

Разработка игры на Python: создание механики передвижения персонажа

Беликов Андрей Геннадьевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье был рассмотрен процесс создания механики передвижения персонажа на языке Python. В исследовании использовалась программа PyCharm. В результате работы была разработана механика передвижения персонажа для игры на языке Python.

Ключевые слова: Геймдев, PyCharm, Python

Python game development: creating character movement mechanics

Belikov Andrey Gennadievich

Sholom-Aleichem Priamursky State University

Student

Abstract

In this article, the process of creating a character movement mechanics in Python was considered. The PyCharm program was used in the study. As a result of the work, the mechanics of character movement for the game in Python were developed.

Keywords: Game Dev, PyCharm, Python

Актуальность создания игры на языке Python обусловлена тем, что данный процесс позволяет связать изучение языка и развлечение.

Цель данной статьи - создать персонажа игрока для игры на языке Python.

Для создания проекта была рассмотрена статья Мялова М.И. в которой рассмотрен процесс создания и разработки игры на языке программирования Python [1]. Так же П.Дж.Вандер описал возможности языка программирования Python [2]. А. Д. Тулегулов рассмотрел и показал примеры и методы обучения и реализации на языке программирования Python [3].

Для того чтобы игра была больше похожа на оригинал сначала изменим размер экрана игры, для этого изменим значение на 700 (Рис.1).

```
pygame.init()
screen = pygame.display.set_mode((700, 800))
pygame.display.set_caption("Космические защитники")
bg_color = (0, 0, 0)
gun = Gun(screen)
```

Рисунок 1 –Изменение игрового экрана

Следующим шагом игровые события, которые будут происходить непосредственно с главным героем на игровом экране мы перемести в отдельный файл. Для этого создаём её в меню и даём соответствующее название «controls.py».

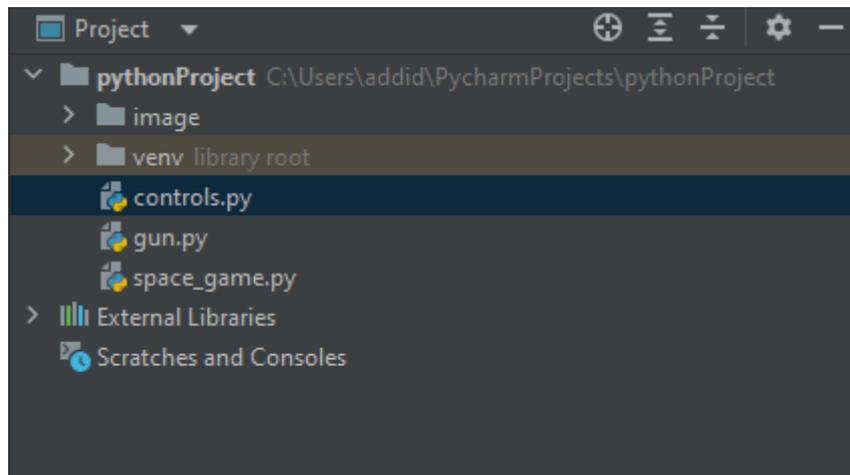


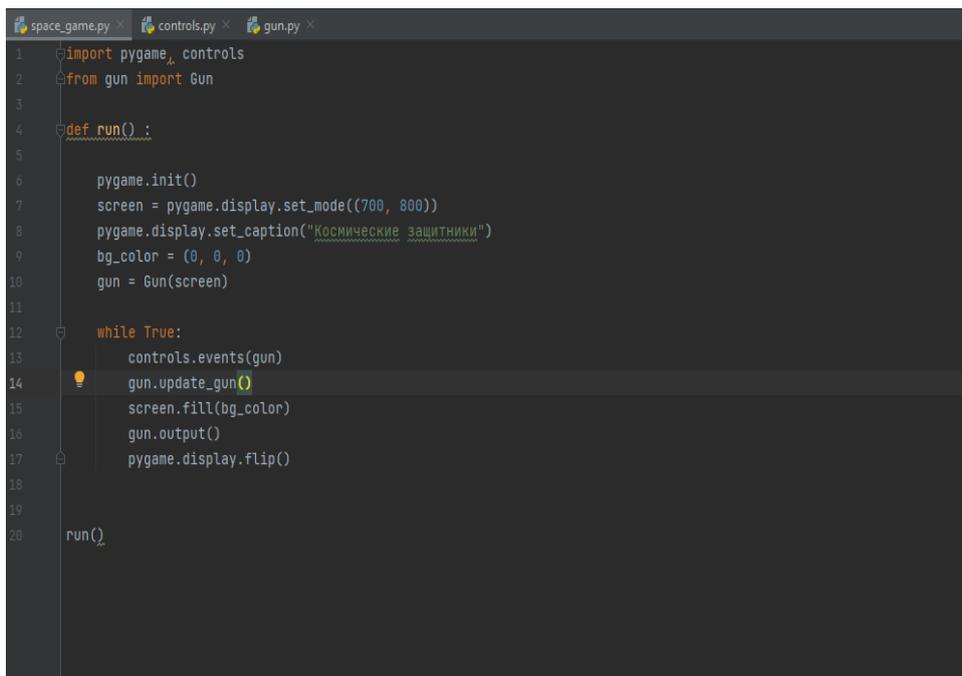
Рисунок 2 – Созданный файл управления персонажем

Следующим действием импортируем файлы передвижения. И прописываем функцию, отслеживающую передвижение и изменение положения игрового персонажа по полю для этого прописываем следующие команды (Рис.3.)

```
space_game.py x controls.py x gun.py x
1 import pygame
2 from gun import Gun
3
4 def run():
5
6     pygame.init()
7     screen = pygame.display.set_mode((700, 800))
8     pygame.display.set_caption("Космические защитники")
9     bg_color = (0, 0, 0)
10    gun = Gun(screen)
11
12    while True:
13
14        screen.fill(bg_color)
15        gun.output()
16        pygame.display.flip()
17
18    run()
```

Рисунок 3 – Импорт файлов управления

В коде прописываем какой объект будет перемещаться, в данном случае это пушка. Данные изменения вносятся в код (Рис.4).

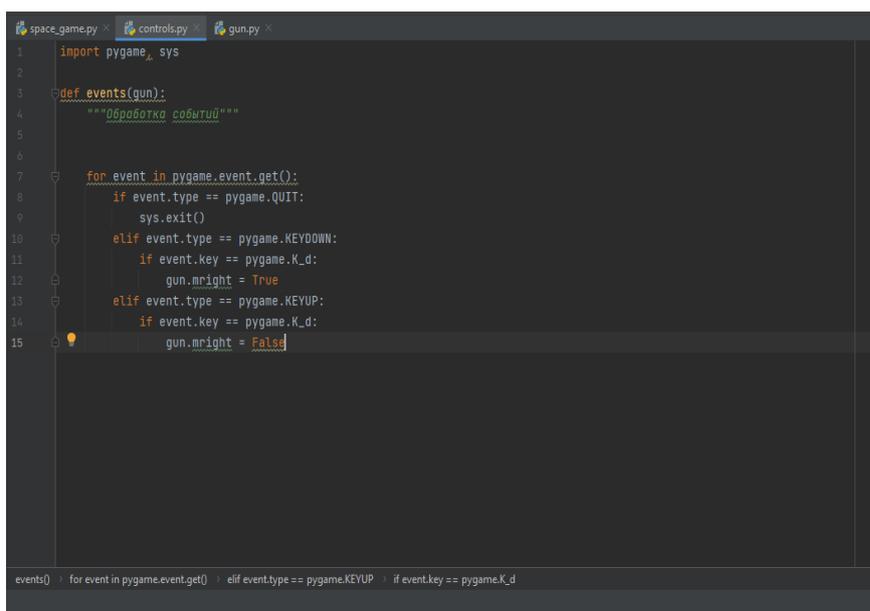


```
1 import pygame, controls
2 from gun import Gun
3
4 def run():
5
6     pygame.init()
7     screen = pygame.display.set_mode((700, 800))
8     pygame.display.set_caption("Космические защитники")
9     bg_color = (0, 0, 0)
10    gun = Gun(screen)
11
12    while True:
13        controls.events(gun)
14        gun.update_gun()
15        screen.fill(bg_color)
16        gun.output()
17        pygame.display.flip()
18
19
20 run()
```

Рисунок 4 – Обозначение объекта передвижения

При перемещении главного героя прописываются условия, а именно действия игрока, которые он совершает для того чтобы пушка как объект переместилась. Здесь описан код, при котором координата x перемещается.

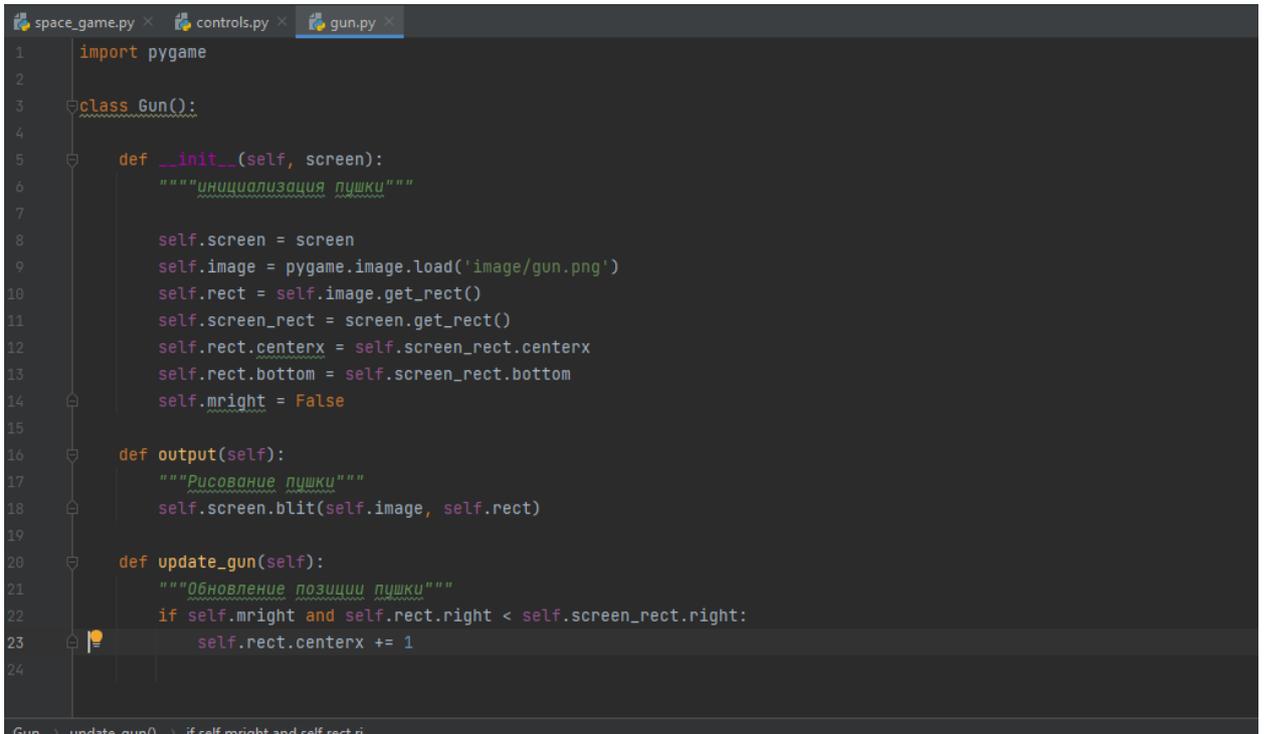
Далее создаём логическую переменную, при которой передвижение будет выполняться на долгое зажатие клавиши и прекращаться (Рис.5).



```
1 import pygame, sys
2
3 def events(qun):
4     """Обработка события"""
5
6
7     for event in pygame.event.get():
8         if event.type == pygame.QUIT:
9             sys.exit()
10
11        elif event.type == pygame.KEYDOWN:
12            if event.key == pygame.K_d:
13                gun.mright = True
14
15        elif event.type == pygame.KEYUP:
16            if event.key == pygame.K_d:
17                gun.mright = False
```

Рисунок 5 – Переменная передвижения объекта

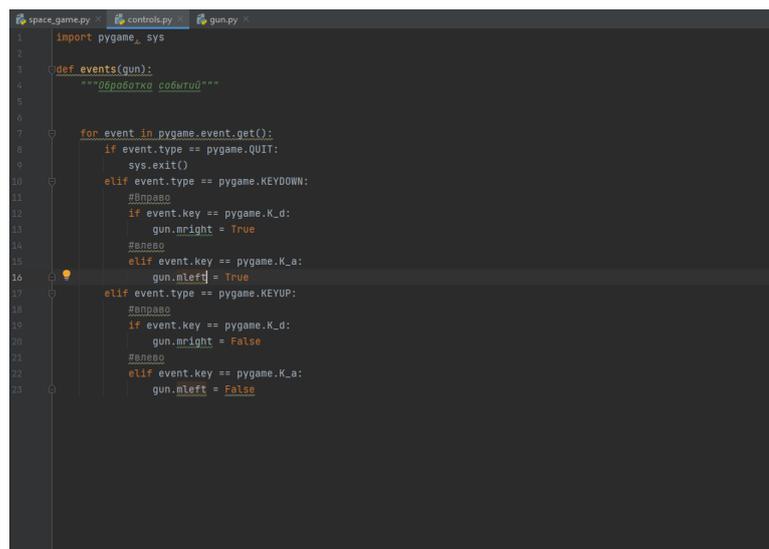
Следующим шагом пропишем обновление позиции пушки для того чтобы игра смогла считывать местонахождение объекта и изменять его передвижение при нажатии клавиши (Рис.6).



```
1 import pygame
2
3 class Gun():
4
5     def __init__(self, screen):
6         """инициализация пушки"""
7
8         self.screen = screen
9         self.image = pygame.image.load('image/gun.png')
10        self.rect = self.image.get_rect()
11        self.screen_rect = screen.get_rect()
12        self.rect.centerx = self.screen_rect.centerx
13        self.rect.bottom = self.screen_rect.bottom
14        self.mright = False
15
16    def output(self):
17        """Рисование пушки"""
18        self.screen.blit(self.image, self.rect)
19
20    def update_gun(self):
21        """Обновление позиции пушки"""
22        if self.mright and self.rect.right < self.screen_rect.right:
23            self.rect.centerx += 1
24
```

Рисунок 6 – Обновление позиции пушки

Далее прописываем передвижение пушки при нажатой клавиши и останавливаем его если клавиша отжата, в данном случае мы указали что движение вправо будет с помощью клавиши «D», а перемещение клавиши влево будет с помощью клавиши «A» (Рис.7).



```
1 import pygame, sys
2
3 def events(gun):
4     """Обработка событий"""
5
6     for event in pygame.event.get():
7         if event.type == pygame.QUIT:
8             sys.exit()
9         elif event.type == pygame.KEYDOWN:
10            #нашао
11            if event.key == pygame.K_d:
12                gun.mright = True
13            #нашао
14            elif event.key == pygame.K_a:
15                gun.mleft = True
16        elif event.type == pygame.KEYUP:
17            #нашао
18            if event.key == pygame.K_d:
19                gun.mright = False
20            #нашао
21            elif event.key == pygame.K_a:
22                gun.mleft = False
23
```

Рисунок 7 – Передвижение пушки

Для обновления положения пушки и для того чтобы она не выходила за экран игрового поля прописываем положения правого и левого края пушки и ставим его меньше самого игрового поля (Рис.8).

```
import pygame

class Gun():
    def __init__(self, screen):
        """инициализация пушки"""
        self.screen = screen
        self.image = pygame.image.load('image/gun.png')
        self.rect = self.image.get_rect()
        self.screen_rect = screen.get_rect()
        self.rect.centerx = self.screen_rect.centerx
        self.rect.bottom = self.screen_rect.bottom
        self.mright = False
        self.mleft = False

    def output(self):
        """Рисование пушки"""
        self.screen.blit(self.image, self.rect)

    def update_gun(self):
        """Обновление позиции пушки"""
        if self.mright and self.rect.right < self.screen_rect.right:
            self.rect.centerx += 1
        if self.mleft and self.rect.left > 0:
            self.rect.centerx -= 1
```

Рисунок 8 – Ограничение передвижения пушки по игровому полю

Проверяем изменение положения и включаем старт нашей игры, как видно положение изменяется (Рис.9).

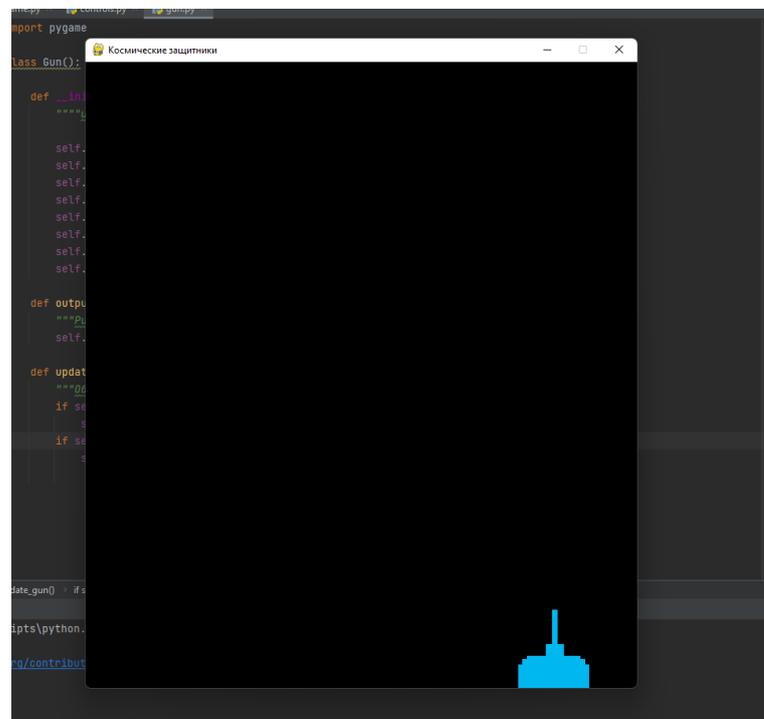


Рисунок 9 – Готовое изображение перемещения объекта по игровому полю

В данной статье был рассмотрен процесс создания механики передвижения персонажа на языке Python. В исследовании использовалась программа PyCharm. В результате работы была разработана механика передвижения персонажа для игры на языке Python.

Библиографический список

1. Мялова М.И, Пазин М.А. Разработка пользовательского интерфейса на основе Python для автоматического выбора алгоритма кластеризации данных. В сборнике: Цифровые инфокоммуникационные технологии. сборник научных трудов. Ростов-на-Дону, 2021. С. 77-81.
2. Вандер П. Дж. Python для сложных задач: наука о данных и машинное обучение. СПб.: Питер, 2018. 576 с.
3. Тулегулов А. Д., Ешпанов В. С., Исмаилов А. Практический опыт обучения методам интеллектуального анализа на платформе Python Anaconda // Цифровое образование: новая реальность: Сборник материалов Всероссийской научно-методической конференции с международным участием, Чебоксары, 16 ноября 2020 года. Чебоксары: Общество с ограниченной ответственностью «Издательский дом «Среда», 2020. С. 197-200.