

## Реализация игры «Змейка» на языке программирования Python

*Маринчук Александр Сергеевич*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

### Аннотация

В данной статье рассмотрена реализация игры «Змейка» на языке программирования Python с использованием библиотеки tkinter.

**Ключевые слова:** Python, Pyinstaller, игра, tkinter.

## The game « Snake» implementation using Python programming language

*Marinchuk Alexander Sergeevich*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

This article discusses the implementation of the game "Snake" in the Python programming language using the tkinter library.

**Keywords:** Python, Pyinstaller, игра, tkinter.

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

В статье Т. Р. Змыговой, Д. С. Приемских, А. Е. Авдюшиной рассмотрены возможности использования языка программирования Python для разработки игр в жанре платформер. Рассматриваются возможности и перспективы дополнения игры новыми уровнями и сценариями [1]. А. О. Кизянов и Д. В. Лучанинов рассмотрели разработку 2D игры «Alien Invasion» с помощью языка программирования Python с использованием дополнительной библиотеки Pygame [2]. Статья С.В. Дрянцева описывает возможность применения компьютерных игр для изучения языков программирования, на примере компьютерной игры Minecraft и языка программирования Python [3].

Целью исследования является процесс разработки игры на языке программирования Python. В разработке были использованы такие средства как PyInstaller и библиотека tkinter.

Перейдем непосредственно к созданию самой игры. Для начала создадим главное окно игры, по которому будет передвигаться змейка в поисках еды (рис. 1).

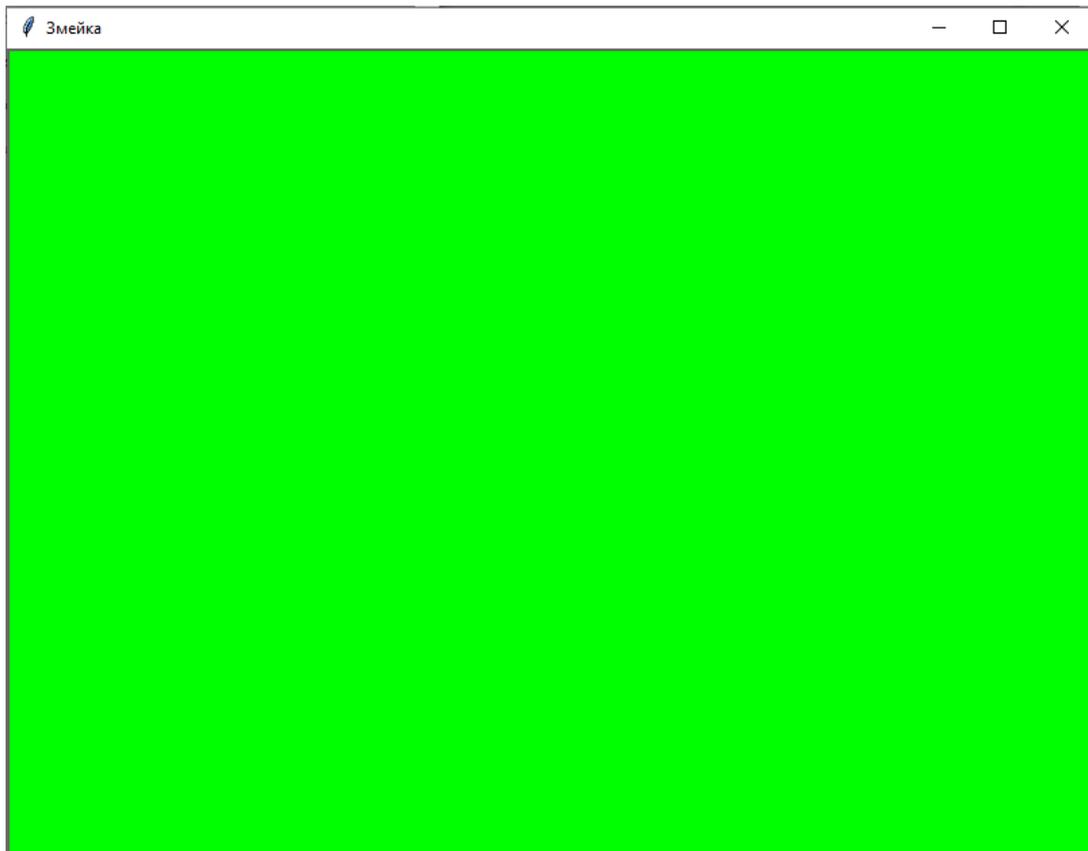


Рисунок 1. Главное окно

Затем создадим непосредственно саму змейку с помощью библиотеки `tkinter`, которая поможет создать блоки, из которых состоит змейка, еду, а также поможет отслеживать передвижение змейки (рис. 2).

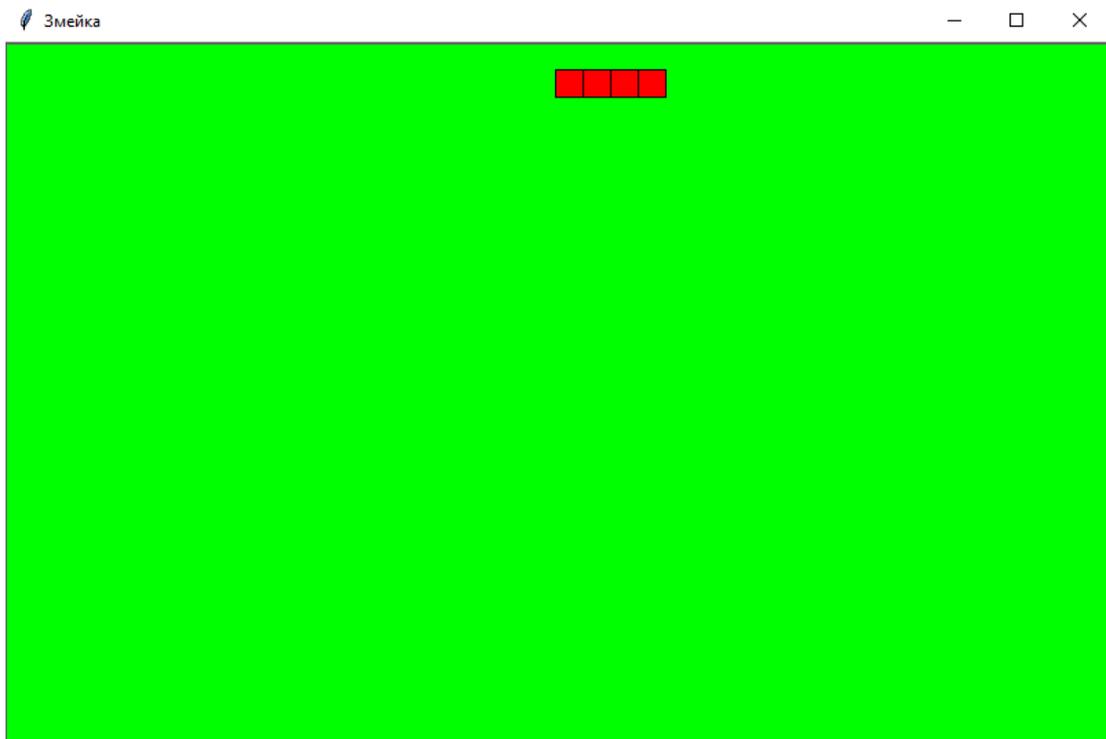


Рисунок 2. Создание тела змейки

Пропишем возможные действия змейки: отследим нажатие той или иной клавиши и сообщим змейке, в каком направлении ей следует двигаться (рис. 3).

```
class Snake(object):
    def __init__(self, segments):
        self.segments = segments

        # список доступных направлений движения змейки
        self.mapping = {"Down": (0, 1), "Up": (0, -1),
                        "Left": (-1, 0), "Right": (1, 0) }

        # изначально змейка двигается вправо
        self.vector = self.mapping["Right"]

    def move(self):
        """ Двигает змейку в заданном направлении """

        # перебираем все сегменты кроме первого
        for index in range(len(self.segments)-1):
            segment = self.segments[index].instance
            x1, y1, x2, y2 = c.coords(self.segments[index+1].instance)
            # задаем каждому сегменту позицию сегмента стоящего после него
            c.coords(segment, x1, y1, x2, y2)

        # получаем координаты сегмента перед "головой"
        x1, y1, x2, y2 = c.coords(self.segments[-2].instance)

        # помещаем "голову" в направлении указанном в векторе движения
        c.coords(self.segments[-1].instance,
                 x1 + self.vector[0]*SEG_SIZE,
                 y1 + self.vector[1]*SEG_SIZE,
                 x2 + self.vector[0]*SEG_SIZE,
                 y2 + self.vector[1]*SEG_SIZE)

    def change_direction(self, event):
        """ Изменяет направление движения змейки """

        # event передаст нам символ нажатой клавиши
        # и если эта клавиша в доступных направлениях
        # изменяем направление
        if event.keysym in self.mapping:
            self.vector = self.mapping[event.keysym]

    def add_segment(self):
        """ Добавляет сегмент змейке """

        # определяем последний сегмент
        last_seg = c.coords(self.segments[0].instance)

        # определяем координаты куда поставить следующий сегмент
        x = last_seg[2] - SEG_SIZE
        y = last_seg[3] - SEG_SIZE

        # добавляем змейке еще один сегмент в заданных координатах
        self.segments.insert(0, Segment(x, y))
```

Рисунок 3. Код обработки движения змейки

Далее создадим еду, за которой будет ползти наша змейка (рис. 4).

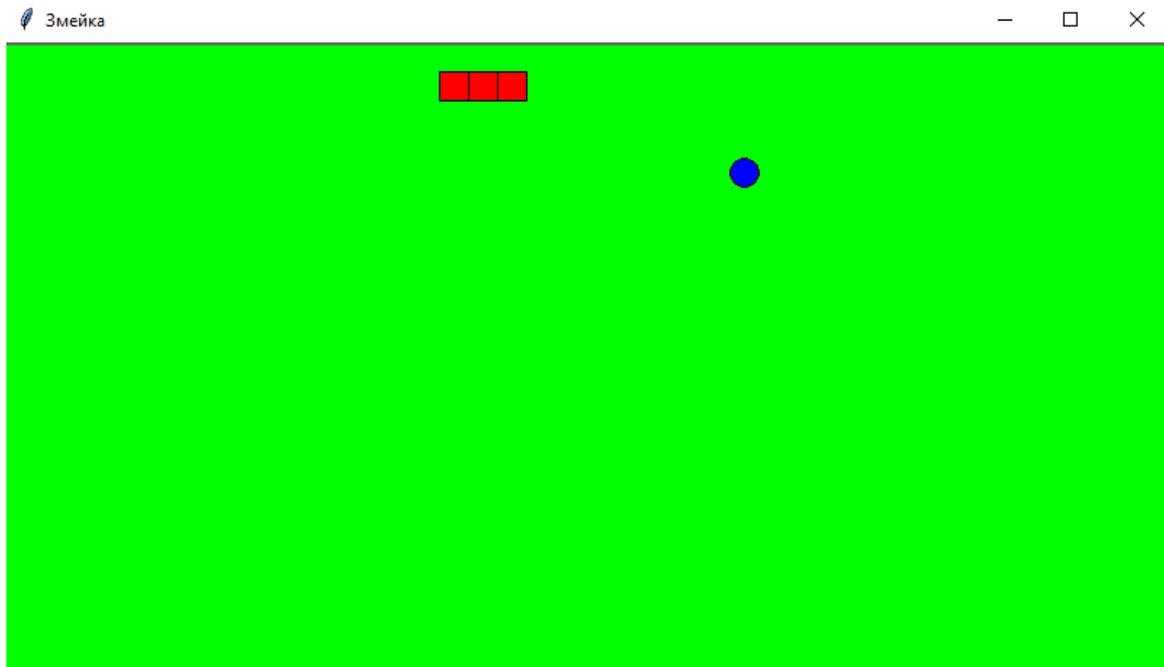


Рисунок 4. Создание еды

После проигрыша будем предлагать перезапустить пользователю игру (рис 5.)

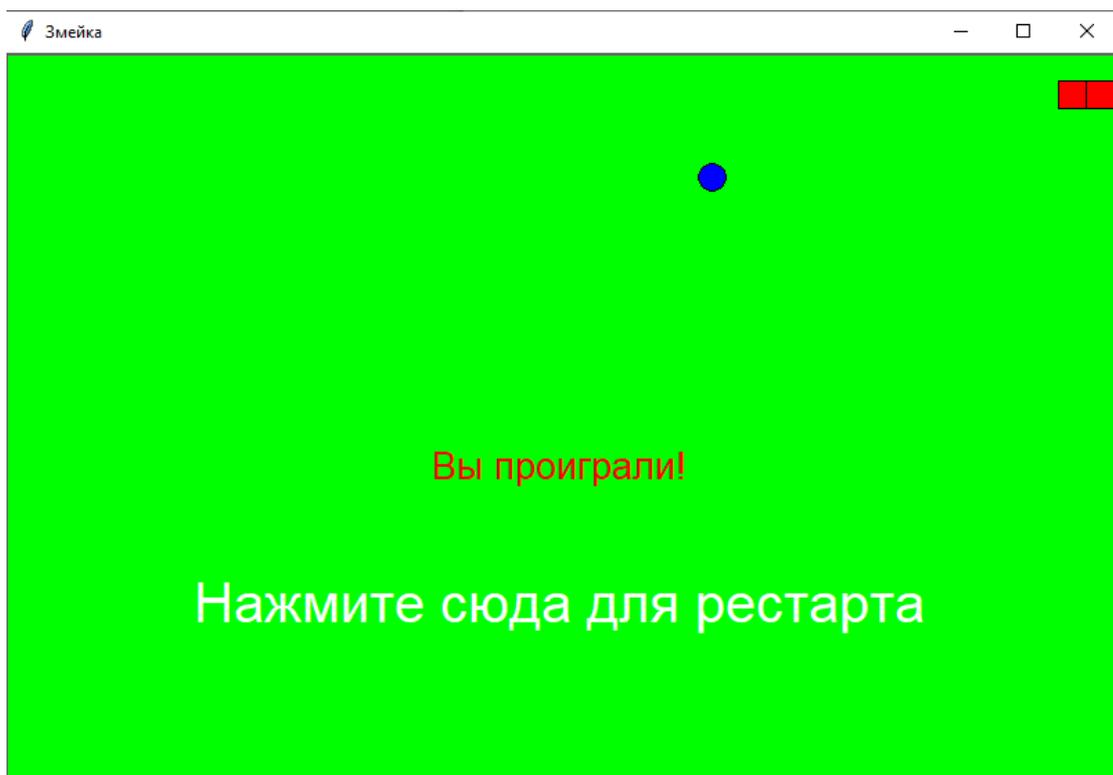


Рисунок 5. Рестарт игры

Осталось собрать нашу игру в один exe-файл. Так как язык программирования Python интерпретируемый, то ему нужен интерпретатор на машине конечного пользователя, чтобы запустить исходные файлы на Python. Решение этой проблемы возможно с помощью сторонней библиотеки

Pyinstaller, она позволяет переводить код Python в самостоятельный файл расширения exe, что позволяет запускать его на разных машинах семейства Windows. При таком виде компиляции программа переводится в множество вспомогательных файлов.

Модуль PyInstaller [4] служит для перевода кода на языке Python в исполняемый файл. Для его установки достаточно написать в консоли операционной системы (Windows/Linux/Mac) команду `pip install Pyinstaller`. На этом установка закончена. После нужно перейти в папку, где установлен Python, найти папку Scripts в ней найти файл Pyinstaller.exe. После этого необходимо перетянуть файл кодов Python на этот файл. В результате будет создана папка dist рядом с исходным кодом, в ней будет exe файл со всеми библиотеками.

В ходе статьи удалось реализовать игру «Змейка» на языке программирования Python. Были рассмотрены библиотека tkinter и модуль PyInstaller для успешного создания игры. В будущем планируется регистрация и добавление таблицы рекордов пользователей.

### **Библиографический список**

1. Змыгова Т. Р., Приемских Д. С., Авдюшина А. Е. Создание игр в жанре платформер с Python // НАУКА XXI ВЕКА: ТЕХНОЛОГИИ, УПРАВЛЕНИЕ, БЕЗОПАСНОСТЬ. Курган: Курганский государственный университет, 2017. С. 343-352.
2. Кизянов А. О., Лучанинов Д. В. Реализация 2D игры "Alien Invasion" на языке программирования Python // ПОСТУЛАТ, 2016.
3. С.В. Дрянцев Применение компьютерных игр в обучении программированию на примере Minecraft // ТЕОРИЯ И ПРАКТИКА СОВРЕМЕННОЙ НАУКИ. 2018. №11. С. 413-418.
4. Модуль Pyinstaller для перевода кода Python в самостоятельный файл. URL: <http://www.pyinstaller.org> (дата обращения: 09.12.2019)