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:

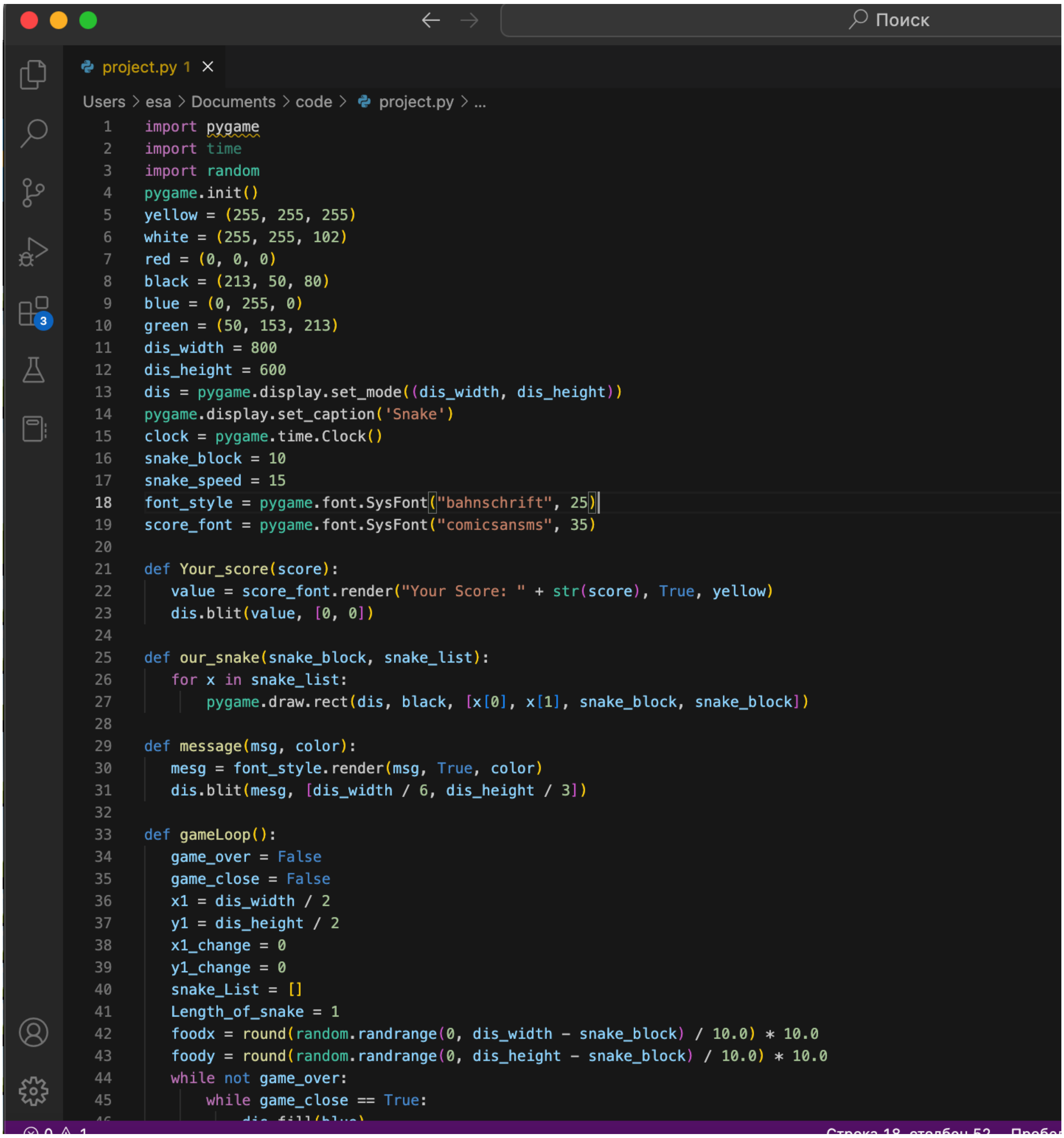
Azamat Amanzhol w65388
Yeslan Nurzhigitov w65381
Bogdan Valishin w63744

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.

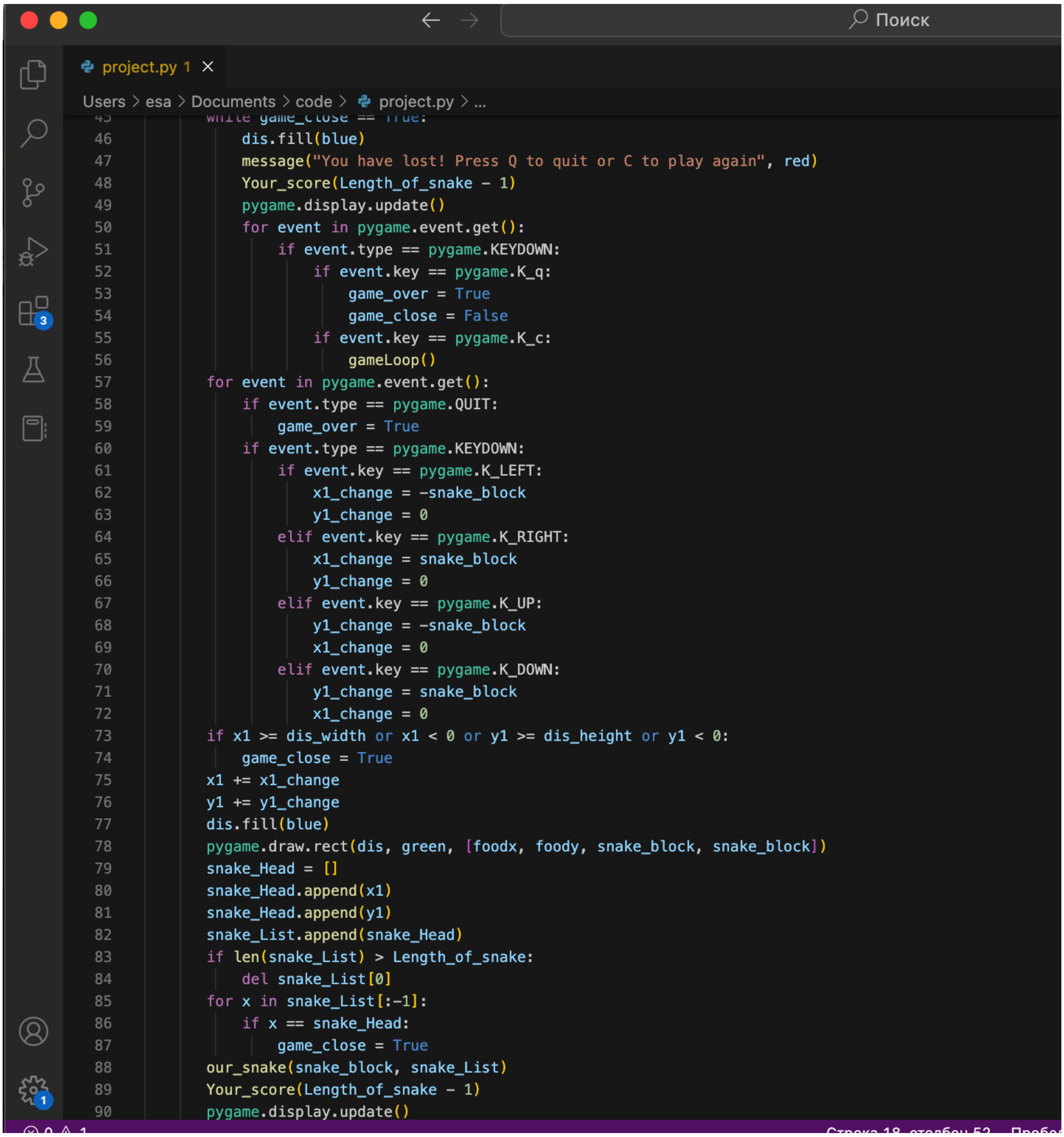

Поиск
project.py 1
Users > esa > Documents > code > project.py > ...
Строка 18, столбец 52 Пробел

```python
import pygame
import time
import random
pygame.init()
yellow = (255, 255, 255)
white = (255, 255, 102)
red = (0, 0, 0)
black = (213, 50, 80)
blue = (0, 255, 0)
green = (50, 153, 213)
dis_width = 800
dis_height = 600
dis = pygame.display.set_mode((dis_width, dis_height))
pygame.display.set_caption('Snake')
clock = pygame.time.Clock()
snake_block = 10
snake_speed = 15
font_style = pygame.font.SysFont("bahnschrift", 25)
score_font = pygame.font.SysFont("comicsansms", 35)

def Your_score(score):
    value = score_font.render("Your Score: " + str(score), True, yellow)
    dis.blit(value, [0, 0])

def our_snake(snake_block, snake_list):
    for x in snake_list:
        pygame.draw.rect(dis, black, [x[0], x[1], snake_block, snake_block])

def message(msg, color):
    mesg = font_style.render(msg, True, color)
    dis.blit(mesg, [dis_width / 6, dis_height / 3])

def gameLoop():
    game_over = False
    game_close = False
    x1 = dis_width / 2
    y1 = dis_height / 2
    x1_change = 0
    y1_change = 0
    snake_List = []
    Length_of_snake = 1
    foodx = round(random.randrange(0, dis_width - snake_block) / 10.0) * 10.0
    foody = round(random.randrange(0, dis_height - snake_block) / 10.0) * 10.0
    while not game_over:
        while game_close == True:
            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.

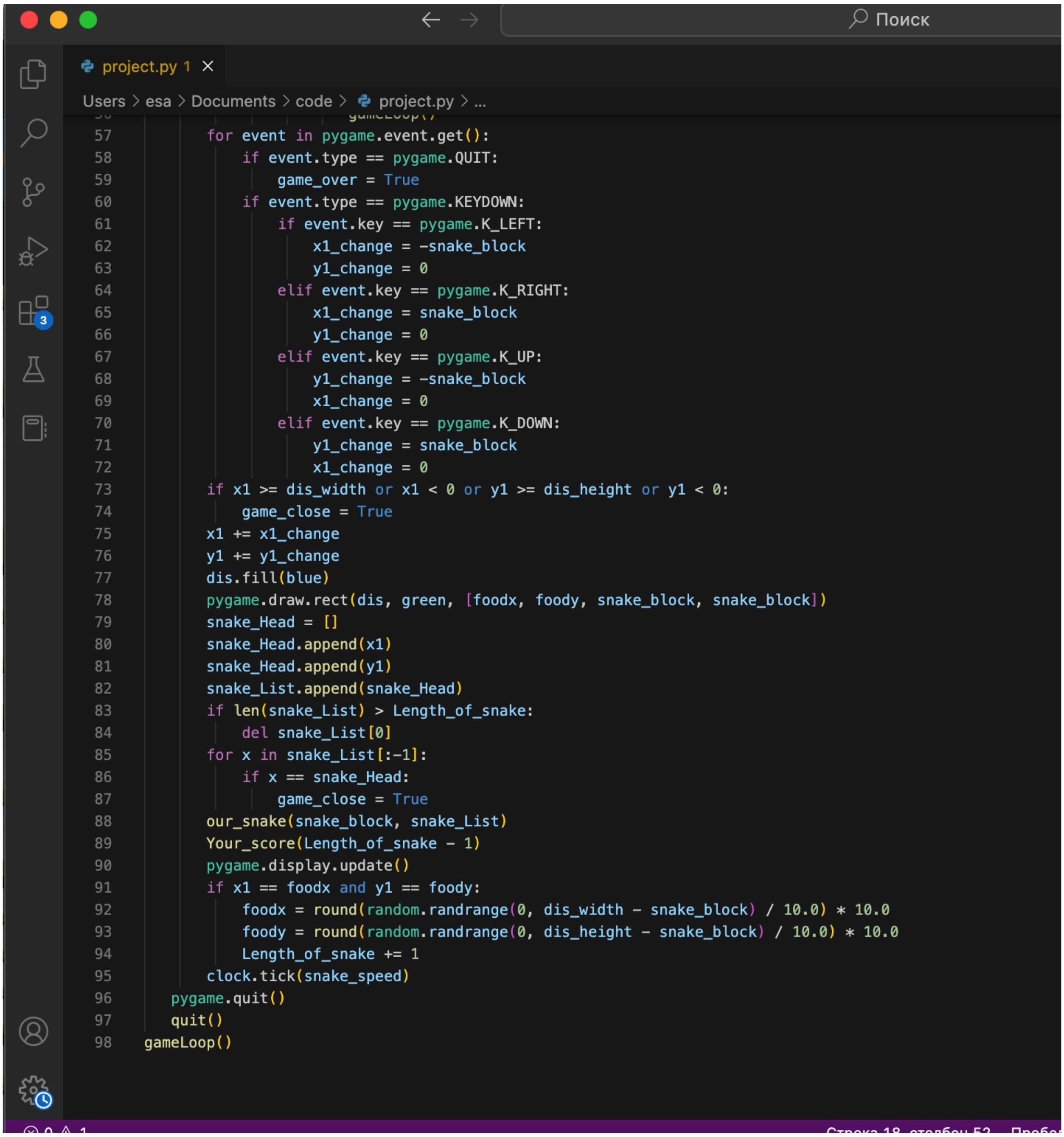
We have created new variables *x1_change* and *y1_change* in order to hold the updating values of the x and y coordinates.



```
        while game_close == True:
            dis.fill(blue)
            message("You have lost! Press Q to quit or C to play again", red)
            Your_score(Length_of_snake - 1)
            pygame.display.update()
            for event in pygame.event.get():
                if event.type == pygame.KEYDOWN:
                    if event.key == pygame.K_q:
                        game_over = True
                        game_close = False
                    if event.key == pygame.K_c:
                        gameLoop()
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                game_over = True
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_LEFT:
                    x1_change = -snake_block
                    y1_change = 0
                elif event.key == pygame.K_RIGHT:
                    x1_change = snake_block
                    y1_change = 0
                elif event.key == pygame.K_UP:
                    y1_change = -snake_block
                    x1_change = 0
                elif event.key == pygame.K_DOWN:
                    y1_change = snake_block
                    x1_change = 0
        if x1 >= dis_width or x1 < 0 or y1 >= dis_height or y1 < 0:
            game_close = True
        x1 += x1_change
        y1 += y1_change
        dis.fill(blue)
        pygame.draw.rect(dis, green, [foodx, foody, snake_block, snake_block])
        snake_Head = []
        snake_Head.append(x1)
        snake_Head.append(y1)
        snake_List.append(snake_Head)
        if len(snake_List) > Length_of_snake:
            del snake_List[0]
        for x in snake_List[:-1]:
            if x == snake_Head:
                game_close = True
        our_snake(snake_block, snake_List)
        Your_score(Length_of_snake - 1)
        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 ill 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.



```
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                game_over = True
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_LEFT:
                    x1_change = -snake_block
                    y1_change = 0
                elif event.key == pygame.K_RIGHT:
                    x1_change = snake_block
                    y1_change = 0
                elif event.key == pygame.K_UP:
                    y1_change = -snake_block
                    x1_change = 0
                elif event.key == pygame.K_DOWN:
                    y1_change = snake_block
                    x1_change = 0
        if x1 >= dis_width or x1 < 0 or y1 >= dis_height or y1 < 0:
            game_close = True
        x1 += x1_change
        y1 += y1_change
        dis.fill(blue)
        pygame.draw.rect(dis, green, [foodx, foody, snake_block, snake_block])
        snake_Head = []
        snake_Head.append(x1)
        snake_Head.append(y1)
        snake_List.append(snake_Head)
        if len(snake_List) > Length_of_snake:
            del snake_List[0]
        for x in snake_List[:-1]:
            if x == snake_Head:
                game_close = True
        our_snake(snake_block, snake_List)
        Your_score(Length_of_snake - 1)
        pygame.display.update()
        if x1 == foodx and y1 == foody:
            foodx = round(random.randrange(0, dis_width - snake_block) / 10.0) * 10.0
            foody = round(random.randrange(0, dis_height - snake_block) / 10.0) * 10.0
            Length_of_snake += 1
        clock.tick(snake_speed)
    pygame.quit()
    quit()
gameLoop()
```

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.