Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Санкт-Петербургский государственный университет промышленных технологий и дизайна»

|  |
| --- |
| **Институт дополнительного профессионального образования** |

|  |  |
| --- | --- |
|  |  |
| Направление подготовки: | 29.03.02 – Технологии и проектирования текстильных изделий |
| Профиль подготовки: | Проектирование, технологии и художественное оформление текстильных изделий |

|  |
| --- |
| **ОТЧЕТ** |
| Программирование на языке Python  Создание игры «Змейка» |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Руководитель  от профильной организации / структурного подразделения СПбГУПТД\*: |  |  | | | | |  |  |
| *(наименование организации)* |  | *(должность, Ф.И.О., печать организации)* | | | | |  | *(подпись, печать)* |
| Руководитель  от СПбГУПТД: |  | Баев Никита Александрович | | | | |  |  |
|  |  | *(должность, ученая степень / звание, Ф.И.О.)* | | | | |  | *(подпись)* |
| Обучающийся: |  | Минхаерова Альфинур Якубовна | | | | |  |  |
|  |  | *(Ф.И.О.)* | | | | |  | *(подпись)* |
| Курс |  | 2 |  | Учебная группа: |  | 2-ТД-8 | |

Санкт-Петербург

2023

Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный университет промышленных технологий и дизайна»

**Совместный рабочий график (план) проведения практики и индивидуальное задание**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Обучающийся | | | | | Минхаерова Альфинур Якубовна | | | | | | | | |
|  | | | | | *(Ф.И.О.)* | | | | | | | | |
| Институт | | | Институт текстиля и моды | | | | | | | | | | |
|  | | | | | *(наименование института)* | | | | | | | | |
| Курс |  | 2 | | Учебная группа | | |  | 2-ТД-8 | | | Форма обучения |  | очная |
|  |  |  | |  | | |  |  | | |  |  |  |
| Направление подготовки (специальность), профиль | | | | | | | | | | 29.03.02 – Технологии и проектирования текстильных изделий | | | |
|  | | | | | | | | | | | | | |
|  | | | | | | | | |  | | | | |
| Сроки прохождения практики | | | | | | с 10 апреля 2023г. по 30 апреля 2023г. | | | | | | | |
|  | | | | | | *(по календарному учебному графику)* | | | | | | | |

**Совместный рабочий график (план) проведения практики**

|  |  |  |
| --- | --- | --- |
| Дата | Содержание выполняемых работ и заданий | |
| Общие (типовые вопросы, изучаемые в ходе практики) | | |
|  | **Раздел 1. Проведение обследований предметной**  **области в рамках решаемой задачи** | |
| Основные методы обследования предметной области | |
| Теоретические аспекты решаемой задачи | |
|  | **Раздел 2. Определение алгоритма решения задачи** | |
| Решение задачи Алгоритмизация | |
| Решение задачи Программный код | |
|  | | **Раздел 3. Написание программного кода (реализация алгоритма)** |
|  | Тестирование программного кода | |
| Оформление отчета | |
| Оформление презентации | |

**Требования по выполнению и оформлению индивидуального задания** Пояснительная записка оформляется в соответствии с требованиямиГОСТ 7.32-2017 «Отчет о научно-исследовательской работе. Структура и правила оформления» и ГОСТ Р 7.0.100-2018 «Библиографическая запись. Библиографическое описание. Общие требования и правила составления».

**Вид (ы) отчетных материалов по практике и требования к их оформлению**

**в соответствии с индивидуальным заданием**

1. *Пояснительная записка (отчет)*
2. *Презентация по материалам практики*

Руководитель практики

от СПбГУПТД \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Баев Н.А.

*(подпись, ф.и.о.)*

Принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Минхаерова А.Я.

*(подпись, ф.и.о. обучающегося)*

Дата получения обучающимся индивидуального задания 13.04.2023 г.

Содержание

[ВВЕДЕНИЕ 5](#_Toc133526944)

[1.ОСНОВНЫЕ МЕТОДЫ ОБСЛЕДОВАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ РЕШАЕМОЙ ЗАДАЧИ 6](#_Toc133526945)

[1.1. Знакомство с основным оператором Python и Pygame 6](#_Toc133526946)

[1.2. Игровой цикл 8](#_Toc133526947)

[2. ОПРЕДЕЛЕНИЕ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ 9](#_Toc133526948)

[2.1. Продумываем и описываем правила игры 9](#_Toc133526949)

[2.2. Алгоритмизация 9](#_Toc133526950)

[2.3. Программный код 11](#_Toc133526951)

[ЗАКЛЮЧЕНИЕ 15](#_Toc133526952)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ: 16](#_Toc133526953)

# ВВЕДЕНИЕ

Змейка — классическая аркадная игра, которая нравится геймерам на протяжении десятилетий. Игра включает в себя управление змеей, когда она движется по игровому полю, ест пищу и становится длиннее с каждым укусом. По мере того, как змея растет, игра становится все более сложной, с препятствиями и опасностями, которых нужно избегать. В этом докладе продемонстрировано создание игры «Змейка» с использованием языка программирования Python. В отчете будет представлен всесторонний обзор шагов, связанных с созданием игры с использованием Python, и того, как эта популярная игра была воплощена в жизнь.

# 1.ОСНОВНЫЕ МЕТОДЫ ОБСЛЕДОВАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ РЕШАЕМОЙ ЗАДАЧИ

* 1. Знакомство с основным оператором Python и Pygame

Pygame — это модуль Python, предназначенный для написания игр и мультимедийного программного обеспечения. Он построен на основе библиотеки Simple DirectMedia Layer (SDL), которая обеспечивает доступ к низкоуровневым функциям программирования игр, таким как графика, звук и обработка ввода.

Pygame упрощает создание игр с помощью Python, предоставляя высокоуровневый API для обработки игровых объектов, графики и программирования, управляемого событиями. Он также включает в себя набор вспомогательных функций и инструментов для создания игровых ресурсов, таких как изображения, листы спрайтов и звуковые файлы.

Некоторые из ключевых особенностей Pygame включают в себя:

* Поддержка 2D-графики и анимации
* Звук и воспроизведение музыки
* Обработка ввода для клавиатуры, мыши и игровых контроллеров
* Поддержка различных форматов изображений, включая PNG, JPEG и GIF.
* Поддержка рендеринга шрифтов и ввода текста
* Кроссплатформенная совместимость с версиями, доступными для Windows, Mac и Linux.

В целом, Pygame — это мощный и гибкий инструмент для создания игр и мультимедийного программного обеспечения на Python.

Установка [Pygame](https://ru.wikipedia.org/wiki/Pygame" \t "_blank)

Первое, что нам надо сделать, это установить библиотеку Pygame. Это можно сделать, просто выполнив следующую команду: pip install pygame

Выполнив это, просто импортируем эту библиотеку и приступим к разработке игры. Но перед этим давайте взглянем на основные функции данной библиотеки, которые мы будем использовать при создании игры.

|  |  |
| --- | --- |
| **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. |

* 1. Игровой цикл

В сердце каждой игры лежит цикл, который принято называть «игровым циклом». Он запускается снова и снова, делая все, чтобы работала игра. Каждый цикл в игре называется кадром.

В каждом кадре происходит масса вещей, но их можно разбить на три категории:

* Обработка ввода (события)

Речь идет обо всем, что происходит вне игры — тех событиях, на которые она должна реагировать. Это могут быть нажатия клавиш на клавиатуре, клики мышью.

* Обновление игры

Изменение всего, что должно измениться в течение одного кадра. Если персонаж в воздухе, гравитация должна потянуть его вниз. Если два объекта встречаются на большой скорости, они должны взорваться.

* Рендеринг (прорисовка)

В этом шаге все выводится на экран: фоны, персонажи, меню. Все, что игрок должен видеть, появляется на экране в нужном месте.

* Время

Еще один важный аспект игрового цикла — скорость его работы. Многие наверняка знакомы с термином FPS, который расшифровывается как Frames Per Second (или кадры в секунду). Он указывает на то, сколько раз цикл должен повториться за одну секунду. Это важно, чтобы игра не была слишком медленной или быстрой. Важно и то, чтобы игра не работала с разной скоростью на разных ПК. Если персонажу необходимо 10 секунд на то, чтобы пересечь экран, эти 10 секунд должны быть неизменными для всех компьютеров.

# 2. ОПРЕДЕЛЕНИЕ АЛГОРИТМА РЕШЕНИЯ ЗАДАЧИ

**2.1. Продумываем и описываем правила игры**

Перед тем как начать писать код, необходимо подумать о том, какие правила мы должны учитывать при создании игры. Базовые правила:

* + - цель игры — получить как змейку как можно длиннее;
    - при поглощении специального объекта на игровом экране длина змейки увеличивается на один блок;
    - игрок может двигаться влево, вправо, вверх или вниз;
    - игрок не может двигаться за пределы экрана, при столкновении с границами игра заканчивается;
    - игра заканчивается, когда змейка врезается сама в себя;
    - ещё игра заканчивается, когда пользователь закрывает окно.

2.2. Алгоритмизация

* Создание экрана

Для создания экрана при помощи Pygame нужно воспользоваться функцией display.set\_mode. Также необходимо пользоваться методом init для инициализации экрана в начале кода и методом quit для его закрытия в конце. Метод update используется для применения каких-либо изменений на экране. 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 экран не исчез через секунду после появления надо добавить цикл while, который будет работать до завершения игры. Теперь чтобы при нажатии на крестик наша игра закрылась, нужно использовать функцию «quit».

* Создание змейки

Перед тем как создать змейку, мы 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.

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 черного (по умолчанию) до белого при помощи метода fill.

Для сохранения изменений координат x и y мы создали две новых переменные: x1\_change и y1\_change.

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. Мы будем использовать для этого переменные, чтобы вы могли потом, при случае, легко внести какие-либо изменения в игру.

Теперь мы добавим еды для змейки, и когда она будет ее пересекать, ее размер будет также увеличиваться на один блок. Помимо этого, я внесу небольшие изменения, которые дадут возможность прекращать игру, а также начинать ее заново в случае поражения.

Следующий код будет увеличивать длину змейки после поглощения ею еды. Также, если змейка сталкивается с собственным хвостом, игра заканчивается и выводится сообщение: “ Вы проиграли! Нажмите C-начать сначала или Q-выход “. Длина змейки хранится в списке, а базовые значения заданы в следующем коде.

1. Программный код

import pygame  
import time  
import random  
  
pygame.init()  
  
white = (50, 153, 213)  
yellow = (255, 255, 102)  
black = (0, 0, 0)  
red = (213, 50, 80)  
green = (0, 255, 0)  
blue = (255, 255, 255)  
  
dis\_width = 600  
dis\_height = 400  
  
dis = pygame.display.set\_mode((dis\_width, dis\_height))  
pygame.display.set\_caption('Змейка')  
  
clock = pygame.time.Clock()  
  
snake\_block = 10  
snake\_speed = 7  
  
font\_style = pygame.font.SysFont("timesnewroman", 15)  
  
  
  
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)  
 message("вы проиграли! Нажмите C-начать сначала или Q-выход", red)  
  
 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\_a:  
 x1\_change = -snake\_block  
 y1\_change = 0  
 elif event.key == pygame.K\_d:  
 x1\_change = snake\_block  
 y1\_change = 0  
 elif event.key == pygame.K\_w:  
 y1\_change = -snake\_block  
 x1\_change = 0  
 elif event.key == pygame.K\_s:  
 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)  
  
 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()

ЗАКЛЮЧЕНИЕ

В ходе нашей работы мы столкнулись с несколькими ошибками, которые мешали корректной работе кода. Большинство этих проблем возникало на начальных этапах создания игры, особенно при написании кода для экрана. Важно учитывать все нюансы, следуя пошаговым инструкциям. Первая проблема, заключалась в том, что окно появлялось на несколько секунд и исчезало. Мы решили эту проблему, добавив цикл while. Однако окно по-прежнему не закрывалось при нажатии на крестик. Чтобы исправить это был добавлен оператор quit(), и игровое поле появлялось и закрывалось, как и предполагалось.

Написание игры «Змейка» на языке программирование Python может быть сложным, но полезным опытом для программистов всех уровней. Используя библиотеку Pygame, разработчики могут создать полнофункциональную игру про змей с захватывающей графикой и плавным геймплеем. На протяжении всего процесса можно столкнуться с различными проблемами, такими как обнаружение столкновений и игровая механика. Тем не менее, эти проблемы можно преодолеть, если проявить настойчивость, навыки решения проблем и внимание к деталям. В конце концов, создание игры про змей на Python — отличный способ для разработчиков улучшить свои навыки и глубже понять концепции программирования.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ:

1. Инструкция и выражения для начинающих // PythonRU 2018-2021гг // (электронный ресурс). Режим доступа: https://pythonru.com/uroki/instrukcii-i-vyrazhenija-dlja-nachinajushhih?ysclid=lgntpj92r864883064
2. Как писать игры на Python с Pygame // timeweb cloud 2023 // электронный ресурс). Режим доступа: https://timeweb.cloud
3. PyGame: учебник по программированию // DMCA: dmca 2023г // (электронный ресурс). Режим доступа: https://www.codeflow.site/ru/
4. Pygame. Основы работы // OTUS JOURNAL 2023г // (электронный ресурс). Режим доступа: https://otus.ru/journal/pygame-osnovy-raboty/?ysclid=lgns4e8659578299498