МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО» ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра компьютерной инженерии и моделирования

ИГРА “Snake”.

Курсовая работа

по дисциплине «Программирование» студента 1 курса группы ПИ-202(2)

Григоркевич Александр Сергеевич

направления подготовки 09.03.04 «Программная Инженерия»

|  |  |  |
| --- | --- | --- |
| Научный руководитель  Старший руководитель кафедры компьютерной инженерии и моделирования | (оценка)    (подпись, дата) | Тимофеева С.В. |

Симферополь, 2021

РЕФЕРАТ

Григоркевич А. С. Разработка игры «Snake» в жанре «Головоломки»

Курсовая работа по специальности 09.03.04 Программная инженерия / Кафедра компьютерной инженерии и моделирования Физико-технического института Крымского федерального университета им. В. И. Вернадского. – Симферополь, 2021. Количество страниц – 23.

**Объект исследования** – компьютерная игра в жанре “Головоломки”.

**Предмет исследования** – особенности разработки/методика разработки компьютерной игры в жанре “Головоломки”.

Цель работы – разработать игру «Snake» с использованием языка программирования Python и библиотек Pygame, Pygame\_menu.

Аннотация работы – в работе проведён анализ инструментальных средств для разработки игры в жанре «Головоломки», была определена концепция игрового процесса. Описан процесс разработки. В конечном счёте был получен рабочий программный продукт. Были закреплены навыки проектирования программного продукта.

Python, Pygame, Pygame\_menu, ФУНКЦИИ И КЛАССЫ, АЛГАРИТМЫ, ТЕСТИРОВАНИЕ

ОГЛАВЛЕНИЕ

[Реферат 2](#_TOC_250007)

[Введение 4](#_TOC_250006)

Глава 1. Анализ инструментальных средств для решения поставленной задачи 5

Глава 2. Программная реализация приложения 7

* 1. [Постановка задачи](#_TOC_250005) 7
  2. [Описание алгоритмов](#_TOC_250004) 9
  3. [Описание структур данных 9](#_TOC_250003)

Глава 3. Тестирование программы 18

Глава 4 Визуальная составляющая игры 19

[Заключение](#_TOC_250001) 22

[Литература](#_TOC_250000) 23

ВВЕДЕНИЕ

Для создания проекта использовалась программа Pycharm. Проект создавался методом проб и ошибок, с постоянными изменениями и улучшениями, как в плане производительности, так и в плане удобства для пользователя. Исходя из того, что нет предела совершенству, можно сказать, что программа не окончена и требует дополнений.

“Змейка” появились в начале 1970-х, когда игровые консоли не были достаточно мощными, чтобы отображать трёхмерную графику или видео. Они были ограничены статическими игровыми мирами, которые помещались на один экран. Змейка может перемещаться вверх и вниз по пополю с яблоками, которые при сьедении увеличивают её размер. В 2008 году вышла последняя версия «Змейки» — "Snakes Subsonic". «Сабсоник» являлся продолжением игры «Snakes», но уже для игрового сервиса «N-Gage», доступного на двух десятках устройств Nokia

Даже сегодня, когда трехмерная графика и открытый мир – обычная игра, не вызывающая особого восторга, выходят достаточно интересные проекты в жанре головоломок. Примерами таких игр послужат:

* Wormax.io – цель игры достигнуть вашим червяком как можно большей длины. Для этого нужно собирать еду на карте, избегая столкновения головой с другими червяками.  
  Использование навыков червяка значительно разнообразит игру.
* Pikan.io– Основная цель игры довольно похожа, но в ней, вместо увеличения длины, вам нужно съедать точки, чтобы получить больше пуль в форме треугольника, которыми вы можете стрелять в других игроков.
* Nokia’s Snake- Можно сказать, что это «Пионер» концепции и отец всех этих современных игр, которые в настоящее время присутствуют на рынке. Это игра предыдущего поколения
* Mitosis – Очень похоже на Slither.io, вы можете съесть других противников, а также другие разбросанные предметы, чтобы расти. Главное отличие - это режимы игры, такие как Флаг, Футбол или Захват, что делает игру более захватывающей. Выпущена создателями Worm.is.

В итоге, можно сделать вывод, что головоломки, все еще очень популярны.

Программа имеет возможность считывать изображения и музыку из файла папке Libs. Ознакомившись с условными обозначениями объектов на карте (их полный список указан в файле ниже), пользователь может самостоятельно дополнять и полностью перерабатывать ее по своему желанию. Конец игры наступает, как только змейка сталкивается сама с собой или со стеной.

**Цель работы:**

* разработать игру «Snake»;
* получить опыт разработки конечно программного продукта;
* закрепить навыки проектирования и тестирования проекта.

Для реализации поставленной цели необходимо решить следующие задачи:

* + подключить библиотеки Pygame, Pygame\_menu;
  + создать игровое поле
  + создать игровые объекты
  + написать и внедрить логику игры;
  + создать изображения;
  + создать игровое меню

ГЛАВА 1

**АНАЛИЗ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ДЛЯ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ**

При выборе библиотеки между Pygame и Tkinter я остановился на Pygame.

Pygame является оберткой Python для [библиотеки SDL](https://www.libsdl.org/) , что означает Simple DirectMedia Layer . SDL обеспечивает кроссплатформенный доступ к базовым компонентам мультимедийного оборудования вашей системы, таким как звук, видео, мышь, клавиатура и джойстик. pygame начал жизнь в качестве замены для [зашедшего](http://www.pygame.org/docs/tut/PygameIntro.html) в тупик [проекта PySDL](http://www.pygame.org/docs/tut/PygameIntro.html) . Кроссплатформенная природа SDL pygame означает, что вы можете писать игры и мультимедийные Python-программы для каждой платформы, которая их поддерживает!

Pygame – это библиотека модулей для языка Python, созданная для разработки 2D игр. Также Pygame могут называть фреймворком. Это «игровая библиотека», набор инструментов, помогающих программистам создавать игры. К ним относятся: Графика и анимация, Звук (включая музыку), Управление (мышь, клавиатура, геймпад и так далее)

Также были использованы базовые модули Python: sys, random Sys — это встроенный модуль языка программирования, предоставляющая интерфейс для взаимодействия с базовой операционной системой, под управлением которой работает Python.

Модуль random предоставляет функции для генерации случайных чисел, букв, случайного выбора элементов последовательности.

ГЛАВА 2

**ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ**

* 1. Постановка задачи

Необходимо написать программу, в которой пользователь имеет возможность управлять змейкой, с целью накопить больше очков.

Программа включает в себя:

* создание змейки, яблок, грибов и игрового поля
* возможность управления змейкой в поле с яблоками, грибами (ползти вправо, влево, верх и низ)
* поедание яблок и увеличение змейки в размерах
* поедание грибов и уменьшение змейки в размерах

В игре существуют следующие объекты:

Таблица 1. Игровые объекты

|  |  |
| --- | --- |
| Тип объекта | Вид |
| Голова змейки направленная вниз |  |
| Голова змейки направленная вправо |  |
| Голова змейки направленная влево |  |
| Голова змейки направленная вверх |  |
| Хвост змейки направленный вниз |  |
| Хвост змейки направленный вправо |  |
| Хвост змейки направленный влево |  |
| Хвост змейки направленный вверх |  |
| Тело змейки в горизонтальном положении |  |
| Тело змейки в вертикальном положении |  |
| Поворот тела змейки в первом положении |  |
| Поворот тела змейки во втором положении |  |
| Поворот тела змейки в третьем положении |  |
| Поворот тела змейки в четвертом положении |  |
| Яблоко |  |
| Гриб |  |

* 1. Описание алгоритмов

Управление происходит с помощью клавиатуры ‘W A D S’

А − при нажатии змейка ползет влево.

D − при нажатии змейка ползет вправо.

W - при нажатии змейка ползет вверх.

S - при нажатии змейка ползет вниз.

* 1. Описание структур данных

1. На Рисунке 2.1 показано создание стартового окна, оно состоит из 30 ячеек по 20 пикселей в ширину и из 20 ячеек по 20 пикселей + 60 пикселей для UI элементов. Также были заданы цвета для игрового поля и инициализирован pugame.

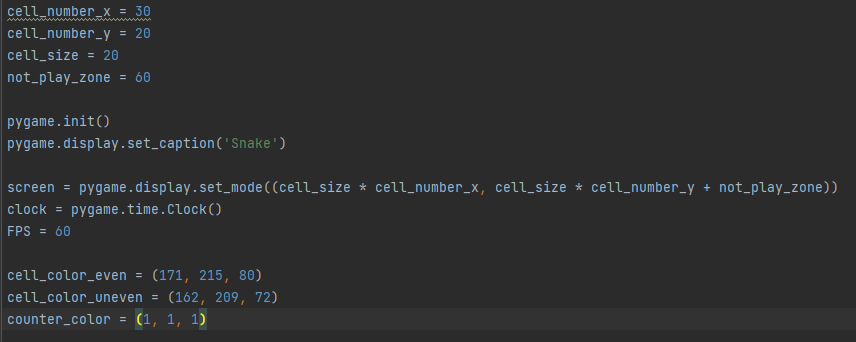


Рисунок 2.1. Размеры окна.

2. Класс изображенный на рисунке 2.2 отвечает за создание игрового меню.

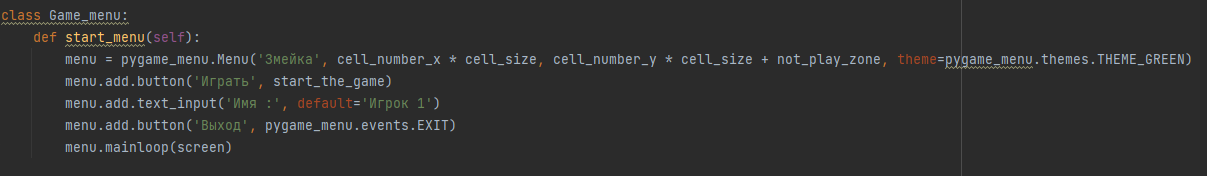


Рисунок 2.2. Класс Game\_menu.

3. Класс на рисунке 2.3 отвечает за создание яблока на игровом поле в случайной точке, его ширина и высота равняются размерам одной ячейки.

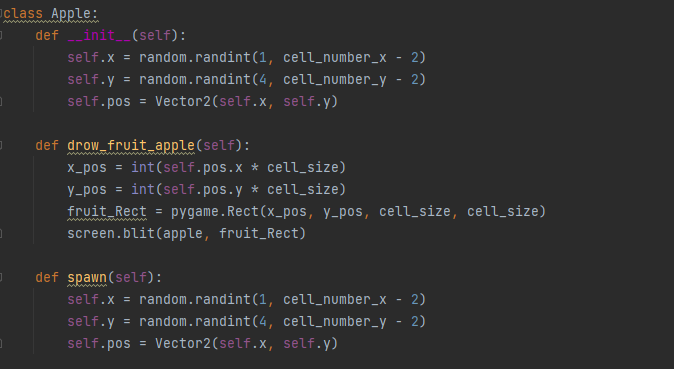


Рисунок 2.3. Класс Apple.

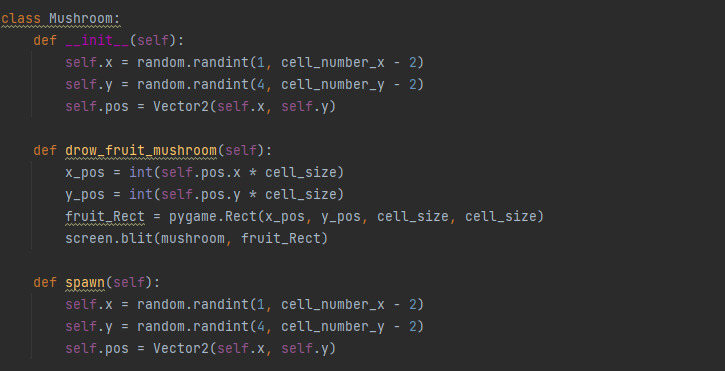
4. Класс на рисунке 2.3 отвечает за создание гриба на игровом поле в случайной точке, его ширина и высота равняются размерам одной ячейки.

Рисунок 2.4. Класс Mushroom.

5. Изображение на рисунке 2.5 отвечает за создание змеи в случайной точке игрового поля.

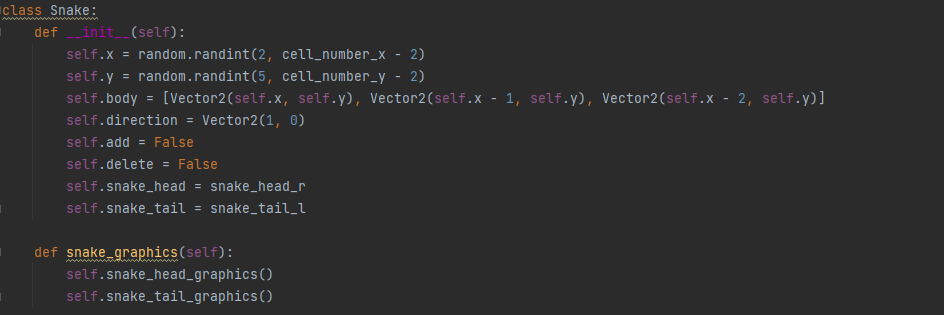


Рисунок 2.5. Создание змеи на игровом поле.

6. Функция drow\_snake на рисунке 2.6. отвечает за рисование змеи на игровом поле.



Рисунок 2.6. Создание змеи на игровом поле.

7. Две функции изображенные на рисунке 2.7. отвечают за графику головы змеи и её хвоста.

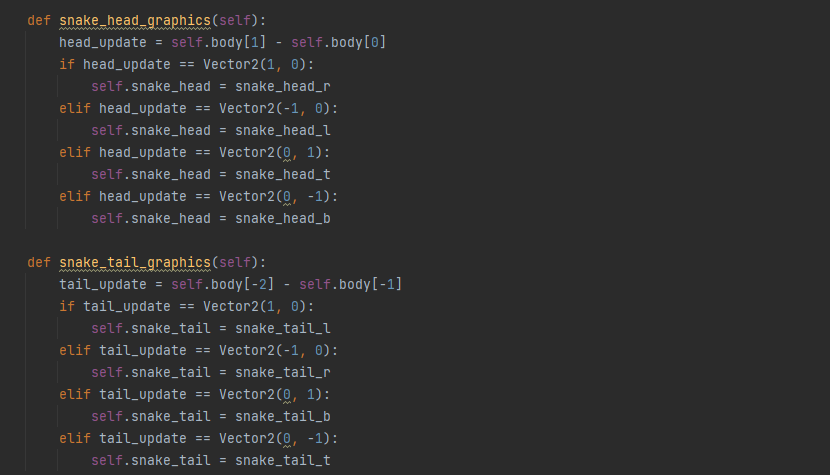


Рисунок 2.7. Создание змеи на игровом поле.

8. На рисунке 2.8. изображены четыре функции, первая отвечает за движение змеи по игровому полю, функция add\_to\_array отвечает за увеличение змеи в размерах, delete\_from\_array отвечает за уменьшение змеи в размерах, spawn отвечает за случайное появление змеи.

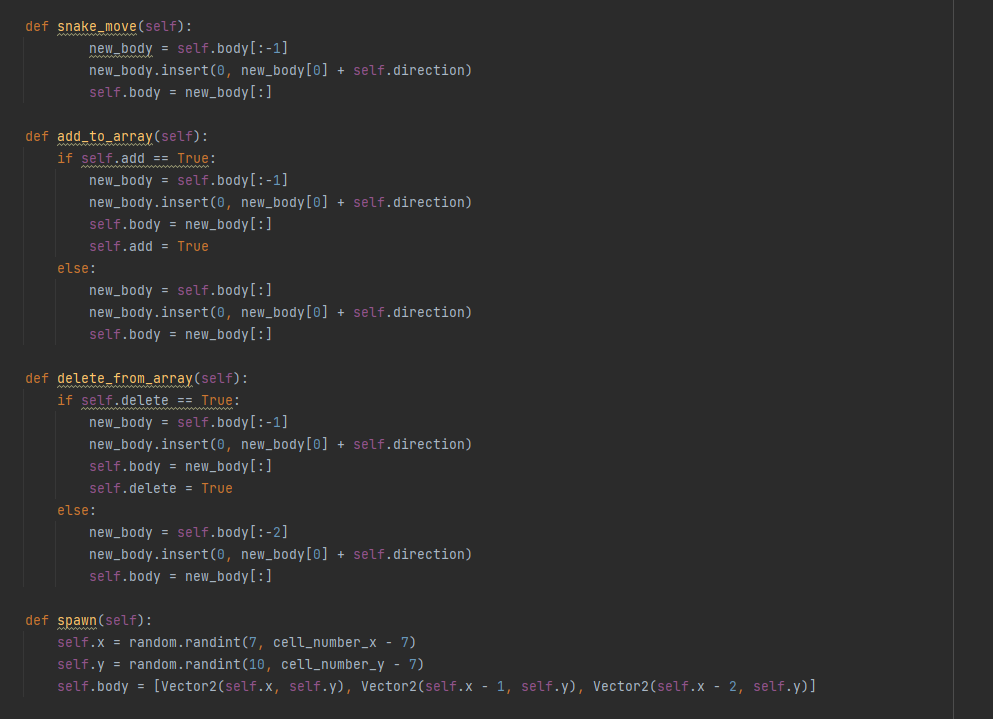


Рисунок 2.8. Движение змеи по полю и взаимодействие с объектами.

9. На рисунке 2.9. изображен класс Grass отвечавший за изображение игрового поля

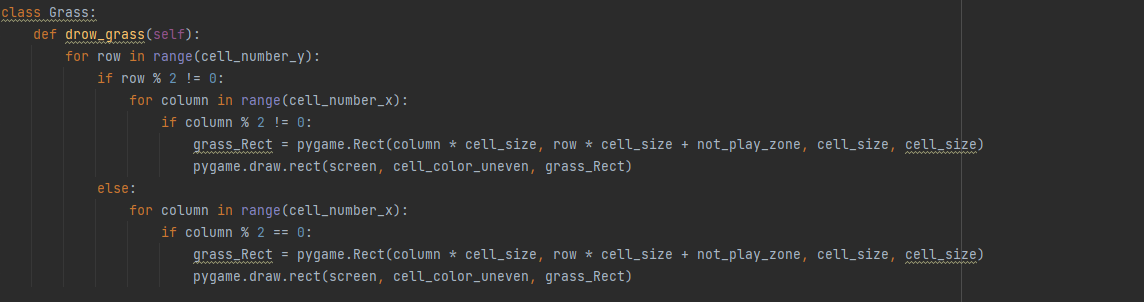


Рисунок 2.9. Класс Grass.

10. Рисунок 2.10. показывает основной класс Main. Функция drow\_elements отвечает за обрисовку все изображений в окне.

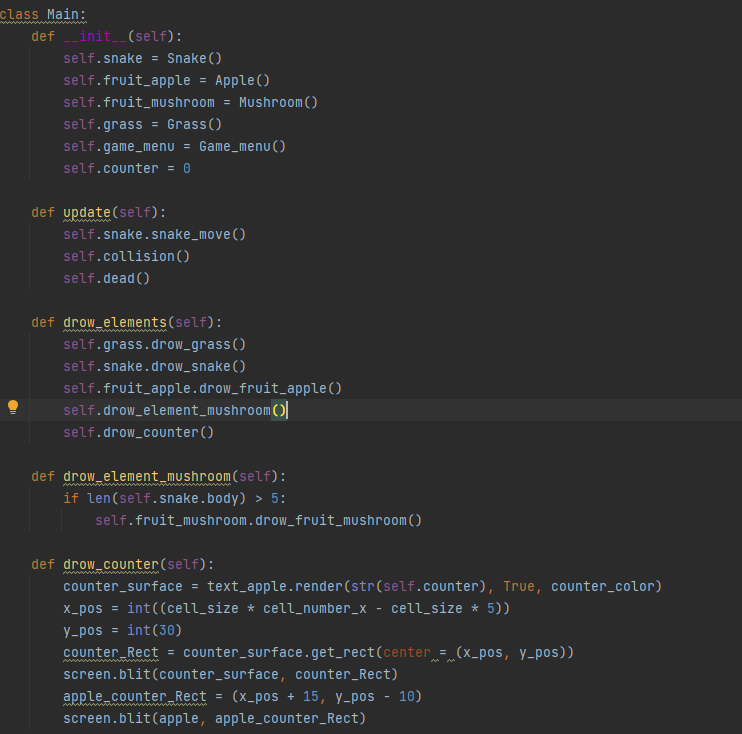


Рисунок 2.10. Класс Grass.

11. На рисунке 2.11. вся графика которая использовалась в проекте.

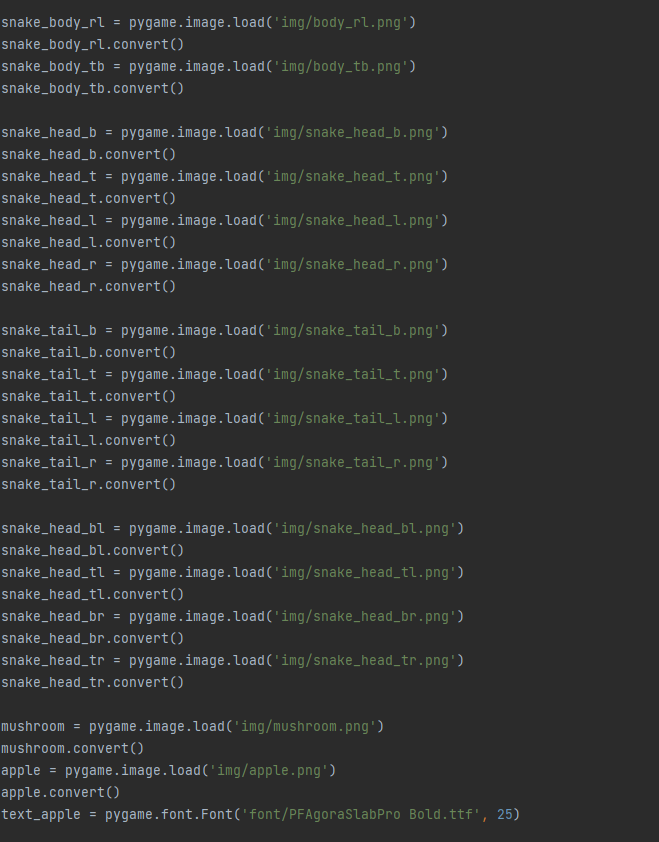


Рисунок 2.11. Вся графика.

12. изображение 2.12. показывает функцию collision которая отвечает за взаимодействие объектов на игровом поле.



Рисунок 2.12. Вся графика.

13. изображение 2.13. показывает функцию, отвечающую за игровой цикл, и управление с помощью клавиатуры.

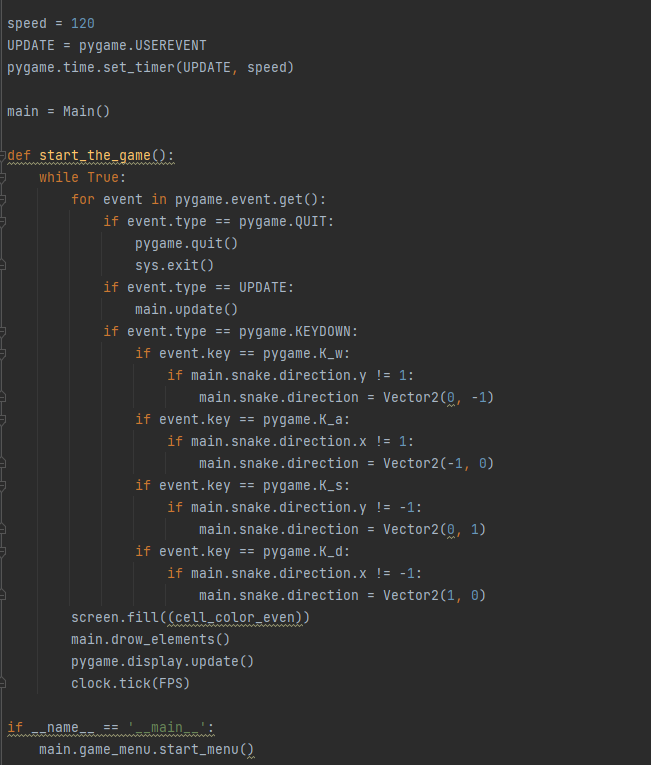


Рисунок 2.13. Функция запуска игры.

ГЛАВА 3

**ТЕСТИРОВАНИЕ ПРОГРАММЫ**

Тестирование приложения, в большей части происходило прямо по ходу написания кода, добавления новых возможностей и функций. Было произведено огромное количество проверок и перезапусков игры, в ходе чего, было

исправлено большое количество ошибок.

При заключительном тестировании были проверены следующие пункты:

1. Управление змейкой
   1. Движение вправо
   2. Движение влево
   3. Движение вниз
   4. Движение вверх
   5. Запуск меню при запуске
2. Коллизия
   1. Столкновения с яблоками
   2. Столкновения с самой змейкой
   3. Столкновение с грибами
   4. Столкновения со стенами

Итог:

* + - На данный момент, явных ошибок не обнаружено
    - Отсутствуют недочеты в плане геймплея

ГЛАВА 4

**ВИЗУАЛЬНАЯ СОСТАВЛЯЮЩАЯ ИГРЫ**

1. На Рисунке 4.1. изображено меню игры, которое запускается при запуске приложения и после смерти игрового персонажа.

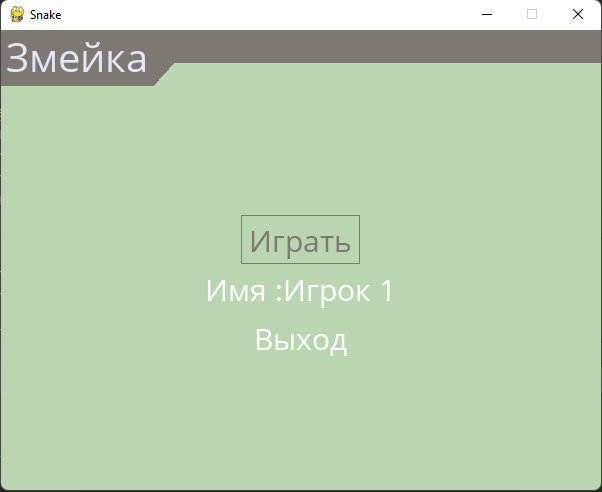


Рисунок 4.1. Игровое меню

1. На рисунке 4.2. игрок появляется в случайной точке на игровом поле.

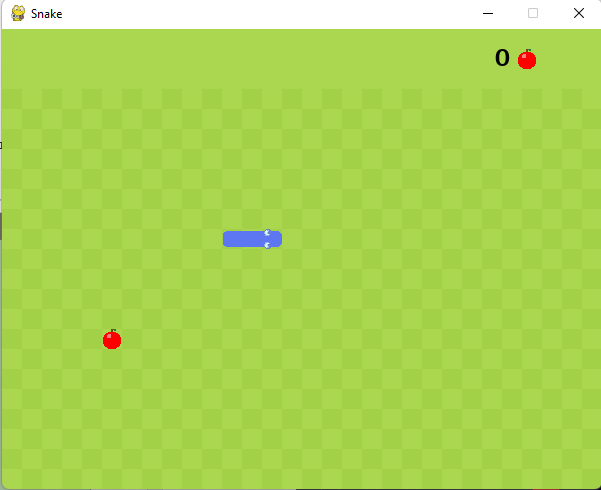


Рисунок 4.2. Начало игры

1. На рисунке 4.3. изображен процесс игры, с появлением яблок и грибов на экране и работой счетчика.

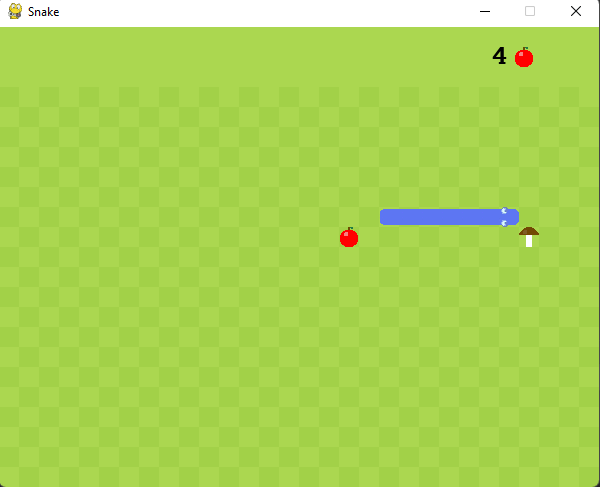


Рисунок 4.3..Игровой процесс.

ЗАКЛЮЧЕНИЕ

В результате работы была создана игра на языке программирования Python с использованием библиотеки Pygame и Pygame\_menu. Получены хорошие практические знания в построении алгоритмов, работой с массивами, функциями и циклами. В итоге создана игра в жанре «Головоломки», в будущем планирую внедрять разные дополнительные возможности для игры дальнейшей доработке и улучшении.

ЛИТЕРАТУРА

* 1. Документация random.
  2. Документация pygame.
  3. Документация pygame\_meny.
  4. Документация sys.