

***ИНСТИТУТ ТЕХНИКИ И ЦИФРОВЫХ ТЕХНОЛОГИЙ***

***ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ***

***КАФЕДРА МАТЕМАТИКИ И ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН***

**Утверждаю**

Заведующий кафедрой

математики и

естественнонаучных дисциплин

\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_ /

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2021

**Отчет**

**По дисциплине:** Технологии и системы коллективной разработки программ

**На тему:** «Разработка видеоигры «”Змейка”**»**

Направление подготовки 01.03.02. Прикладная математика и информатика

Выполнил:

студент группы ПМИ-18

Калиновский С.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Королев 2021

**Оглавление**

[Введение 3](#_Toc95746935)

[1. Подготовка к созданию игры 4](#_Toc95746936)

[2. Создание игры 6](#_Toc95746937)

[2.1. Создание окна приложения 6](#_Toc95746938)

[2.2. Создание змейки 7](#_Toc95746939)

[2.3. Границы и взаимодействие с ними 8](#_Toc95746940)

[2.4. Добавление еды 9](#_Toc95746941)

[2.5. Добавление вывода счёта на экран 11](#_Toc95746942)

[Заключение 13](#_Toc95746943)

[Список литературы 14](#_Toc95746944)

[Приложение 15](#_Toc95746945)

# Введение

**Актуальность:** видеоигры с каждым годом становятся только актуальнее. Люди разных возрастов ежедневно играют в игры. Существует множество видеоигр,«Змейка» одна из самых распространённых аркад в мире, именно поэтому я решил выбрать её для проекта.  
 В связи с огромным спросом, разработчики видеоигр с каждым годом становятся очень востребованными, поэтому тема разработки игр в наша время является очень актуальной.

**Цель проекта:** разработать и модернизировать игру «Змейка».

**Задачи:** —Выбрать язык программирования для разработки игры;

—Создать поле из клеточек, где случайным образом появляется еда;

—Добавить змейку, которая всё время двигается и которой можно управлять;

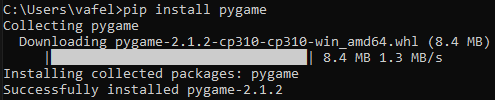
—Добавить условие при котором, если змейка на своём пути встречает еду — еда исчезает, появляется в новом месте, а змейка удлиняется на одну клетку;

—Добавить условие при котором, если змейка врежется в границу поля или в саму себя, игра заканчивается.

# Подготовка к созданию игры

Проанализировав в интернете возможные языки программирования, я решил остановиться на языке Python, потому что он очень универсальный и отлично подходит не очень опытному программисту.

Первое, что надо сделать, это установить библиотеку PyGame. Это можно сделать выполнив команду в командной строке Windows, изображённую на рисунке 1.



**Рис. 1.** Установка библиотеки PyGame через Командную строку

Все основные функции Pygame перечислены в таблице 1.

**Таблица. 1.** Основные функции PyGame [4]

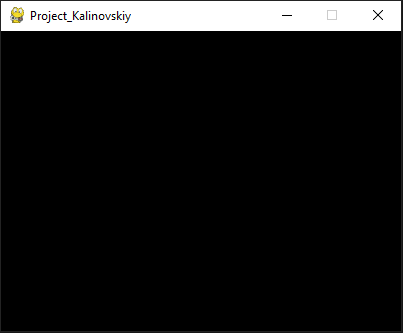
|  |  |
| --- | --- |
| **Функция** | **Описание** |
| init() | Инициализирует все модули PyGame (возвращает кортеж в случае успеха или неудачи). |
| display.set\_mode() | Для создания поверхности принимает в качестве параметра либо список либо кортеж (кортеж предпочтительней). |
| update() | Обновляет экран. |
| quit() | Используется для деинициализации всех модулей. |
| set\_caption() | Устанавливает текст заголовка в верхней части экрана. |
| event.get() | Возвращает список всех событий. |
| Surface.fill() | Заполняет пространство сплошным цветом. |
| time.Clock() | Отслеживание времени. |
| font.SysFont() | Задает шрифт Pygame, используя системные ресурсы. |

# Создание игры

# Создание окна приложения

С помощью инструкции import подключаю нужные для работы модули: pygame, time и random [3]. Для создания графического окна при помощи Pygame нужно воспользоваться функцией display.set\_mode(). Также необходимо пользоваться методом init() для инициализации окна в начале кода и методом quit() для его закрытия в конце. Метод update() используется для применения каких-либо изменений в окне. Еще существует метод flip(), который работает похожим с update() образом. Разница заключается лишь в том, что метод flip() переписывает весь экран целиком, а метод update() применяет именно изменения (хотя если его использовать без параметров, то он тоже переписывает весь экран). Чтобы окно не закрывалось сразу после запуска кода, нужно воспользоваться циклом while и функциями event.get() и display.set\_caption() (эта функция даёт имя всплывающему окну).

Теперь у нас есть графическое окно для игры(рис.2), но при взаимодействии с кнопкой close, оно не закроется. Это потому, что мы не предусмотрели такого поведения. Для решения этой задачи в Pygame предусмотрено событие «QUIT».



**Рис. 2.** Окно игры "Змейка"

# Создание змейки

Перед тем как создать змейку, инициируем несколько цветовых переменных для раскрашивания самой змейки, еды и экрана. В Pygame используется цветовая схема RGB (RED, GREEN, BLUE). Установка всех цветов в 0 соответствует черному цвету, а в 255 — соответственно, белому. Фактически, наша змейка является прямоугольником. Чтобы нарисовать прямоугольник в Pygame, можно воспользоваться функцией draw.rect(), которая нарисует нам прямоугольник заданного цвета и размера [2]. На рисунке 3 изображён полученный результат.

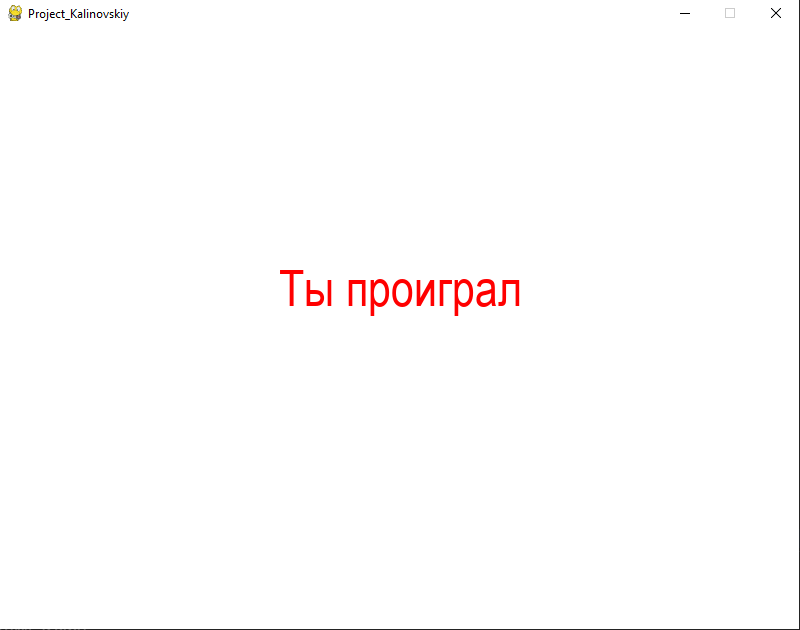
Теперь нужно научить змейку двигаться. Чтобы передвигать змейку, мы будем использовать ключевые события из класса KEYDOWN библиотеки Pygame. События K\_UP, K\_DOWN, K\_LEFT, и K\_RIGHT заставят змейку двигаться вверх, вниз, влево и вправо соответственно. Также, цвет дисплея меняется от синего до белого при помощи метода fill(). Для сохранения изменений координат x и y мы создали две новых переменные: x1\_change и y1\_change.



**Рис. 3.** Змейка экране

# Границы и взаимодействие с ними

В игре змейка игрок проигрывает, если касается границы экрана. Чтобы задать такое поведение, мы должны использовать инструкцию if, которая будет следить за тем, чтобы координаты X и Y были меньше размеров экрана. Также, если змейка врезается в себя, игра заканчивается и выводит сообщение на экран(рис.4).



**Рис. 4.** Сообщение на экране при проигрыше

# Добавление еды

Нужно обязательно в игру добавить еду для змейки, чтобы, после поглощения еды, змейка увеличивалась. Процесс роста змейки показан на рисунках 5 и 6.



**Рис. 5.** Змейка до того как съела еду



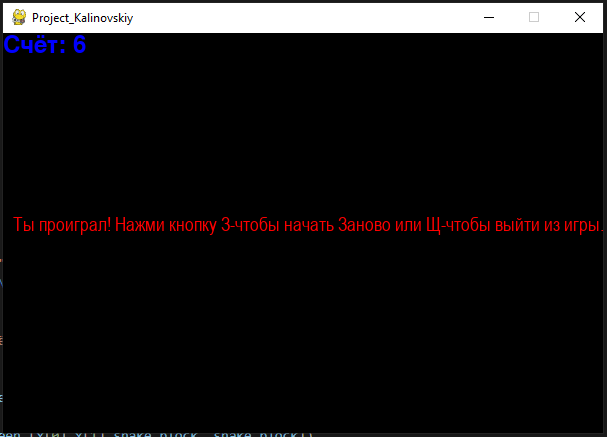
**Рис. 6.** Змейка после того как несколько раз съела еду

# Добавление вывода счёта на экран

Последнее, но весьма важное дело: отображение счёта игрока. Для этого создаю новую функцию your\_score. Это функция будет показывать размер змейки за вычетом 1 (так как это начальный размер змейки) [1]. Вывод счёта будет отображаться как во время игры(рис.7), так и после неё(рис.8).



**Рис. 7.** Вывод подсчёта очков во время игры



**Рис. 8.** Вывод очков после игры

# Заключение

Мною выполнена разработка видеоигры «змейка». В дальнейшем полученный продукт можно будет модернизировать добавив систему уровней и т.д., используя для этого не только Python.

# Список литературы

1. Учебник PyGame // [электронный ресурс] URL: <https://pythobyte.com/pygame-graphical-hi-lo-game-21f96a07/> (Дата обращения: 01.02.2022)
2. Змейка на Python // [электронный ресурс] URL: <http://pythonicway.com/python-games/python-arcade/18-python-snake> (Дата обращения: 27.01.2022)
3. Snake Game in Python – Using Pygame module // [электронный ресурс] URL: <https://www.geeksforgeeks.org/snake-game-in-python-using-pygame-module/> (Дата обращения: 27.01.2022)
4. How To implement Snake Game in Python? // [электронный ресурс] URL: <https://www.edureka.co/blog/snake-game-with-pygame/> (Дата обращения: 20.01.2022)

# Приложение

**Код разработанной видеоигры «змейка»**

import pygame

import time

import random

pygame.init()

dis\_width=600

dis\_height=400

dis=pygame.display.set\_mode((dis\_width,dis\_height))

pygame.display.set\_caption('Project\_Kalinovskiy')

green=(0,128,0)

blue=(0,0,255)

red=(255,0,0)

white=(255,255,255)

black=(0,0,0)

yellow=(255,255,0)

clock=pygame.time.Clock()

snake\_block=10

snake\_speed=12

font\_style=pygame.font.SysFont("Arial",19)

score\_font=pygame.font.SysFont(None,35)

def your\_score(score):

    value=score\_font.render("Счёт: "+str(score),True,blue)

    dis.blit(value,[0,0])

def our\_snake(snake\_block, snake\_list):

    for x in snake\_list:

        pygame.draw.rect(dis,green,[x[0],x[1],snake\_block, snake\_block])

def message(msg,color):

    mesg=font\_style.render(msg,True,color)

    dis.blit(mesg, [10,180])

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(black)

            message("Ты проиграл! Нажми кнопку З-чтобы начать Заново или Щ-чтобы выйти из игры.", 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\_o:

                        game\_over=True

                        game\_close=False

                    if event.key==pygame.K\_p:

                        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(yellow)

        pygame.draw.rect(dis,red,[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()