Файл main.py:

1. Импорт модулей и библиотек: используется библиотека Pygame для графического интерфейса и обработки событий. Также импортируются различные вспомогательные модули, которые обеспечивают загрузку изображений, уровней и другие функции.
2. Инициализация Pygame и переменных: здесь происходит инициализация Pygame и задание основных переменных, таких как размер экрана, количество уровней, шрифт и другие.
3. Загрузка изображений: Изображения для различных элементов игры (например, стены, пустые клетки, финиш, монеты) загружаются и сохраняются в словаре.
4. Функции генерации уровня и загрузки уровня: Эти функции используются для создания игрового уровня на основе текстового файла и для загрузки нового уровня при необходимости.
5. Классы Tile и Player: Класс Tile представляет собой отдельный тайл (или клетку) на игровом поле, а класс Player представляет игрока. У игрока есть метод move, который позволяет ему двигаться по игровому полю.
6. Цикл игры: В этом цикле обрабатываются события Pygame, обновляется состояние игры и отрисовывается игровое поле. Если игрок достигает конца уровня или нажимает клавишу ESC, загружается новый уровень.
7. Анимация персонажа: В коде также присутствует функция для анимации персонажа, которая меняет изображение персонажа в зависимости от направления его движения.

Модуль utils:

1. Функция get\_number\_of\_levels() используется для подсчета количества уровней в игре. В директории ‘data/levels’ производится поиск файлов, начинающихся с ‘map’ и заканчивающихся на ‘.txt’. Количество таких файлов и возвращается функцией.
2. Функция load\_image(name, color\_key=None) загружает изображение из директории ‘data/images’. Если файл не найден, выводится сообщение об ошибке и происходит выход из программы. В случае успешной загрузки изображения, оно конвертируется в нужный формат в зависимости от параметра color\_key.
3. Функция load\_level(filename, pos\_x, pos\_y) используется для загрузки уровня из файла. Файл считывается, удаляются символы перевода строки, и подсчитывается максимальная длина строки. Каждая строка дополняется пустыми клетками (‘.’) до максимальной длины. Затем происходит сдвиг уровня относительно начальной позиции.
4. Функция terminate() используется для завершения работы программы.

Модуль scenes:

1. Функция end\_screen(screen, score, level, font\_name='freesansbold.ttf', font\_size=32) отображает экран окончания игры, на котором показываются очки и уровень игрока. Также создается кнопка для возврата в главное меню.
2. Функция level\_choice\_screen(screen) отображает экран выбора уровня. Игрок может выбрать уровень, используя клавиши вверх и вниз, и подтвердить выбор, нажав клавишу ввода.
3. Функция start\_screen(width, height, fps, screen, clock) отображает стартовый экран игры. На этом экране отображается фоновое изображение и текст приветствия. Игра начинается при нажатии любой кнопки.

Модуль labyrinth:

1. В основе модуля находится класс Labyrinth, который инициализируется с двумя параметрами: n и m, представляющими размеры лабиринта.
2. Метод start\_point\_generate используется для генерации начальной точки лабиринта, а метод finish\_point\_generate - для генерации конечной точки.
3. Метод transition\_choice выбирает следующую точку для перехода в процессе генерации лабиринта, учитывая текущую позицию и матрицу достижимости.
4. Метод create\_labyrinth генерирует сам лабиринт, создавая матрицу достижимости и матрицу переходов. В процессе генерации выбираются начальная и конечная точки, а затем происходит построение пути между ними.
5. Функция level\_view\_of\_the\_matrix преобразует сгенерированный лабиринт в игровой вид, заменяя ячейки на соответствующие символы. Затем эта функция сохраняет полученный лабиринт в файл map\_generated.txt в директории data/levels.

Папка data:

1. Поддиректория images используется для хранения всех изображений, которые используются в проекте. Это могут быть спрайты персонажей, фоны, текстуры и другие графические элементы. Все эти изображения загружаются в процессе выполнения программы и используются для визуализации игрового интерфейса и элементов игры.
2. Поддиректория levels служит для хранения файлов уровней игры. Каждый файл уровня содержит информацию о конкретном уровне игры, включая расположение стен, начальную и конечную точки, расположение монеток и другие элементы уровня. Эти файлы используются для генерации игровых уровней в процессе выполнения программы.