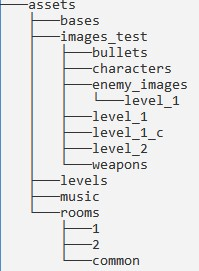
The Assets directory in games is a crucial part of the game development process, as it contains all the assets used in the game. These assets can include 3D models, textures, audio files, animations, and scripts, among other thingsSprite is commonly known as a small, animated graphic character or object that moves independently on a display

Содержимое папки assets (только папки)



Теперь расскажу о каждой поподробнее.

Начнем с простого, с папки music, в ней, как и говорит название хранится музыка.

В images\_test хранятся все спрайты, рассмотрим каждую директорию по отдельности.

1. Bullets в нем хранятся изображения пуль
2. Characters в нем хранятся изображения персонажей, которые после правильного отображения воспроизводят анимации
3. Enemy images в нем хранятся изображения врагов для уровня
4. level\_1, level\_2 содержат изображения для уровней
5. weapons содержат изображения оружия

В rooms содержатся папки 1, 2 в них хранятся конфигурации для каждого уровня.

ФОТО!!!!

1. 1.enemy\_room.special.room – в нем хранится конфигурация для комнаты с врагами №1.
2. 1.treasure\_room.special.room – в нем хранится конфигурация для комнаты с сундуком №1.
3. 2.enemy\_room.special.room – в нем хранится конфигурация для комнаты с врагами №2.
4. portal\_room.room – комната с порталом
5. signed\_pictures.special.sprite – список в виде
6. sprites.road – айдишники для дорог
7. sprites.wall – айдишники для стен
8. start\_room.room – стартовая комната

Структура room файлов была инспирирована с формата PGM.

A PGM file is a grayscale image file saved in the portable **gray** **map** (PGM) **format** and encoded with one or two bytes (8 or 16 bits) per pixel. It contains header information and a grid of numbers representing different shades of **gray**, from black (0) to white (up to 65,536). PGM files are typically stored in ASCII text **format** but also have a binary representation.

1. A "magic number" for identifying the file type. A pgm image's magic number is the two characters "P5".
2. Whitespace (blanks, TABs, CRs, LFs).
3. A width, formatted as ASCII characters in decimal.
4. Whitespace.
5. A height, again in ASCII decimal.
6. Whitespace.
7. The maximum gray value (Maxval), again in ASCII decimal. Must be less than 65536, and more than zero.
8. A single whitespace character (usually a newline).
9. A raster of Height rows, in order from top to bottom. Each row consists of Width gray values, in order from left to right. Each gray value is a number from 0 through Maxval, with 0 being black and Maxval being white. Each gray value is represented in pure binary by either 1 or 2 bytes. If the Maxval is less than 256, it is 1 byte. Otherwise, it is 2 bytes. The most significant byte is first.

A row of an image is horizontal. A column is vertical. The pixels in the image are square and contiguous.

ФОТО содержимого файла .room

В common общие элементы конфигурации.

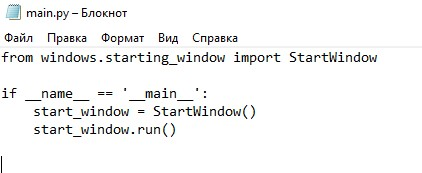
ФОТО!!!!

Наконец-то папка bases в нем хранятся базы данных

Вставить ФОТО бд диаграммы из dbeaver

Теперь как я показал, где хранятся изображения я расскажу об основных файлах.

Начнем с main и в нем ничего нет)



Но те три строчки что в нем содержатся запускают стартовое окно(меню).

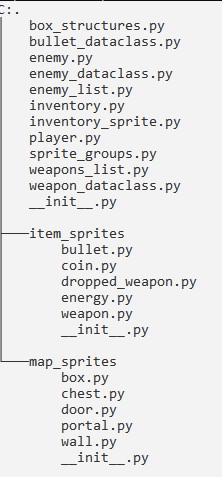
Два фундаментальных файлов, на которых держится вся игра это assets и camera.

Assets говорит сам за себя в нем подгружаются картинки, базы данных, музыка из папки assets.

Camera отрисовывает, декларирует какие спрайты используются, и отвечает за отображение на определенной координате.

Теперь я вкратце расскажу об остальных группах файлов.

* level.py создает сам уровень на котором играет юзер
* logger.py создает логи и сохранят их в папку logs
* в папке sprites вот такая структура



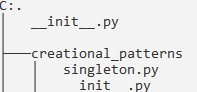
1. box\_structures – создает коробки по заданым в параемтрах шаблону
2. Серия датаклассов: bullet\_dataclass, enemy\_dataclass, weapon\_dataclass в них содержатся датаклассы этих сущностей
3. Player.py спрайт игрока: движение, стрельба и тд
4. Enemy.py спрайт врага: искусственный интеллект который заставляет его двигаться и нападать на главного игрока
5. В item\_sprites спрайты такие, как bullet (пуля, которая выстреливается из любого оружия). Coin, energy это монеты(пока нигде не используются) и мана(используется, для стрельбы из оружия)
6. В map\_sprites – сущности которые есть на карте коробкa(box.py), сундук(chest.py), портал(portal.py), стена(wall.py), дверь(door.py)
7. Enemy\_list, weapon\_list они содержат классы в которых хранятся массивы оружия, пуль, врагов. Также в нем коллекция из методов для работы с ними.
8. Inventory в нем хранится логика взаимодействия с инвентарем, inventory\_sprite класс спрайта (отрисовка и тд.)

* Папка windows содержит

1. choose\_player\_window.py (окно выбора игрока)
2. end\_screen.py (финальный экран со статистикой)
3. settings\_window.py (экран настроек)
4. starting\_window.py (меню для начала игры, в котором три кнопки (начать игру, настройка, выйти из игры)
5. \_\_init\_\_.py

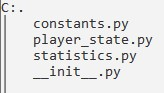
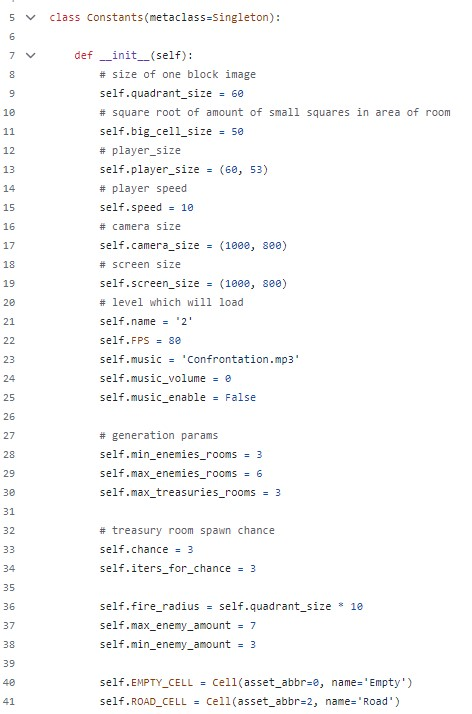
* В creational\_patterns лежит метакласс который реализует Singleton.

 In software engineering, the singleton pattern is a software design pattern that restricts the instantiation of a class to a singular instance, ensuring that only one object of that class exists in a system and providing easy access to that instance[4](https://en.wikipedia.org/wiki/Singleton_pattern). In genetics, a singleton refers to a person or an animal that is not a twin or part of a multiple birth[3](https://www.dictionary.com/browse/singleton). In general, a singleton can refer to a single item or a unique instance of a particular kind.



* В файле statistics.py считается статистика по игре. На файлах constnts.py и player\_state.py остановимся подольше.

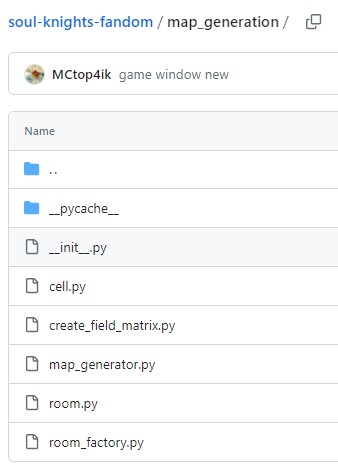
(1) statistics.py

Файловая структура папки settings.py

В constants хранятся основные настройки игры, сущности для пустоты и дороги, размер одного квадратика (сущность Cell) [quadrant\_size], размер игрока, скорость игрока, область видимости камеры и размер экрана, какой сейчас уровень, настройки вероятности генерирования уровня, музыка уровня, fps. В player\_state написана такая важная информация как, какие уровни осталось пройти, здоровье игрока, кол-во монет, маны.

* Map Generation



Эти 5 файлов – сердце игры, на них стоит ключевая роль создания карты.

Cell.py и Room.py два датакласса. Cell это минимальная структурная единица при генерации карты, Room это объект состоящий из Cell. Map\_generator.py (хранится класс MapGenerator) – самый первый файл в проекте, в нем первая часть алгоритма для создания карты. Он создает матрицу и начинает рандомно блуждать из центра матрицы, так создавая точки в которых будут комнаты. Он в конечном счете обрезает эту матрицу. И отправляет дальше на create\_field\_matrix (хранится класс CreateFieldMatrix) в этом файле эта матрица преобразовывается в огромное поле состоящее из Cell, это и станет нашей будущей картой. Строятся дороги и создаются важные комнаты. Room\_factory – фабрика которая создает объекты комнат и квадратиков, основываясь на данных из assets/rooms

Создание exe файла приложения

Creating an executable file (.exe) from a Python script using PyInstaller is a straightforward process that allows you to distribute your Python applications as standalone executables.



QR-code на Github с кодом игры

ПРИЛОЖЕНИЕ

Структура файлов и папок