

--	--	--

Отчет о разработке игры на ругаме “Питон”

Состав нашей команды:

1. Антон Прохоров *разработчик*
2. Антон Рыбаков *тестировщик* ← это я
3. Михаил Довченко *разработчик*
4. Люшенко Денис *аналитик*

Мы распределили роли и “аналитик” проекта организовал scrum таблицу.

статус	разработчик	аналитик	тестировщик	В процессе
Нужно сделать	1 релиз			
разработчик	разработка v1.0.0 (14 ноя)	работа с разработчиком над v1.0.0 (14 ноя)	meeting с разработчиком (14 ноя)	код v1.0.0 (14 ноя)
	фиксы v1.0.0 (16 ноя)	работа с тестировщиком над v1.0.0 (15 ноя)	работа над тестом v1.0.0 (15 ноя)	тест v1.0.0 (15 ноя)
	разработка v1.0.1 (17 ноя)	работа с разработчиком над v1.0.1 (16 ноя)	meeting с разработчиком (16 ноя)	фикс v1.0.0 (16 ноя)
	фиксы v1.0.1 (19 ноя)	работа с тестировщиком над v1.0.1 (18 ноя)	работа над тестом v1.0.1 (18 ноя)	код v1.0.1 (17 ноя)
	разработка v1.0.2 (21 ноя)	работа с разработчиком над v1.0.2 (21 ноя)	meeting с разработчиком (19 ноя)	тест v1.0.1 (18 ноя)
	фиксы v1.0.2 (23 ноя)	работа с тестировщиком над v1.0.2 (22 ноя)	работа над тестом v1.0.2 (22 ноя)	фикс v1.0.1 (19 ноя)
	настройка exe (23 ноя)	работа с тестировщиком над exe (23 ноя)	тест exe (23 ноя)	код v1.0.2 (21 ноя)
	фикс exe			тест v1.0.2

Было выделено три релиза проекта, три версии v1.0.0 - v1.0.2.

1 релиз:

v1.0.0

Поле 300x300p

Питон - квадрат, бежит по полю

2 релиз:

Обновить игру "Питон"

v1.0.2

Добавить границу поля, зашел за границу - Game over

3 релиз:

Обновить игру "Питон"

v1.0.1

Добавить "яблоки", съел + размер питона.

--	--	--

--	--	--

1 релиз

```
import pygame

pygame.init()

BLACK = (0, 0, 0)
GRAY = (127, 127, 127)
WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
BLUE = (0, 0, 255)
YELLOW = (255, 255, 0)
CYAN = (0, 255, 255)
MAGENTA = (255, 0, 255)
ORANGE = (255,100,10)

dis = pygame.display.set_mode((800, 600))
pygame.display.set_caption('Dream team')

game_over = False

x = 150
y = 150

x_change = 0
y_change = 0

clock = pygame.time.Clock()

while not game_over:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_over = True
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_UP:
                y_change = -5
                x_change = 0
            elif event.key == pygame.K_DOWN:
                y_change = 5
                x_change = 0
            elif event.key == pygame.K_LEFT:
                x_change = -5
                y_change = 0
            elif event.key == pygame.K_RIGHT:
                x_change = 5
                y_change = 0

        x += x_change
        y += y_change
        dis.fill(GREEN)
        pygame.draw.rect(dis, ORANGE, [x, y, 5, 5])

    pygame.display.update()

    clock.tick(20)

pygame.quit()
quit()
```

--	--	--

--	--	--

Документация. Код 1.

Сначала мы импортируем библиотеку `pygame`, с которой мы будем работать.

Делаем `init` библиотеки для того чтобы с ней можно было работать.

В 5-14 строке мы задаем цвета, которые будем использовать для создания змейки.

```
import pygame

pygame.init()

BLACK = (0, 0, 0)
GRAY = (127, 127, 127)
WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
BLUE = (0, 0, 255)
YELLOW = (255, 255, 0)
CYAN = (0, 255, 255)
MAGENTA = (255, 0, 255)
ORANGE = (255, 100, 10)

dis = pygame.display.set_mode((800, 600))
pygame.display.set_caption('Dream team')
```

16 строка - задаем параметры для дисплея

17 строка - название файла.

21-22 строка начальное положение змейки.

24-25 строка - переменные, отвечающие за передвижение змейки.

27 строка - задержка, с которой будет передвигаться змейка.

29-45 строка - подключаем клавиши для управления.

--	--	--

--	--	--

47-48 строка - передвижение.

49 строка - устанавливаем зеленый цвет фона.

50 строка - отображение игры.

52 - изменение отображение.

54 строка - задержка, с которой двигается змейка.

56-57 строка - выход из строки.

Работа над первым релизом была закончена в сроки 14 – 16 ноября.

Тестировщик не обнаружил никаких проблем, фикс не выкатывался.

2 релиз

```
import pygame
import time
import random

pygame.init()

BLACK = (0, 0, 0)
GRAY = (127, 127, 127)
WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
BLUE = (0, 0, 255)
YELLOW = (255, 255, 0)
CYAN = (0, 255, 255)
MAGENTA = (255, 0, 255)
ORANGE = (255, 100, 10)

dis = pygame.display.set_mode((1440, 1080))
pygame.display.set_caption('Dream team')

clock = pygame.time.Clock()

snake = 20
snake_speed = 15

font_style = pygame.font.SysFont(None, 30)

def message(msg, color):
    mesg = font_style.render(msg, True, color)
    dis.blit(mesg, [1440/3, 1080/3])

def gameLoop(): # creating a function
    close = False
    over = False
```

--	--	--

--	--	--

```

x = 1440 / 2
y = 1080 / 2

foodx = round(random.randrange(0, 1440 - snake) / 10.0) * 10.0
foody = round(random.randrange(0, 1080 - snake) / 10.0) * 10.0
xc = 0
yc = 0

while not over:

    while close == True:
        dis.fill(GREEN)
        message("Game over! 'r' - restart, q - 'quit'", BLACK)
        pygame.display.update()

        for event in pygame.event.get():
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_q or event.key == pygame.K_й:
                    over = True
                    close = False
                if event.key == pygame.K_r or event.key == pygame.K_к:
                    gameLoop()

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            over = True
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_UP:
                yc = -snake
                xc = 0
            elif event.key == pygame.K_DOWN:
                yc = snake
                xc = 0
            elif event.key == pygame.K_LEFT:
                xc = -snake
                yc = 0
            elif event.key == pygame.K_RIGHT:
                xc = snake
                yc = 0

    x+=xc
    y+=yc

    if x >= 1440 or x < 0 or y >= 1080 or y < 0:
        close = True

    dis.fill(GREEN)
    pygame.draw.rect(dis, RED, [foodx, foody, snake, snake])
    pygame.draw.rect(dis, YELLOW, [x, y, snake, snake])
    pygame.display.update()

    clock.tick(snake_speed)

pygame.quit()
quit()

gameLoop()

```

--	--	--

--	--	--

Документация. Код 2.

1-3 строка - импортируем библиотеки.

5 строка - инициализируем библиотеку pygame.

7-16 строка - цвета, которые будем использовать для игры

18 строка - размер окна с игрой.

19 строка - название окна с игрой.

23 строка - размер змейки.

24 строка - скорость передвижения змейки.

26 строка - стиль шрифта.

29-31 строка - функция для вывода сообщения.

34 строка - функция с игрой.

35-36 строка - переменные для циклов.

38-39 строка - начальные координаты змейки.

41-42 - генератор яблок.

46 - цикл для игры.

48-59 - цикл на случай проигрыша:

49-51 строка - визуал экрана проигрыша.

--	--	--

--	--	--

53-59 строки - считывают нажатую клавишу и после либо закрывают игру, либо начинают её заново.

61-78 строка - считывают нажатие клавиши в режиме игры и изменяют координаты змейки в зависимости от нажатой клавиши.

80-81 строка - проверяют, не вышла ли змейка за пределы экрана. Если вышла, то вызывают экран проигрыша.

83-86 строки - отвечают за визуальное передвижения змейки.

88 строка - скорость передвижения змейки.

90-94 строки - заканчивают цикл игры.

Работа над вторым релизом была закончена в сроки 17 – 19 ноября.

Тестировщик не обнаружил никаких проблем, фикс не выкатывался.

3 релиз

```
import pygame
import time
import random

pygame.init()

font_style = pygame.font.SysFont("bahnschrift", 25)
score_font = pygame.font.SysFont("comicsansms", 35)

BLACK = (0, 0, 0)
GRAY = (127, 127, 127)
WHITE = (255, 255, 255)
RED = (255, 0, 0)
GREEN = (0, 255, 0)
BLUE = (0, 0, 255)
YELLOW = (255, 255, 0)
CYAN = (0, 255, 255)
MAGENTA = (255, 0, 255)
ORANGE = (255, 100, 10)

dis = pygame.display.set_mode((1080, 720))
```

--	--	--

--	--	--

```

pygame.display.set_caption('Dream team')

clock = pygame.time.Clock()

snake = 20
snake_speed = 15

font_style = pygame.font.SysFont(None, 30)

def message(msg, color):
    mesg = font_style.render(msg, True, color)
    dis.blit(mesg, [1080/3, 720/3])

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

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

def gameLoop():
    close = False
    over = False

    x = 1080 / 2
    y = 720 / 2

    foodx = round(random.randrange(0, 1080 - snake) / 10.0) * 10.0
    foody = round(random.randrange(0, 720 - snake) / 10.0) * 10.0
    xc = 0
    yc = 0
    snake_list = []
    snake_len = 1

    while not over:

        while close == True:
            dis.fill(GREEN)
            message("Game over! 'r' - restart, q - 'quit'", BLACK)
            pygame.display.update()

            for event in pygame.event.get():
                if event.type == pygame.KEYDOWN:
                    if event.key == pygame.K_q or event.key == pygame.K_й:
                        over = True
                        close = False
                    if event.key == pygame.K_r or event.key == pygame.K_к:
                        gameLoop()

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                over = True
            if event.type == pygame.KEYDOWN:
                if event.key == pygame.K_UP:
                    yc = -snake
                    xc = 0
                elif event.key == pygame.K_DOWN:
                    yc = snake

```

--	--	--

--	--	--

```

        xc = 0
        elif event.key == pygame.K_LEFT:
            xc = -snake
            yc = 0
        elif event.key == pygame.K_RIGHT:
            xc = snake
            yc = 0
    if x >= 1080 or x < 0 or y >= 720 or y < 0:
        close = True
    x+=xc
    y+=yc

    if x >= 1080 or x < 0 or y >= 720 or y < 0:
        close = True

    dis.fill(GREEN)
    pygame.draw.rect(dis, RED, [foodx, foody, snake, snake])
    snake_Head = []
    snake_Head.append(x)
    snake_Head.append(y)
    snake_list.append(snake_Head)
    if len(snake_list) > snake_len:
        del snake_list[0]

    for i in snake_list[:-1]:
        if i == snake_Head:
            close = True

    snakes(snake, snake_list)
    scores(snake_len - 1)
    pygame.draw.rect(dis, YELLOW, [x, y, snake, snake])
    pygame.display.update()

    if x == foodx and y == foody:
        snake_len+=1
        foodx = round(random.randrange(0, 1080 - snake) / 10.0) * 10.0
        foody = round(random.randrange(0, 720 - snake) / 10.0) * 10.0

    clock.tick(snake_speed)

pygame.quit()
quit()

gameLoop()

```

22 ноября тестировщик обнаружил проблемы с кодом 3 релиза. Питон плохо ел яблоки, надо было увеличить хит бокс яблока.

23 ноября был сделан фикс 3 релиза.

```

snakes(snake, snake_list)
scores(snake_len - 1)
pygame.draw.rect(dis, YELLOW, [x, y, snake, snake])
pygame.display.update()
if (-10 <= x - foodx <= 10) and (-10 <= y - foody <= 10):
    snake_len+=1
    foodx = round(random.randrange(0, 1080 - snake) / 10.0) * 10.0
    foody = round(random.randrange(0, 720 - snake) / 10.0) * 10.0

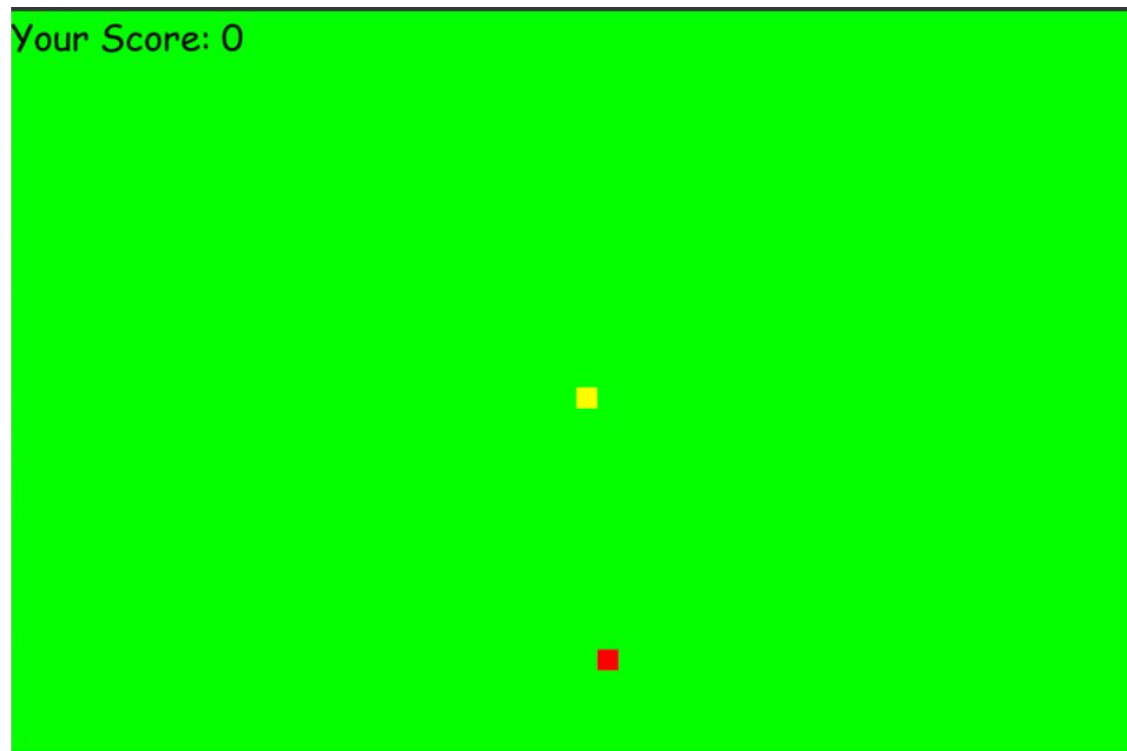
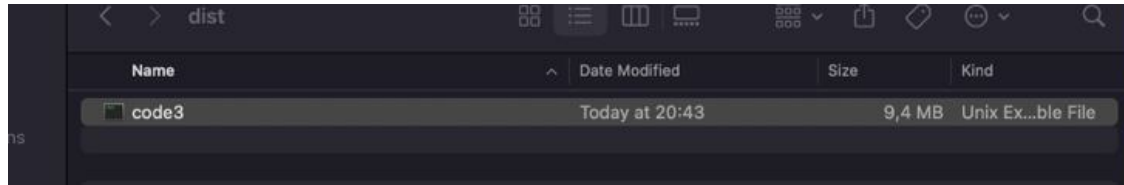
```

--	--	--

--	--	--

23 ноября тестировщик одобрил фикс v1.0.2.

Был создан exe v1.0.2.



Git



--	--	--

--	--	--

<https://github.com/Khail909/dev-tools-14>

Тесты:

```
def scores(score):
```

```
    score -= 1
```

```
    return score
```

```
def gameLoop(dvizh):
```

```
    close = False
```

```
    over = False
```

```
    snake = 10
```

```
    x = 1080 / 2
```

```
    y = 720 / 2
```

```
    xc = 0
```

```
    yc = 0
```

```
    snake_list = []
```

```
    snake_len = 1
```

```
    while not over:
```

```
        if dvizh == 'w':
```

```
            yc = -snake
```

```
            xc = 0
```

```
        elif dvizh == 's':
```

```
            yc = snake
```

```
            xc = 0
```

```
        elif dvizh == 'a':
```

```
            xc = -snake
```

```
            yc = 0
```

--	--	--

--	--	--

```

    elif dvizh == 'd':
        xc = snake
        yc = 0
    if x >= 1080 or x < 0 or y >= 720 or y < 0:
        close = True
        return y
    x+=xc
    y+=yc

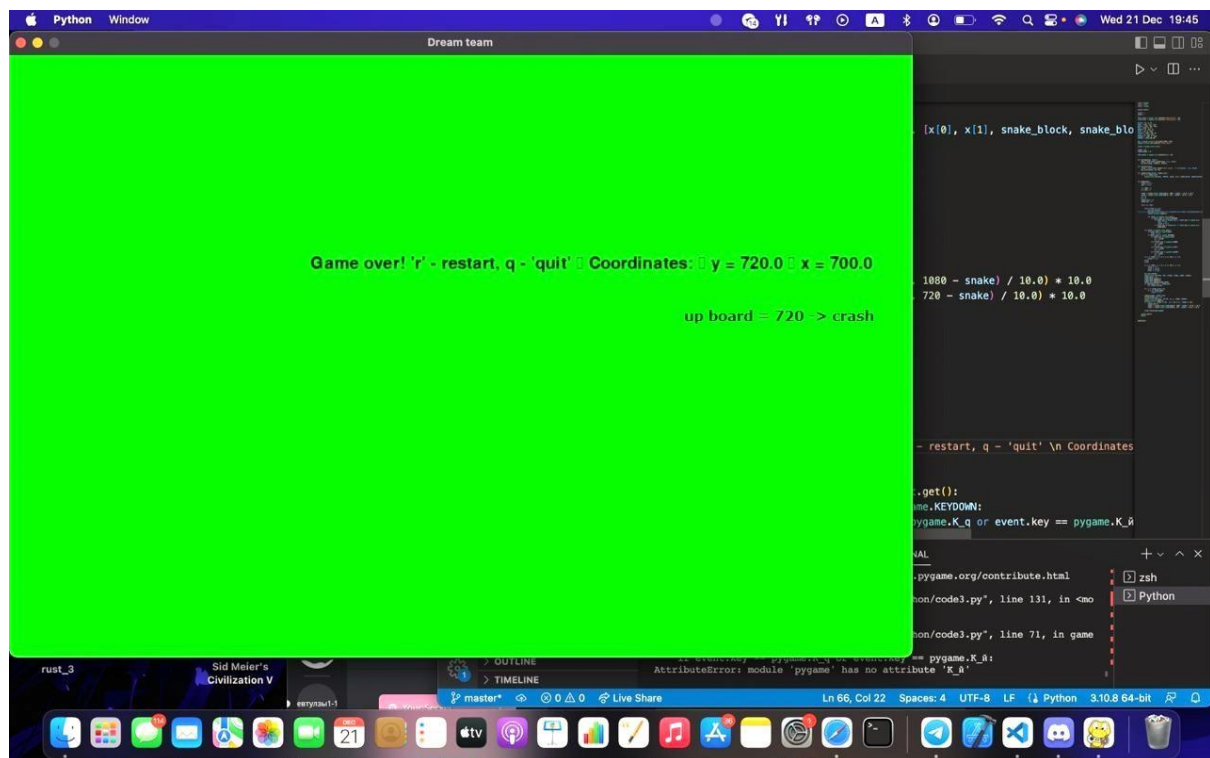
    if x >= 1080 or x < 0 or y >= 720 or y < 0:
        close = True
        return y
////////////////////////////////////
import unittest
from unitte import scores
from unitte import gameLoop

class TestZmeika(unittest.TestCase):
    def test_scores(self):
        self.assertEqual(scores(1), 0)
    def test_gameLoop(self):
        self.assertEqual(gameLoop('w'), 720)

```

--	--	--

--	--	--



```
import unittest
```

```
from unittest import TestCase
```

```
from unittest import TestCase
```

```
class TestZmeika(unittest.TestCase):
```

```
    def test_scores(self):
```

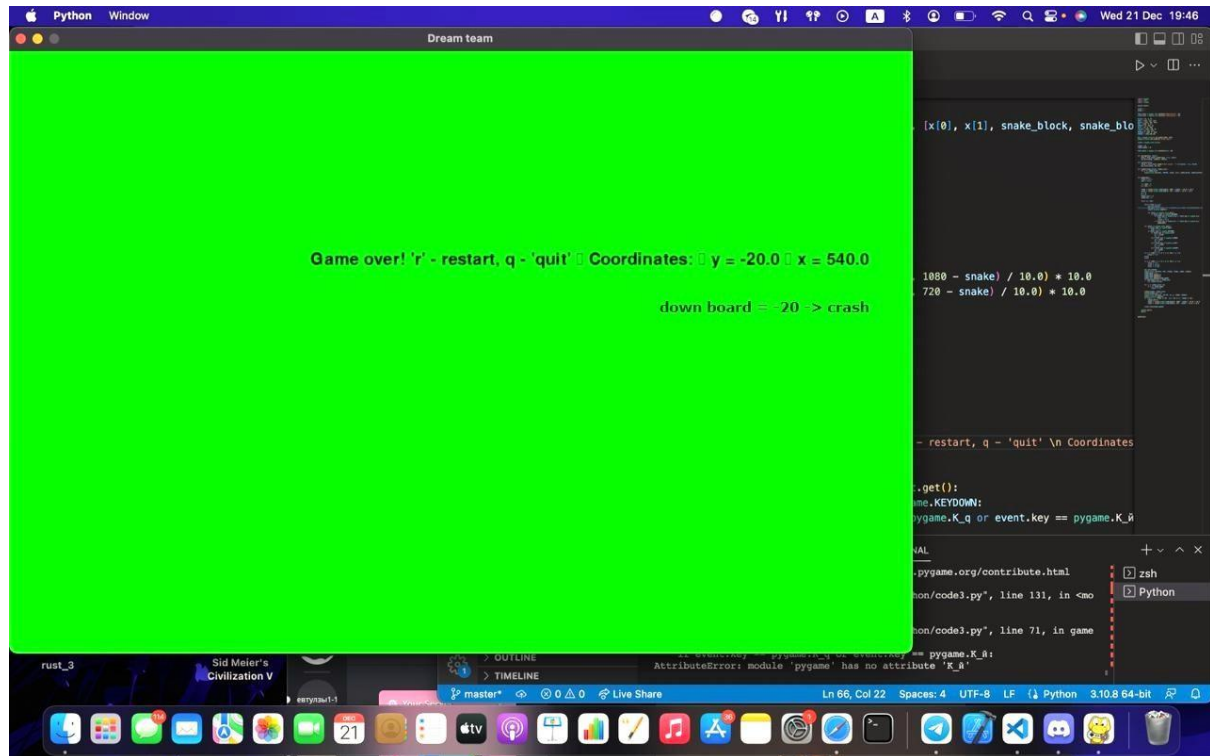
```
        self.assertEqual(scores(1), 0)
```

```
    def test_gameLoop(self):
```

```
        self.assertEqual(gameLoop('a'), 720)
```

--	--	--

--	--	--



import unittest

from unittest import scores

from unittest import gameLoop

class TestZmeika(unittest.TestCase):

def test_scores(self):

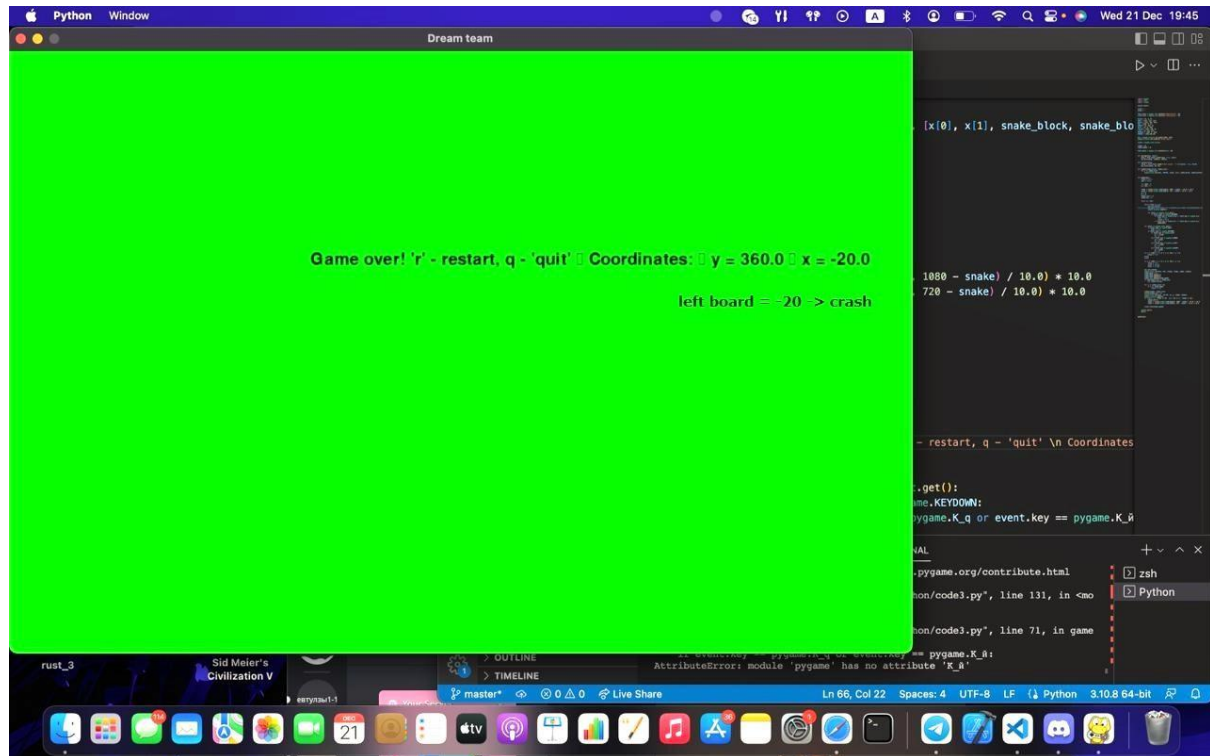
self.assertEqual(scores(1), 0)

def test_gameLoop(self):

self.assertEqual(gameLoop('d'), 720)

--	--	--

--	--	--



```
import unittest
```

```
from unittest import scores
```

```
from unittest import gameLoop
```

```
class TestZmeika(unittest.TestCase):
```

```
    def test_scores(self):
```

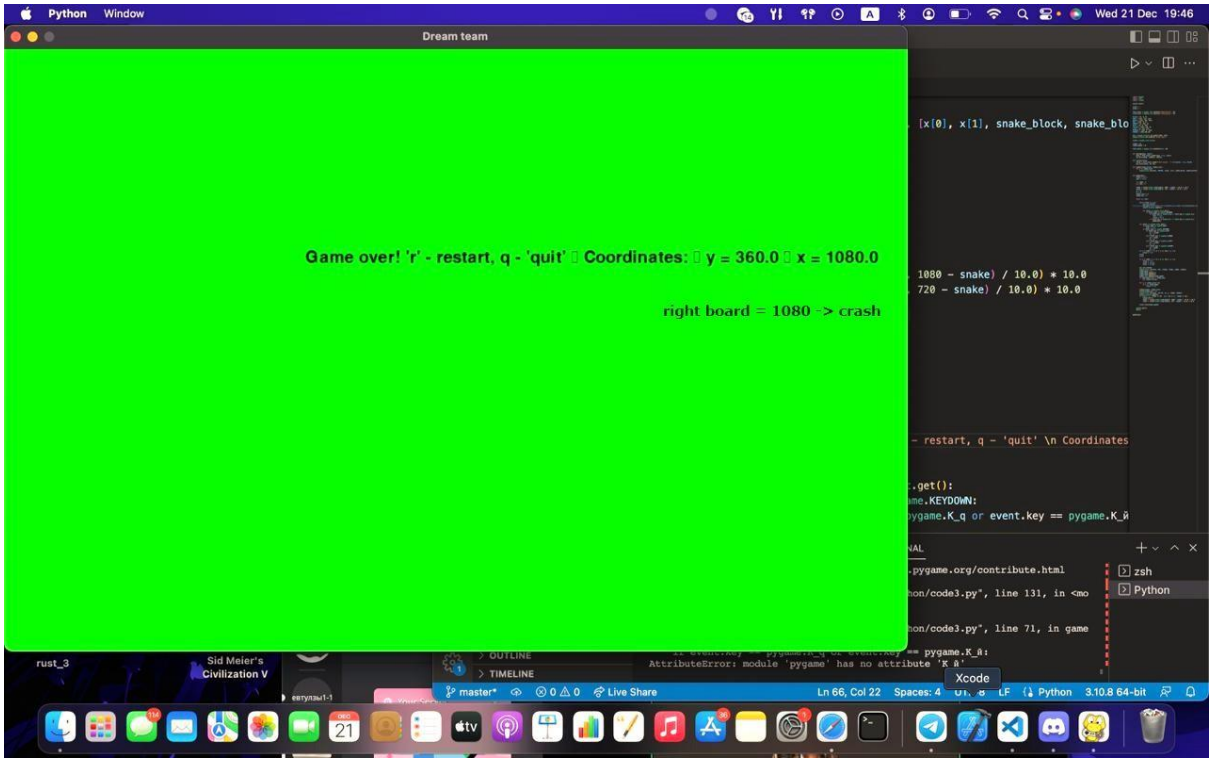
```
        self.assertEqual(scores(1), 0)
```

```
    def test_gameLoop(self):
```

```
        self.assertEqual(gameLoop('s'), 720)
```

--	--	--

--	--	--



--	--	--