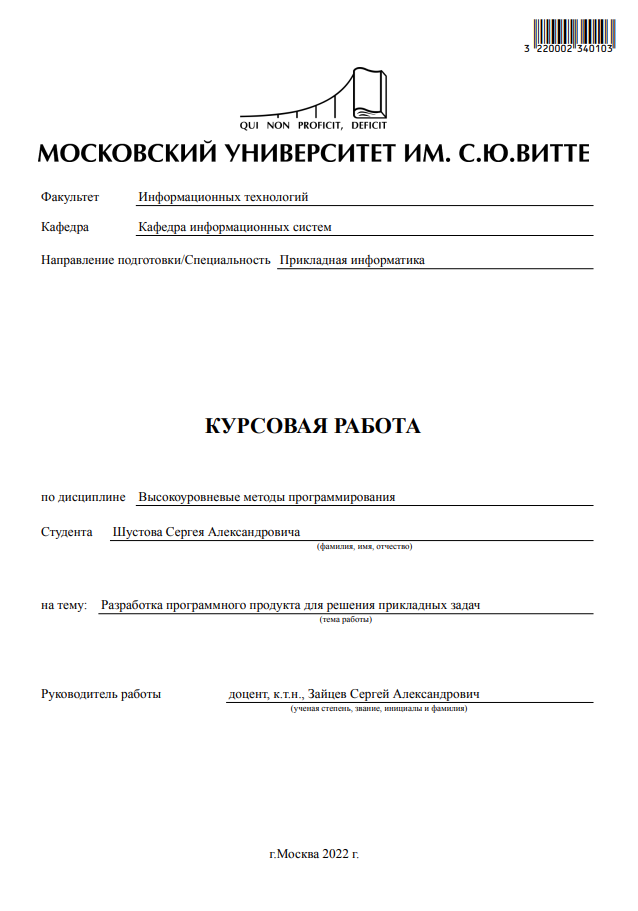
****

**ВВЕДЕНИЕ**

В условиях быстро развивающейся современной высокотехнологичной экономики актуальной задачей является разработка различных автоматизированных систем. Они позволяют наиболее эффективно использовать различные современные средства информационной вычислительной техники и решать прикладные задачи. Для успешного решения таких задач нужно уметь использовать высокоуровневые средства разработки и декомпозировать задачу.

В данной работе решаются задачи высокоуровневого программирования, согласно индивидуальному заданию.

Актуальность курсовой работы связана с наличием большого количества прикладных задач, которые необходимо эффективно решать с помощью инструментов высокоуровневых методов программирования.

*Проект выполняется совместно с Жиденко А. В.*

Оглавление

[**1.** **ПОСТАНОВКА ЗАДАЧИ КУРСОВОЙ РАБОТЫ** 3](#_Toc102592640)

[**1.1.** **Исходные данные к задаче курсовой работы.** 3](#_Toc102592641)

[1.1.1. Язык программирования python 3](#_Toc102592642)

[1.1.2. Основные библиотеки pygame, time, os, numpy, sys, math 3](#_Toc102592643)

[1.1.3. Игра сделана в формате 2D. Игра имеет функциональное меню с кнопками 'Start', 'Continue', 'Options', 'About', 'Quit', из которых все функциональны и стабильно работают. Функция 'Options' включает в себя настройку полноэкранного режима и возможность вкл/выкл 'Показать ФПС'. Кнопки Start, Continue, Quit - работают соответственно. Сама игра сделана в стиле известной игры 'Mario' : ГГ(Главный герой) может передвигатьсяь влево/вправо, прыгать/падать. Задача игрока добраться до финиша на определённом уровне. Когда игрок пройдёт все уровни - он выиграет. 3](#_Toc102592644)

[1.2. Анализ исходных данных курсовой работы 3](#_Toc102592645)

[1.2.1. Грамотно распределить время работы в команде для более координированной работы с кодом. 3](#_Toc102592646)

[1.2.2. Для более удобной работы игры необходимо создать главное меню игры: присваиваем кажой кнопке определённый индекс, добавляем обновление кнопки при нажатии(анимация),добавляем проверку события для нажатой и отпущенной кнопки, добавляем задний фон и анимацию фона и прочее. 3](#_Toc102592647)

[1.2.3. Для удобства пользователя необходимо добавить функцию полного экрана и возможность убрать/показать FPS в меню 'Options' 4](#_Toc102592648)

[1.2.4. При написании кода понадобится импортировать библиотеки: Pygame, sys, os, time, numpy(Некоторые будут убраны после оптимизации кода) 4](#_Toc102592649)

[1.2.5. Будет добавлен файл data/engine.py - в котором будет прописаны: базовая физика и механика игры, прорисовка и отображение игрового мира, подгрузка файлов. А так же это файл отвечает за полную настройку игры. - Ядро игры. 4](#_Toc102592650)

[1.2.6. Необходимо нарисовать все необходимые изображения и включить их в код, а именно: кнопки(в нажатом и отпущенном виде), игрок(в статичном положении и в движении), фон игры(в игре и в состоянии паузы), эффекты частиц и прочего, игровые объекты(платформы, цепи, стены) 4](#_Toc102592651)

[1.3. Выбор и обоснование необходимых библиотек и среды разработки. 4](#_Toc102592652)

[1.3.1. Поскольку в методических указаниях к выполнению курсовой работы указано использование языка программирования Python, среды разработки PyCharm, а также стандартных библиотек и модулей стандартной библиотеки, то для решения заданий курсовой работы были выбраны следующие средства: Язык программирования Python3 – согласно рекомендациям; Интегрированная среда разработки PyCharm – согласно рекомендациям; 4](#_Toc102592653)

[1.3.2. Pygame— библиотека Python для упрощения написания игрового кода на движке Python 5](#_Toc102592654)

[1.3.3. Sys— библиотека Python для прямого взаимодействия с интерпритатором. 5](#_Toc102592655)

[1.3.4. Os— библиотека Python для получения наглядной информации о работоспособности кода 5](#_Toc102592656)

[1.3.5. Time— библиотека Python для более удобной работы с секундами игры, FPS, выводом его на экран и прочее. 5](#_Toc102592657)

[1.3.6. Math— библиотека Python для более удобной работы с математическими функциями кода, и для составления более сложных вычислений. 5](#_Toc102592658)

[1.3.7. Numpy— библиотека Python необходима для составления высокоуровневых математических функций, составления массивов и пр. 5](#_Toc102592659)

[**1.4.** **Выводы по 1 главе.** 5](#_Toc102592660)

[1.4.1. Ссылка на проект Github: https://github.com/Shustovson/game-project 5](#_Toc102592661)

[**2.** **РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА** 6](#_Toc102592662)

[**2.1.** **Создание движка(ядра) игры.** 6](#_Toc102592663)

[**2.1.1.** **Для этого импортируем библиотеки: pygame, math, os, sys, в файл data/engine а так же из библиотеки pygame импортируем locals.** 6](#_Toc102592664)

[**2.1.2.** **Добавляем класс Font - Массив, который содержит в себе названия переменных, символов , которыми пользуется игра для корректного чтения кода.** 6](#_Toc102592665)

[2.1.3. В файле engine.py храниться ядро игры(движок) а конкретнее: Физика, коллизия, шрифты, рендер и сущность игры. А конкретнее: 6](#_Toc102592666)

[**2.1.4.** **Класс physics\_obj – Объект физики** 6](#_Toc102592667)

[**2.1.5.** **entity:** 7](#_Toc102592668)

1. **ПОСТАНОВКА ЗАДАЧИ КУРСОВОЙ РАБОТЫ**
   1. **Исходные данные к задаче курсовой работы.**

Формулировка: Создание 2D игры на языке программирования Python:

* + 1. Язык программирования python
    2. Основные библиотеки pygame, time, os, numpy, sys, math

* + 1. Игра сделана в формате 2D. Игра имеет функциональное меню с кнопками 'Start', 'Continue', 'Options', 'About', 'Quit', из которых все функциональны и стабильно работают. Функция 'Options' включает в себя настройку полноэкранного режима и возможность вкл/выкл 'Показать ФПС'. Кнопки Start, Continue, Quit - работают соответственно. Сама игра сделана в стиле известной игры 'Mario' : ГГ(Главный герой) может передвигатьсяь влево/вправо, прыгать/падать. Задача игрока добраться до финиша на определённом уровне. Когда игрок пройдёт все уровни - он выиграет.
  1. Анализ исходных данных курсовой работы

Для выполнения задания нужно проделать следующие этапы:

* + 1. Грамотно распределить время работы в команде для более координированной работы с кодом.
    2. Для более удобной работы игры необходимо создать главное меню игры: присваиваем кажой кнопке определённый индекс, добавляем обновление кнопки при нажатии(анимация),добавляем проверку события для нажатой и отпущенной кнопки, добавляем задний фон и анимацию фона и прочее.
    3. Для удобства пользователя необходимо добавить функцию полного экрана и возможность убрать/показать FPS в меню 'Options'
    4. При написании кода понадобится импортировать библиотеки: Pygame, sys, os, time, numpy(Некоторые будут убраны после оптимизации кода)
    5. Будет добавлен файл data/engine.py - в котором будет прописаны: базовая физика и механика игры, прорисовка и отображение игрового мира, подгрузка файлов. А так же это файл отвечает за полную настройку игры. - Ядро игры.
    6. Необходимо нарисовать все необходимые изображения и включить их в код, а именно: кнопки(в нажатом и отпущенном виде), игрок(в статичном положении и в движении), фон игры(в игре и в состоянии паузы), эффекты частиц и прочего, игровые объекты(платформы, цепи, стены)
  1. Выбор и обоснование необходимых библиотек и среды разработки.

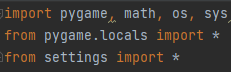
* + 1. Поскольку в методических указаниях к выполнению курсовой работы указано использование языка программирования Python, среды разработки PyCharm, а также стандартных библиотек и модулей стандартной библиотеки, то для решения заданий курсовой работы были выбраны следующие средства: Язык программирования Python3 – согласно рекомендациям; Интегрированная среда разработки PyCharm – согласно рекомендациям;
    2. Pygame— библиотека Python для упрощения написания игрового кода на движке Python
    3. Sys— библиотека Python для прямого взаимодействия с интерпритатором.
    4. Os— библиотека Python для получения наглядной информации о работоспособности кода
    5. Time— библиотека Python для более удобной работы с секундами игры, FPS, выводом его на экран и прочее.
    6. Math— библиотека Python для более удобной работы с математическими функциями кода, и для составления более сложных вычислений.
    7. Numpy— библиотека Python необходима для составления высокоуровневых математических функций, составления массивов и пр.
  1. **Выводы по 1 главе.**

Изучив полученный материал из главы 1, можно сделать вывод, что задание включает наиболее часто используемые прикладные задачи высокоуровневого программирования – это построение приложений с графическим пользовательским интерфейсом, использование алгоритмов, работа в наиболее популярной среде разработки, ставшей де-факто стандартом для разработчиков на языке программирования Python.

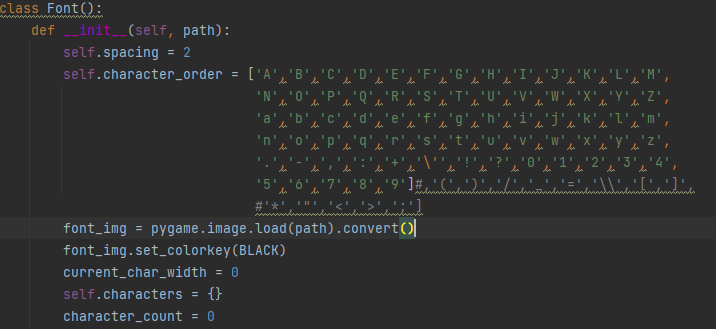
Успешное выполнения задания будет свидетельствовать о полноте усвоенного материала в рамках дисциплины «Высокоуровневые методы программирования» и умение решать задачи низкой и средней сложности.

* + 1. Ссылка на проект Github: https://github.com/Shustovson/game-project

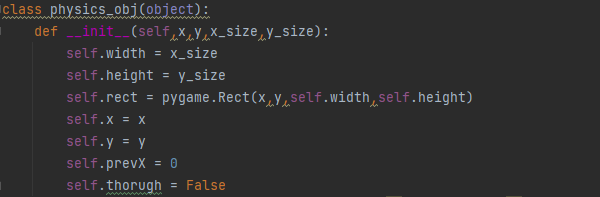
1. **РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА**
   1. **Создание движка(ядра) игры.** 
      1. **Для этого импортируем библиотеки: pygame, math, os, sys, в файл data/engine а так же из библиотеки pygame импортируем locals.**



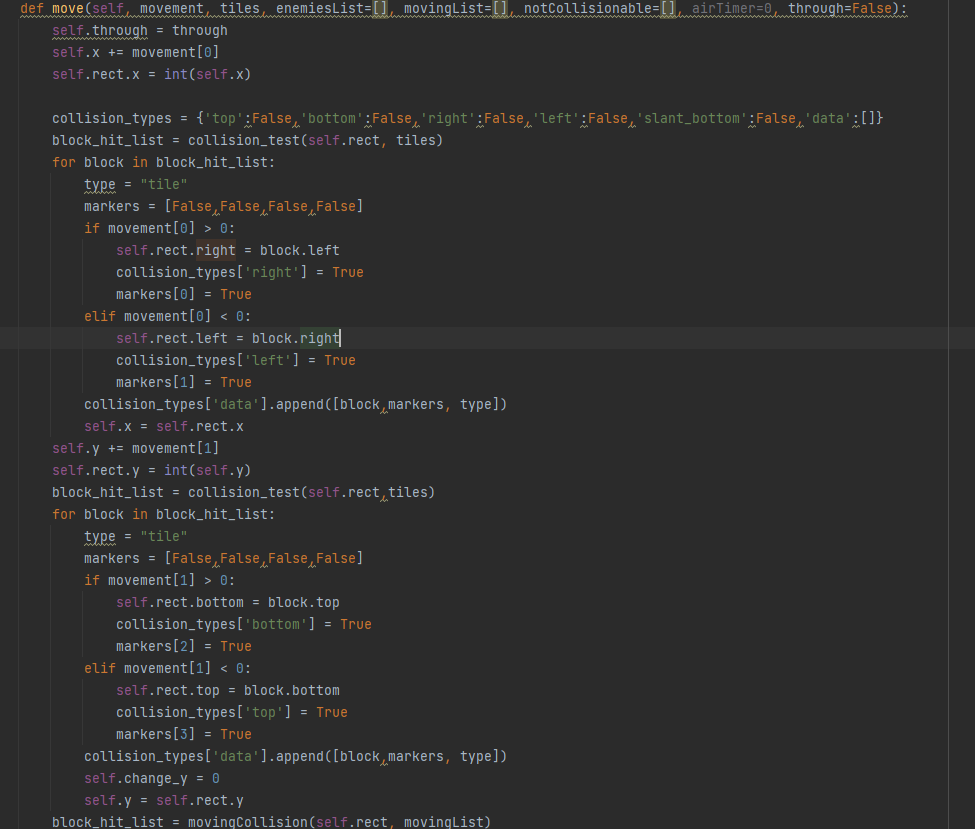
* + 1. **Добавляем класс Font - Массив, который содержит в себе названия переменных, символов , которыми пользуется игра для корректного чтения кода.**



* + 1. В файле engine.py храниться ядро игры(движок) а конкретнее: Физика, коллизия, шрифты, рендер и сущность игры. А конкретнее:
    2. **Класс physics\_obj – Объект физики**

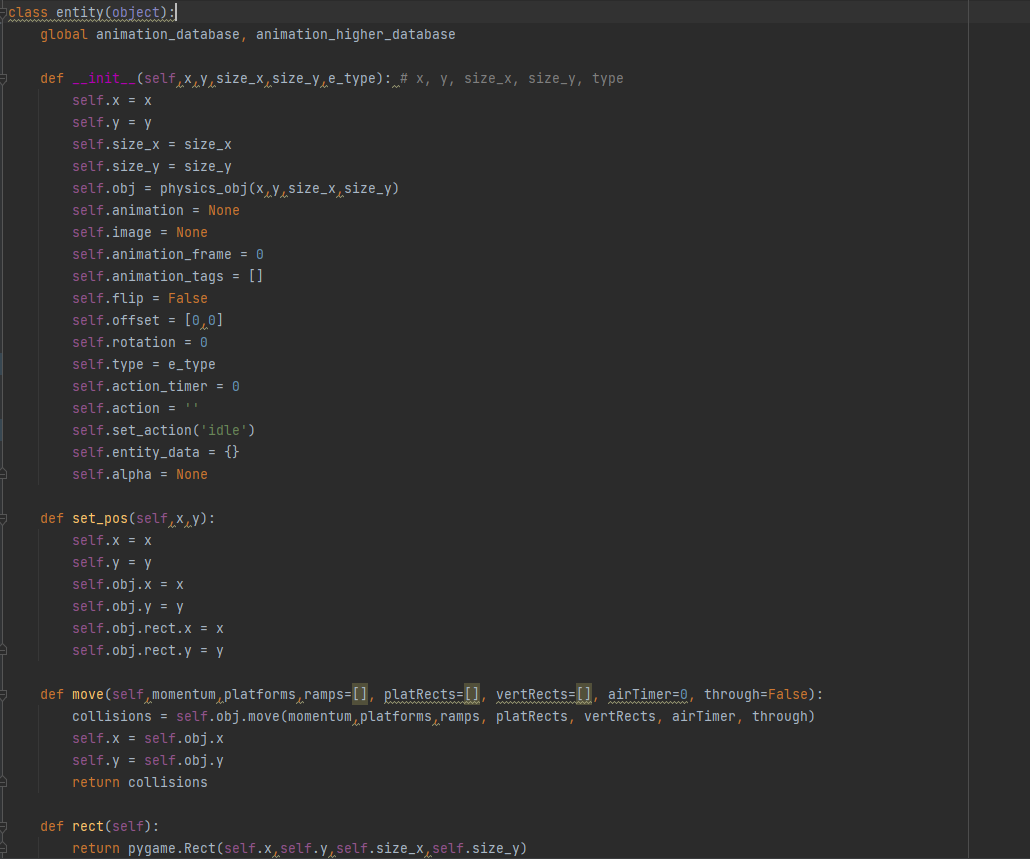


* + - 1. Функция инициализации



Устанавливается значения физики объектов и обрабатывает коллизию объектов

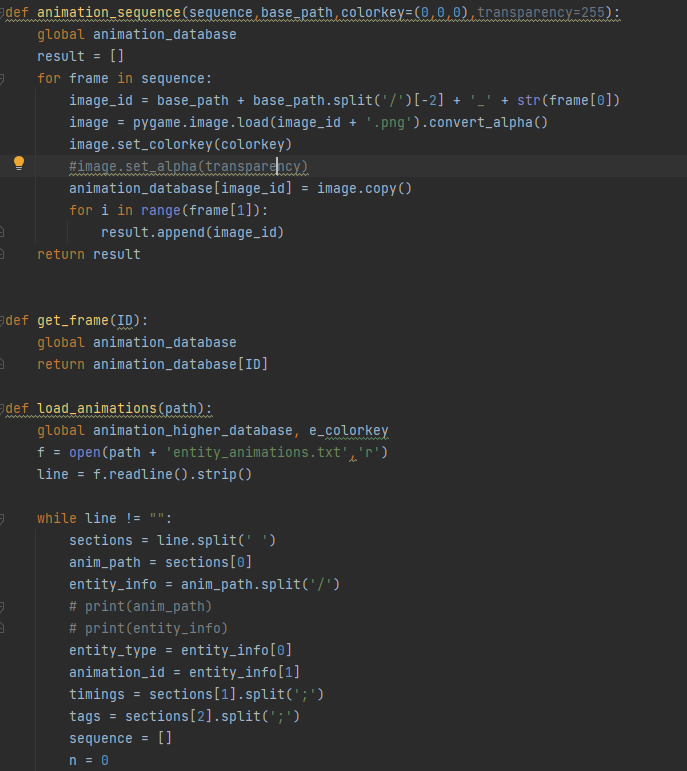
* + 1. **entity:**

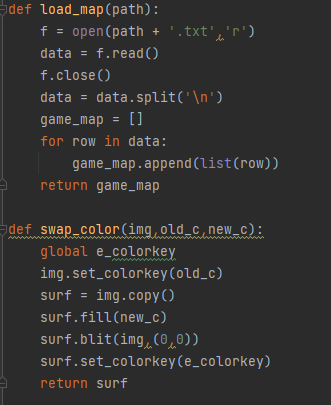


* + - 1. Класс, отвечающий за спрайты в игре. Класс выполняет функции инициализации, установки позиции объектов, движения, отрисовки, обновления, центрирования, таймера анимации и т.д.
    1. particle:

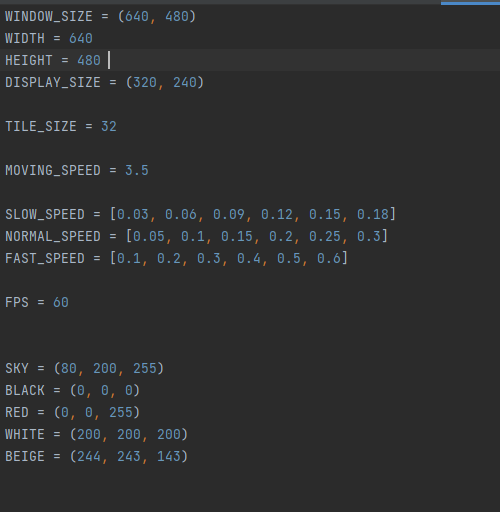


* + - 1. Класс, где происходит инициализация, отрисовка и обновление рендера частиц. Кроме того, в файле прописаны отдельные от класса функции последовательности анимации кадров, и загрузки анимации из файлов

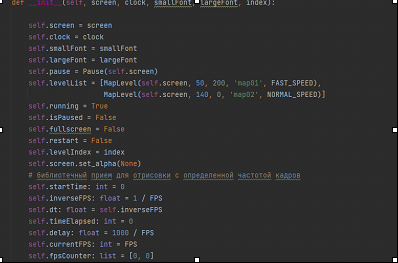


И функции отвечающие за погрузку карты и смену цвета

* + 1. В файле settings.py задаются глобальные константные переменные:



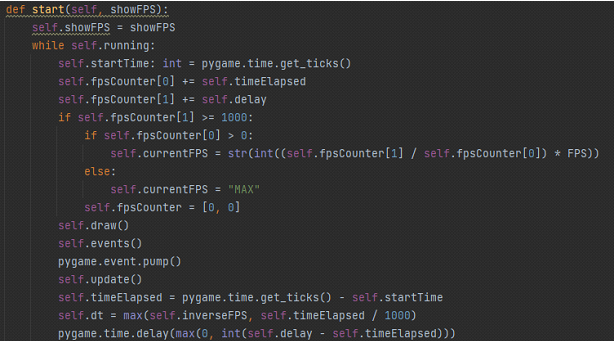
* + 1. В файл Game.py был добавлен классы, отвечающие за Игровой процесс:

Game:

Инициализация класс происходит с параметрами screen , clock , smalFont, largeFont, index в которых прописывается информация о времени, параметрах окна, индекс уровней и параметрах шрифта

* + - 1. Кроме того, внутри класса задаются функции:

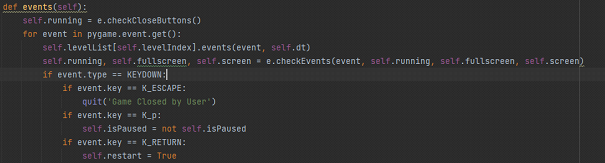
start:

Запуск игры с бесконечным стартовым циклом, задающим стартовое время, отображение FPS и его хранением.

draw:

Происходит отрисовка по индексу уровня

events:



Функция обработки событий нажатия на клавиши

update:

Функция обновления экрана и проверки работы функций FPS а так же проверка паузы и рестарта

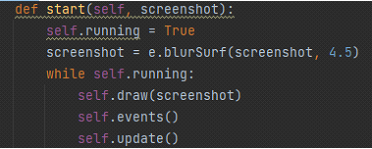
* + 1. Pause:

Инициализируем класс с параметрами экрана и заднего фона

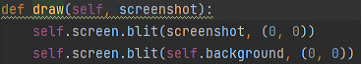


Кроме того, внутри класса задаются функции:

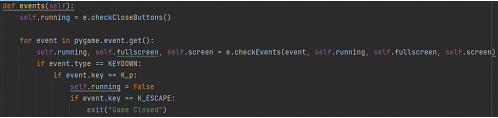
start:

Останавливаем игру во время нажатия и оставляет на экране скриншот с блюром пока игра запущена.

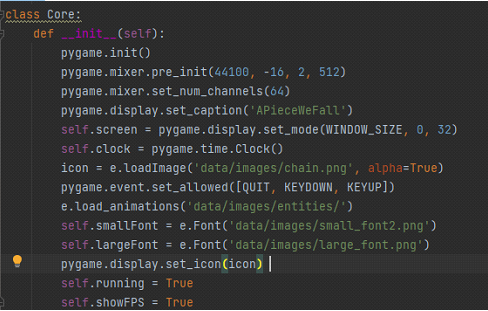
draw:

Функция отрисовки вызываемая функцией start

events:

Проверка событий нажатия клавиш

* + 1. Core:

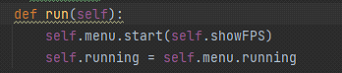
Класс создаёт окно, называет его, подгружает файлы шрифтов и картинок и анимаций, а так же задаёт клавиши читаемые программой в главном меню

Кроме того, внутри класса задаются функции:

new:

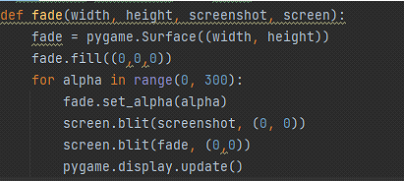
Функция создания окна

run:

Запуск окна.

Кроме того, в этом файле прописываются отдельная от класса функции:

fade:

Функция отвечает за затемнение для окна паузы

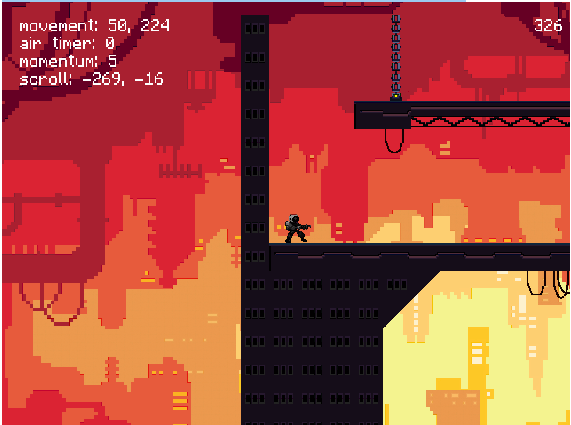
* 1. **Вывод к главе 2:**

⦁ В языке программирования python совмещает в себе множество библиотек с помощью, которых разработчик программного кода может решать прикладные задачи любого уровня сложности.

⦁ Правильное структурирование функций программного кода значительно упрощают дальнейшую разработку и повышает читабельность кода для разработчиков.

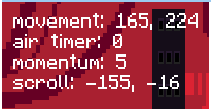
⦁ Среда разработки Pycharm с помощью своих встроенных функций и огромного количества разнообразных инструментов, упрощает написание кода для разработчика и программиста.

1. **Тестирование**
   1. Начало новой игры. Игрок появляется на начальной платформе, откуда начинается прохождение уровня.

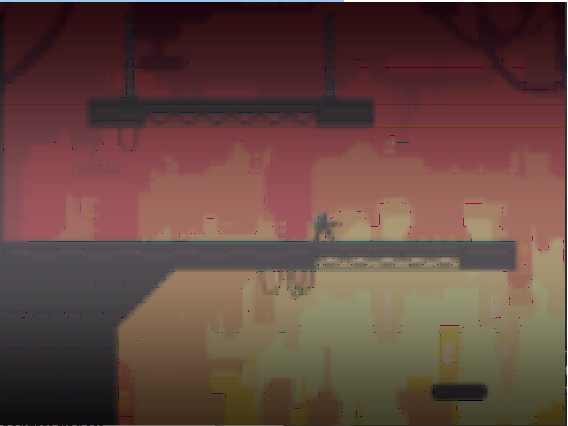


В левой верхней части экрана отобржён датчик:

координат(movement), время в полёте(air moment), ускорение падения(momentum), движение(scroll)

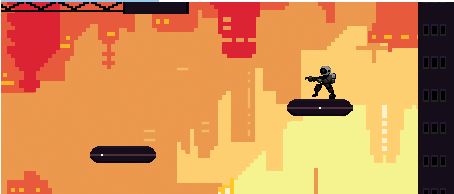


Пользователь управляет игроком с помощью кнопок WASD(или стрелок) остальные аспекты были не реализованы(стрельба, быстрый бег, ходьба). Так же пользователь может нажать на паузу, нажав на кнопку P. И польностью закрыть игру, нажав ESC.



* 1. Игровые объекты.

В игре присутсвтуют стены, а так же 2 основных вида платформ - Движущиеся и не движущиеся.

Так же присутсвуют колючки, при падении на которые, игрок возвращается на последнюю контрольную точку.

Пройдя уровень игрок приходит к порталу, откуда попадает на новый уровень и цикл повторяется:

Конца у этой игры нет - так как было добавлено только 2 уровня. Тем самым, дойдя до портала на втором уровне игра закроется.

* 1. **Вывод**

Основной целью данной курсовой работы являлось написание функциональной игры. Осуществлена попытка решения предложенных задач, в ходе решения которых были углублены знания языка Python и его инфраструктуры, были реализованы алгоритмы и графические пользовательские интерфейсы программ, использованы некоторые дополнительные библиотеки. Помимо этого, было уделено внимание тестированию и отладке программного кода, что является неотъемлемой частью разработки программного обеспечения.

Получившиеся в результате разработки программы успешно выполняются интерпретатором языка Python.

1. **Список используемой Литературы**
   1. Любанович Б. Простой Python. Современный стиль программирования. – С-П.:Питер, - 2019
   2. Туториал по написанию кода на Python[Электронный ресурс], - <https://pythonru.com/uroki/biblioteka-pygame-chast-1-vvedenie>
   3. Пример написания игры на Pygame[Электронный ресурс], - <https://coderslegacy.com/python/pygame-rpg-building-the-base/>
   4. Содержательный туториал по библиотеке Pygame[Электронный ресурс], - <https://habr.com/ru/post/588605/>