

Проект по теме Pygame

Nothing Special Tower Defense

Выполнил: Коканов Павел

Руководитель: Токаревская Светлана

Цель игры

Цель игры заключается в защите базы от волн врагов, идущих по определенному пути, с помощью постройки защитных башен.

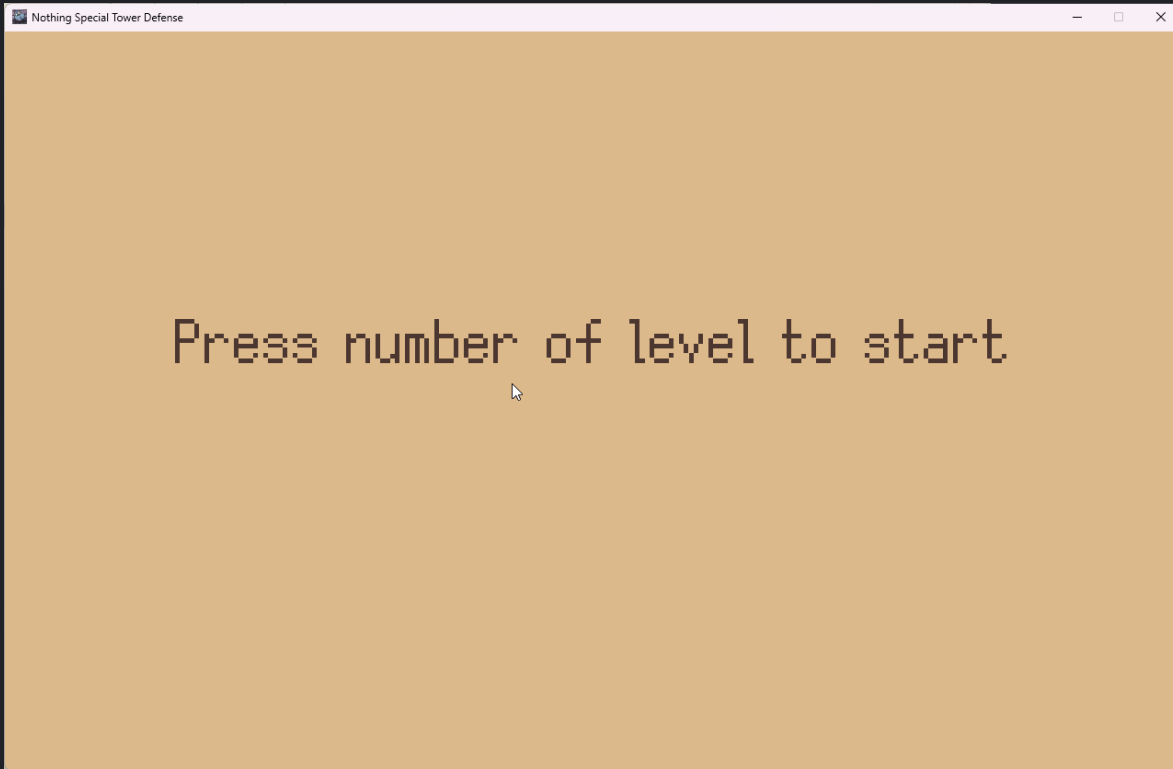
Задачи

- Изучение принципов работы игр на ругате
- Разработка логики работы игры
- Создание меню игры
- Подключение и создание различных уровней

Структура проекта

- `assets` - ассеты игры (аудио, шрифты, спрайты)
- `levels` - уровни для игры
- `button.py` - отвечает за кнопки в магазине
- `enemy.py` - класс врагов
- `inventory.py` - класс инвенторя (магазин)

- `level.py` - класс уровня (соединение уровня и врагов)
- `loader.py` - класс загрузки уровня и создания клетчатого поля
- `main.py` - запуск игры
- `settings.py` - настройки игры
- `tower.py` - класс башен

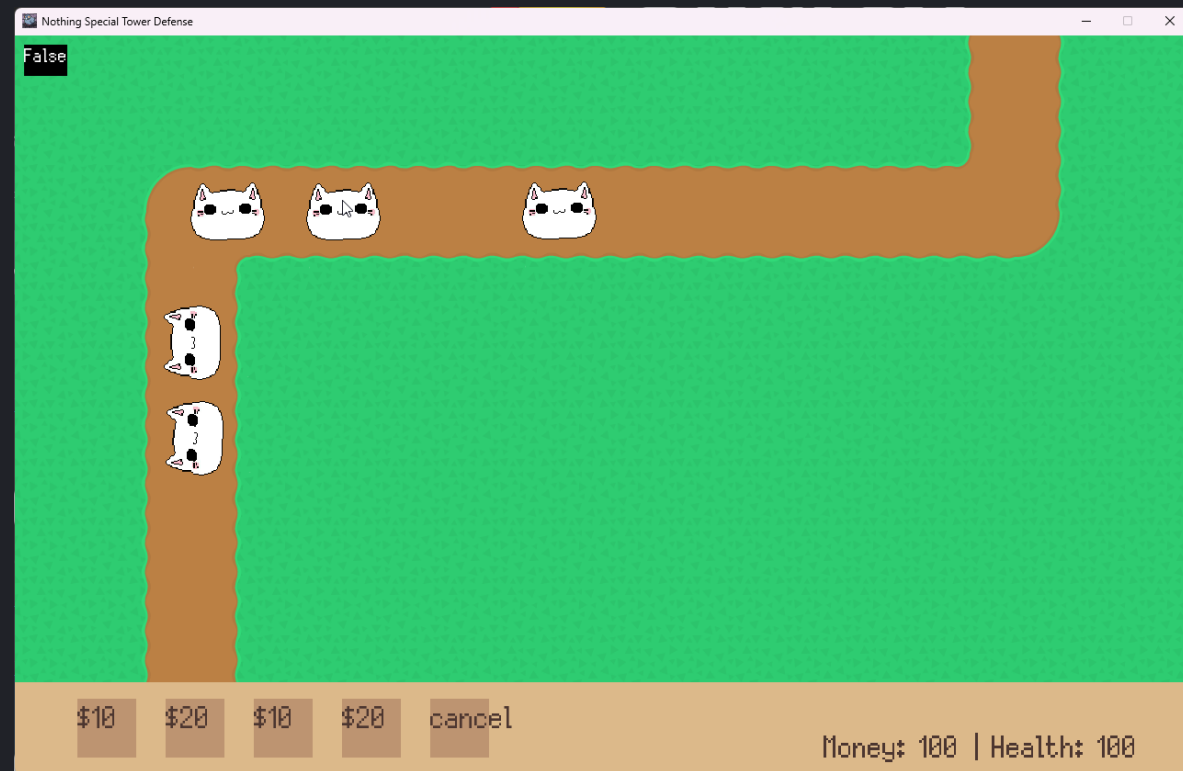


Меню

При нажатии цифры,
соответствующей
номеру уровня,
запускает уровень

Враги

Враги идут по
заданному пути



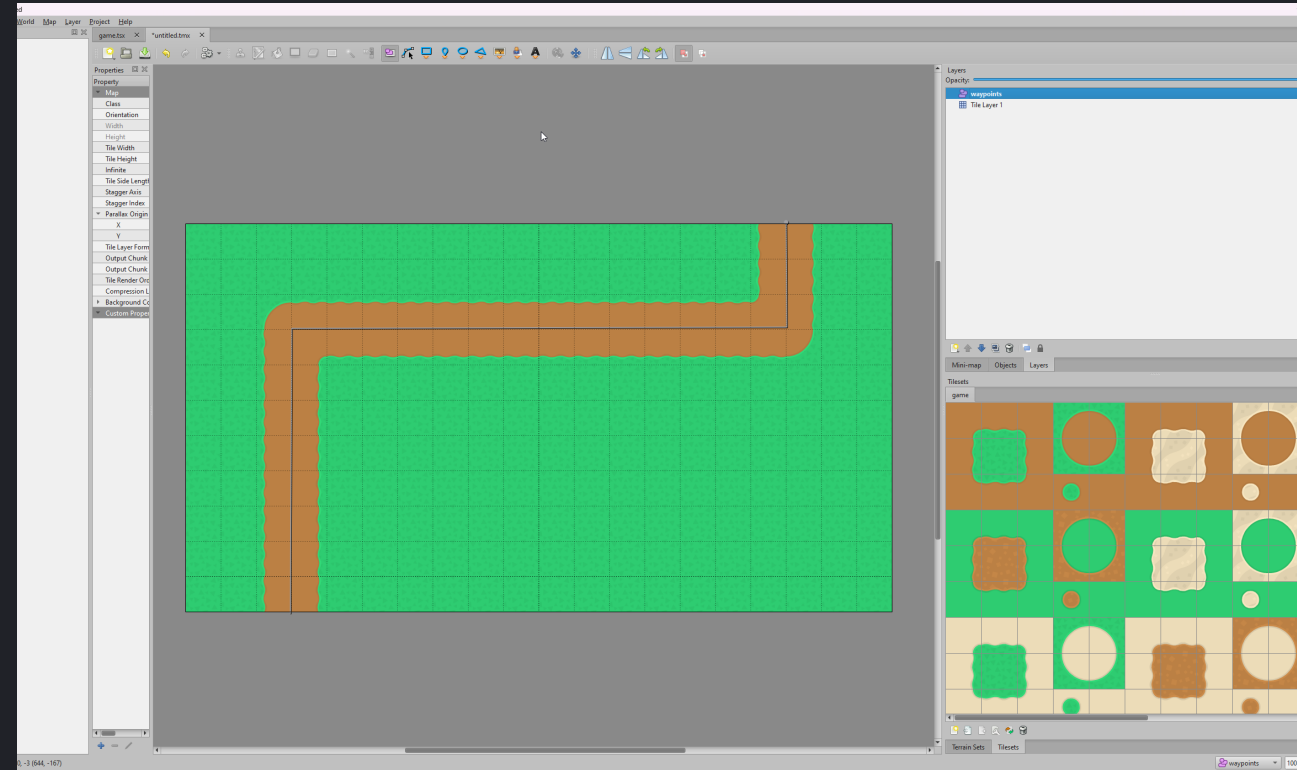


Башни

При нажатии на ячейку
магазина снизу
появляется
ВОЗМОЖНОСТЬ
расставлять башни

Структура уровней

Уровни созданы с помощью приложения Tiled. Из него можно экспортировать карту в формате png, сетку карты и список waypoint в формате json



Структуры классов

Класс `Game` (`main.py`)

- В `init` создается окно `pygame`
- В `run` запускается цикл игры с меню выбора уровня, инициализируются объекты классов уровня, сетки уровня и инвенторя

Класс Level (level.py)

- В **init** создается группа спрайтов врагов, запускается функция **setup**
- В **setup** загружается карта уровня, запускаются функции класса Board для создания сетки уровня и получения waupoint-ов
- В **draw** рисуется карта уровня
- В **run** вызывает draw, рисует и обновляет спрайты врагов

Класс Board (loader.py)

- В **init** создается двумерная матрица
- В **load** загружается json уровня, получение списка материалов карты и waypoint-ов
- В **fill_board** двумерная матрица заполняется списками из 0 и 1 (1 - можно поставить башню)
- В **check** ячейка матрицы проверяется на свободу

Класс Inventory (inventory.py)

- В **init** загружаются объекты для магазина, создаются группа спрайтов башен
- В **setup** создается список объектов класса Button
- В **run** рисуются кнопки, спрайты башен и текст инвенторя, происходит проверка нажатия на кнопку

Класс Button (button.py)

- В **init** создается `surface` и `rectangle` для текста кнопки
- В **input** происходит проверка нажатия на кнопку

Класс Enemy (enemy.py)

- В **init** создается список спрайтов, задается следующий waypoint
- В **move** происходит перемещение к следующему waypoint
- В **rotate** происходит вращение спрайта врага

Класс Tower (tower.py)

- В **init** создается список спрайтов и cooldown анимации, задается позиция центра клетки сетки
- В **play_animation** происходит смена спрайта анимации на следующий

Используемые библиотеки

- pygame
- sys
- math
- json
- random

Реализуемые технологии

Игра на ругате с использованием ООП и
подгрузкой уровней из json файлов

Спасибо за внимание!

Презентация сделана с помощью [Marp](#)