



**UNIVERSITY of INFORMATION
TECHNOLOGY and MANAGEMENT**
in Rzeszow, POLAND

C# Programming Project

Project made by:

Azamat Amanzhol w65388

Yeslan Nurzhigitov w65381

Bogdan Valishin w63744

**For Professor
Marcin Jagiela**

Table of contents:

- Team members
- Introduction
- Project Description
- Conclusions

Team members:

Azamat Amanzhol w65388

Yeslan Nurzhigitov w65381

Bogdan Valishin w63744

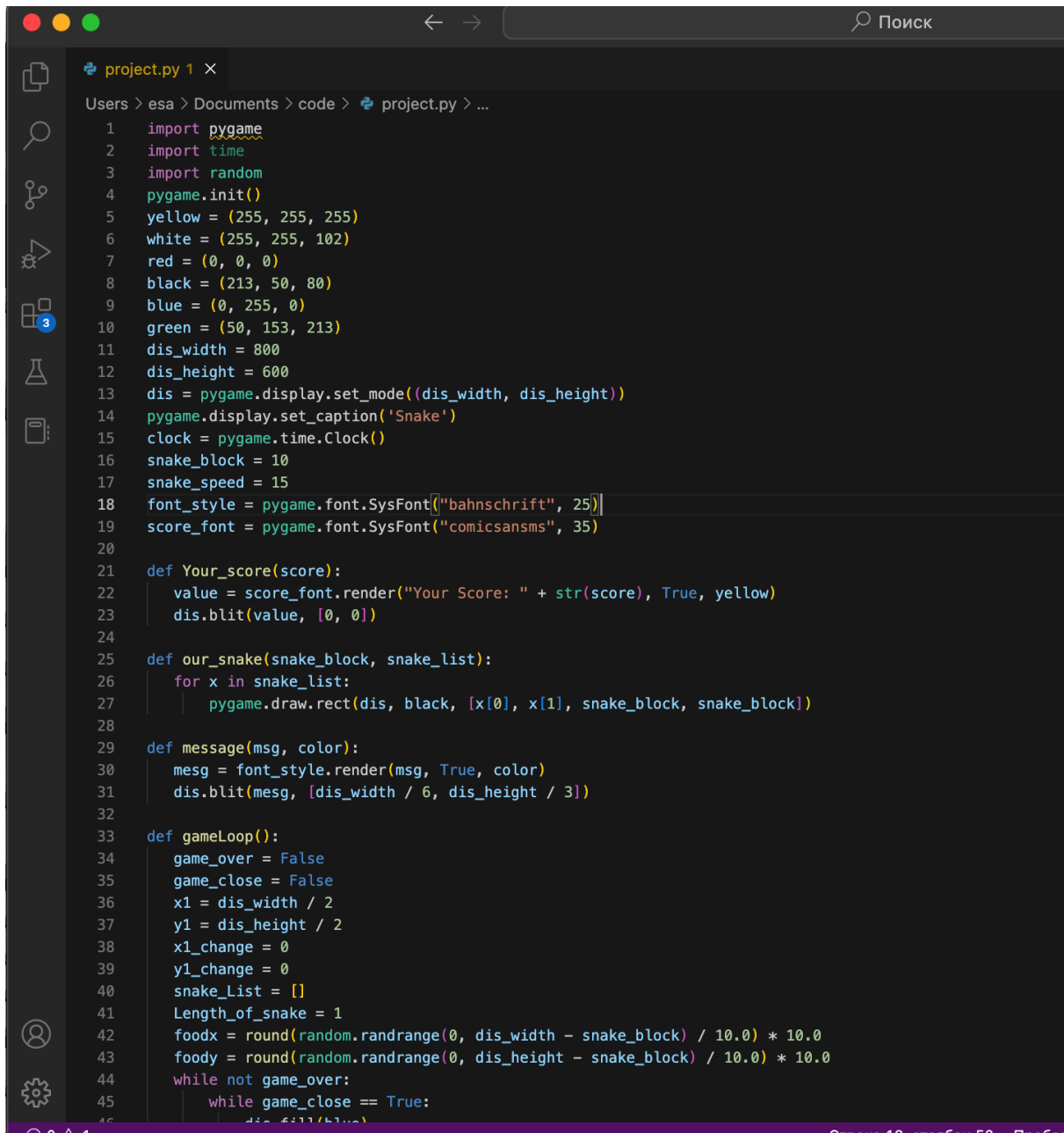
Introduction:

This project is a “Snake” game made by Python language.

Project Description:

We decided to create a “Snake” game in Python. To create the screen using Pygame, we used the *display.set_mode()* function. Also, we used the *init()* and the *quit()* methods to initialize and uninitialize everything at the start and the end of the code. The *update()* method is used to update

any changes made to the screen.

A screenshot of a code editor window titled 'project.py 1'. The editor shows a Python script for a snake game. The script includes imports for pygame, time, and random. It initializes the pygame display with a width of 800 and a height of 600. It defines several color variables: yellow, white, red, black, blue, and green. It also defines the snake block size, snake speed, and font styles. The script includes functions for displaying the score, drawing the snake, displaying messages, and the main game loop. The game loop initializes game variables, sets the initial snake position and direction, and starts the game. The script is written in a dark-themed editor with a sidebar on the left containing icons for file explorer, search, and other tools. The status bar at the bottom indicates 'Строка 18, столбец 52' and 'Python 3.10.11'.

```
1 import pygame
2 import time
3 import random
4 pygame.init()
5 yellow = (255, 255, 255)
6 white = (255, 255, 102)
7 red = (0, 0, 0)
8 black = (213, 50, 80)
9 blue = (0, 255, 0)
10 green = (50, 153, 213)
11 dis_width = 800
12 dis_height = 600
13 dis = pygame.display.set_mode((dis_width, dis_height))
14 pygame.display.set_caption('Snake')
15 clock = pygame.time.Clock()
16 snake_block = 10
17 snake_speed = 15
18 font_style = pygame.font.SysFont("bahnschrift", 25)
19 score_font = pygame.font.SysFont("comicsansms", 35)
20
21 def Your_score(score):
22     value = score_font.render("Your Score: " + str(score), True, yellow)
23     dis.blit(value, [0, 0])
24
25 def our_snake(snake_block, snake_list):
26     for x in snake_list:
27         pygame.draw.rect(dis, black, [x[0], x[1], snake_block, snake_block])
28
29 def message(msg, color):
30     mesg = font_style.render(msg, True, color)
31     dis.blit(mesg, [dis_width / 6, dis_height / 3])
32
33 def gameLoop():
34     game_over = False
35     game_close = False
36     x1 = dis_width / 2
37     y1 = dis_height / 2
38     x1_change = 0
39     y1_change = 0
40     snake_List = []
41     Length_of_snake = 1
42     foodx = round(random.randrange(0, dis_width - snake_block) / 10.0) * 10.0
43     foody = round(random.randrange(0, dis_height - snake_block) / 10.0) * 10.0
44     while not game_over:
45         while game_close == True:
46             dis.fill(blue)
```

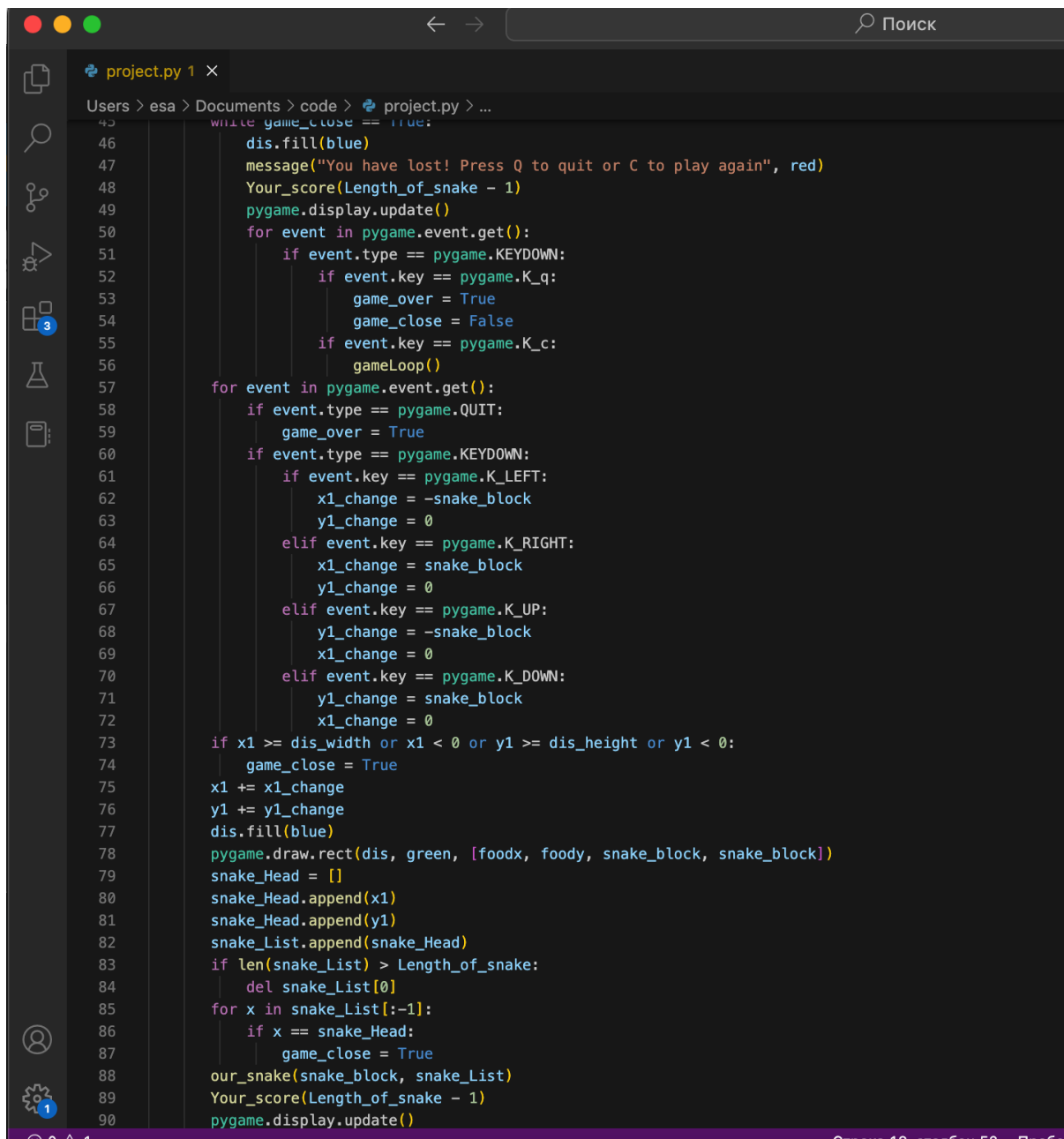
To create the snake, We will first initialize a few color variables in order to color the snake, food, screen, etc. The color scheme used in Pygame is RGB i.e “Red Green Blue”. In case you set all these to 0’s, the color will be black and all 255’s

will be white. So our snake will actually be a rectangle. To draw rectangles in Pygame, we used of a function called *draw.rect()* which help us draw the rectangle with the desired color and size.

To move the snake, we used the key events present in the KEYDOWN class of Pygame. The events that are used over here are, K_UP, K_DOWN, K_LEFT, and K_RIGHT to make the snake move up, down, left and right respectively. Also, the display screen is changed from the default black to white using the *fill()* method.

We have created new variables *x1_change* and *y1_change* in order to hold the updating values of the x and y

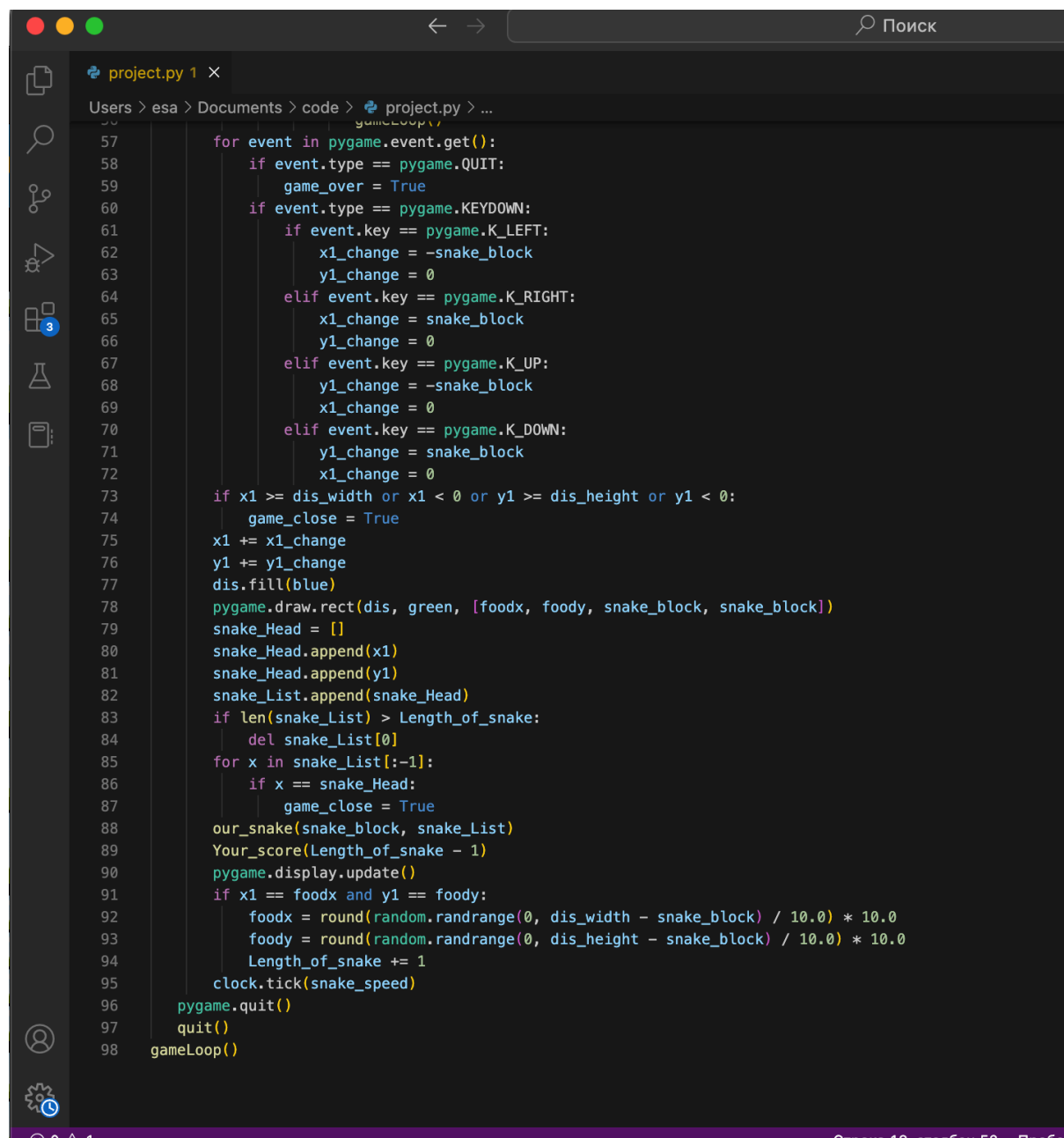
coordinates.



```
45 while game_close == True:
46     dis.fill(blue)
47     message("You have lost! Press Q to quit or C to play again", red)
48     Your_score(Length_of_snake - 1)
49     pygame.display.update()
50     for event in pygame.event.get():
51         if event.type == pygame.KEYDOWN:
52             if event.key == pygame.K_q:
53                 game_over = True
54                 game_close = False
55             if event.key == pygame.K_c:
56                 gameLoop()
57     for event in pygame.event.get():
58         if event.type == pygame.QUIT:
59             game_over = True
60         if event.type == pygame.KEYDOWN:
61             if event.key == pygame.K_LEFT:
62                 x1_change = -snake_block
63                 y1_change = 0
64             elif event.key == pygame.K_RIGHT:
65                 x1_change = snake_block
66                 y1_change = 0
67             elif event.key == pygame.K_UP:
68                 y1_change = -snake_block
69                 x1_change = 0
70             elif event.key == pygame.K_DOWN:
71                 y1_change = snake_block
72                 x1_change = 0
73     if x1 >= dis_width or x1 < 0 or y1 >= dis_height or y1 < 0:
74         game_close = True
75     x1 += x1_change
76     y1 += y1_change
77     dis.fill(blue)
78     pygame.draw.rect(dis, green, [foodx, foody, snake_block, snake_block])
79     snake_Head = []
80     snake_Head.append(x1)
81     snake_Head.append(y1)
82     snake_List.append(snake_Head)
83     if len(snake_List) > Length_of_snake:
84         del snake_List[0]
85     for x in snake_List[:-1]:
86         if x == snake_Head:
87             game_close = True
88     our_snake(snake_block, snake_List)
89     Your_score(Length_of_snake - 1)
90     pygame.display.update()
```

In this snake game, if the player hits the boundaries of the screen, then he loses. To specify that, we have made use of an 'if' statement that defines the limits for the x and y coordinates of the snake to be less than or equal to that of the screen.

The following code will increase the size of our snake when it eats the food. Also, if the snake collides with his own body, the game is over and you will see a message as “You Lost! Press Q-Quit or C-Play Again“. The length of the snake is basically contained in a list and the initial size that is specified in the following code is one block.

A screenshot of a code editor window titled 'project.py 1 x'. The editor shows a Python script for a snake game. The code is as follows:

```
57     for event in pygame.event.get():
58         if event.type == pygame.QUIT:
59             game_over = True
60         if event.type == pygame.KEYDOWN:
61             if event.key == pygame.K_LEFT:
62                 x1_change = -snake_block
63                 y1_change = 0
64             elif event.key == pygame.K_RIGHT:
65                 x1_change = snake_block
66                 y1_change = 0
67             elif event.key == pygame.K_UP:
68                 y1_change = -snake_block
69                 x1_change = 0
70             elif event.key == pygame.K_DOWN:
71                 y1_change = snake_block
72                 x1_change = 0
73     if x1 >= dis_width or x1 < 0 or y1 >= dis_height or y1 < 0:
74         game_close = True
75     x1 += x1_change
76     y1 += y1_change
77     dis.fill(blue)
78     pygame.draw.rect(dis, green, [foodx, foody, snake_block, snake_block])
79     snake_Head = []
80     snake_Head.append(x1)
81     snake_Head.append(y1)
82     snake_List.append(snake_Head)
83     if len(snake_List) > Length_of_snake:
84         del snake_List[0]
85     for x in snake_List[:-1]:
86         if x == snake_Head:
87             game_close = True
88     our_snake(snake_block, snake_List)
89     Your_score(Length_of_snake - 1)
90     pygame.display.update()
91     if x1 == foodx and y1 == foody:
92         foodx = round(random.randrange(0, dis_width - snake_block) / 10.0) * 10.0
93         foody = round(random.randrange(0, dis_height - snake_block) / 10.0) * 10.0
94         Length_of_snake += 1
95     clock.tick(snake_speed)
96     pygame.quit()
97     quit()
98     gameLoop()
```

The editor interface includes a sidebar with icons for file explorer, search, and other tools. The status bar at the bottom shows 'Строка 18, столбец 52' and 'Пробел'.

Conclusions:

We realized how important something like Github is for developing large projects. Since it allows us to keep track of all code changes, it also allows us to quickly share and merge our code. We all liked to play the snake game as children and so we decided to go back in time and play this beautiful game again.