draw.rect(DISPLAYSURF, DARKGRAY, rect)
```

Цел: визуелно ги прикажува бариерите што ги генерира generateObstacles().

За секој случај ако не се симне програмата како што треба ја имам воспоставено на github:

<https://github.com/DimitarBogojoski/Wormy.git>