

Игра «Курица-воин» В жанре Roguelike

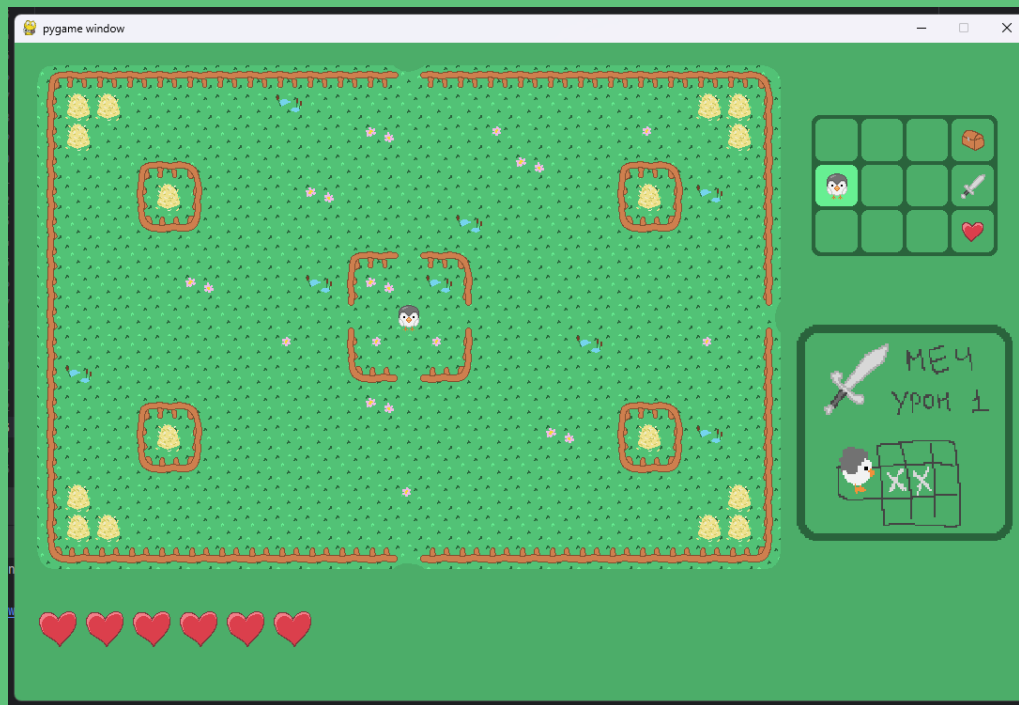


Кан Артур

Roguelike

Игра обладает основными свойствами данного жанра:

- Случайная генерация уровня
- Пошаговость
- Необратимая смерть персонажа



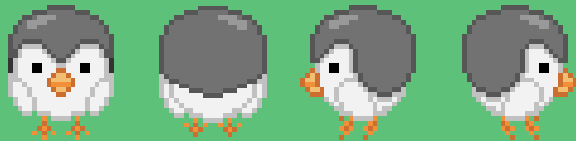
Игрок

Управлять можно как на стрелочки, так и на wasd

Изначально у игрока 6 очков здоровья -> 

Чтобы атаковать, надо зажать пробел

Изначально у игрока есть меч с 1 уроном



Класс игрока инициализирует
почти всю информацию о нём

Оружие

У оружия есть свой собственный класс

В игре присутствует 5 видов оружия

**Информация о каждом виде содержится в файле
weapoons_data.json**

**Класс выводит игроку
Информацию о текущем оружии**



```
{  
  "sword": {  
    "atk": 1,  
    "x_area": [[1, 0], [2, 0]],  
    "y_area": [[0, 1], [0, 2]],  
    "-x_area": [[-2, 0], [-1, 0]],  
    "-y_area": [[0, -2], [0, -1]]  
  },  
}
```

**В файле хранится урон и территория,
по которой наносится урон в
Зависимости от направления игрока**

Враг

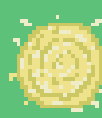
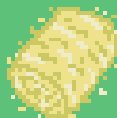
Класс врага – основной класс не только для врагов, но и некоторых препятствий

Каждый враг имеет свой класс, который является наследником класса врага, и имеет некоторые методы, уникальные для конкретного врага

```
class HayRoll(Enemy):  
    └ Kan Artur  
    def __init__(self, position):  
        super().__init__(position)  
        self.change_pic("hay_roll/hay_roll_1.png")  
        self.picID = "1"  
        self.hp = 3  
        self.direction = [-1, -1]  
  
2 usages (2 dynamic) └ Kan Artur  
    def set_direction(self, direction):  
        self.direction = direction
```

Для конкретно этого врага важно направление, поэтому реализован метод установки направления

Как видно, он также инициализирует здоровье и картинку



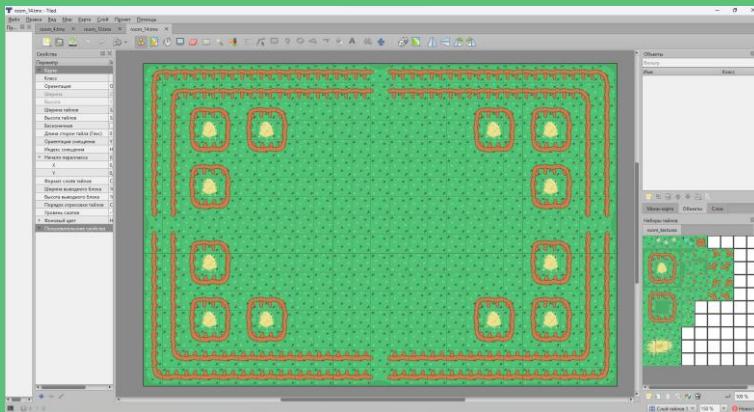
Комната

Класс комнаты содержит в себе информацию о, собственно, комнатах

То, какие враги где должны заспавниться в комнате, записано в файле `enemies_data.json`

```
{  
  "1": [{"crow", [12, 8]]},  
  "2": [{"crow", [7, 3]}, {"crow", [17, 13]}],  
  "3": [{"crow", [16, 8]}],  
  "4": [{"hay_roll", [12, 8]}],  
  "5": [{"hay_roll", [15, 5]}, {"hay_roll", [9, 11]}],  
  "6": [{"crow", [7, 1]}, {"crow", [17, 15]}],  
  "7": [{"hay_roll", [7, 3]}, {"crow", [12, 8]}],  
}
```

**Все комнаты были созданы с помощью Tiled,
Реализованы с помощью pytmx**



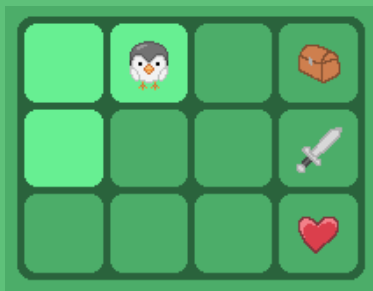
Также в классе комнаты прописано движение врагов

```
def boss_roll_path(self, start, direction):  
    new_direction = direction  
    next_x = start[0] + new_direction  
    if not self.is_free((next_x, start[1])):  
        new_direction = -new_direction  
        next_x = start[0] + new_direction  
    return next_x, start[1], new_direction
```

Карта

Класс карты комнаты создаёт весь уровень игры, который представляет собой матрицу 3 на 4

Именно класс карты генерирует рандомные комнаты



Класс также отображает игроку карту уровня, на которой видно местоположение, зачищенные комнаты, специальные комнаты



Класс карты также блокирует выходы из комнаты, ведущие за край уровня

```
self.id_map = [[f"room_{random.randint(a: 1, RR_AM)}", f"room_{random.randint(a: 1, RR_AM)}",  
                f"room_{random.randint(a: 1, RR_AM)}", f"treasure_room_1"],  
               ["start_room", f"room_{random.randint(a: 1, RR_AM)}",  
                f"room_{random.randint(a: 1, RR_AM)}", "room_boss"],  
               [f"room_{random.randint(a: 1, RR_AM)}", f"room_{random.randint(a: 1, RR_AM)}",  
                f"room_{random.randint(a: 1, RR_AM)}", f"treasure_room_2"]]
```

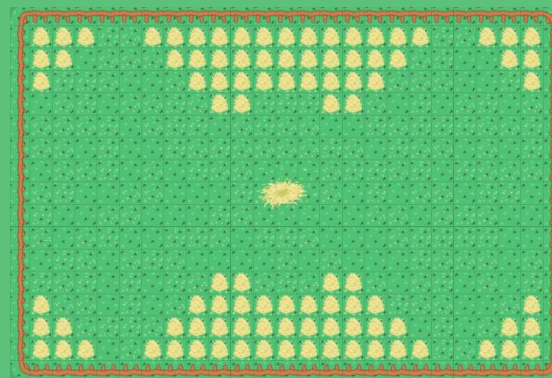
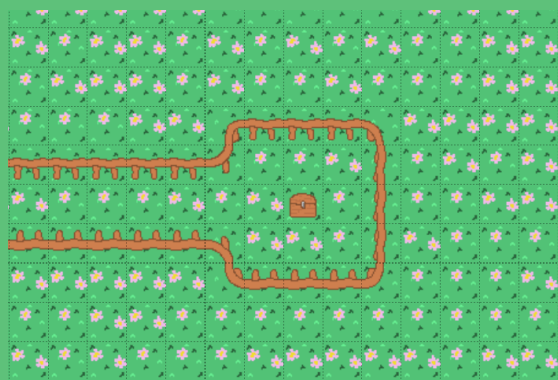
Карта

В игре есть три специальные комнаты:

Сокровищница с оружием (открывается после зачистки четырех комнат) – игрок получает новое случайное оружие

Сокровищница со здоровьем (после шести комнат) – игрок восполняет своё здоровье до случайного значения от 6 до 9

Комната босса (после зачистки всего уровня) – комната с самим боссом, из которой нельзя выйти



Частицы

Класс частиц создан для украшения игры

Частицы появляется каждый раз, когда враг убит, игрок получает урон и когда игрок убит

Все частицы: 



Игрок получил урон

Игра

Класс игры связывает абсолютно все вышеупомянутые классы в единое целое

Класс игры совершает очень много проверок: получение урона игроком или врагом, кол-во врагов в комнате, перемещение по комнатам, передвижение игрока и врагов

В классе игры реализован также счёт

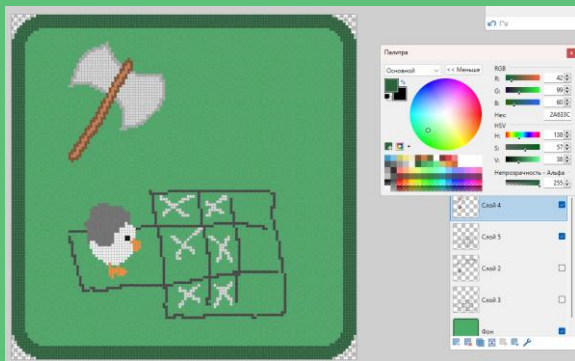
Это всё – один метод:

```
def check_room(self):
    if not self.boss_music_start and self.map.curr_yx == (1, 3):
        self.boss_music_start = True
        pygame.mixer.music.load("data/sound/Fever-Pitch.mp3")
        pygame.mixer.music.set_volume(0.65)
        pygame.mixer.music.play()

    if not self.map.curr_room.clear:
        if len(self.map.curr_room.enemies) == 0 and "treasure" not in self.map.curr_room.roomID:
            self.map.curr_room.clear = True
            if "boss" not in self.map.curr_room.roomID:
                with open("data/maps/enemies_data.json") as f:
                    enemies_data = json.load(f)
                    self.score += len(enemies_data[self.map.curr_room.roomID[5:-4]]) * 1000
            for enemy in self.map.curr_room.enemies:
                if enemy.hp <= 0:
                    if enemy.__class__.__name__ == "Boss":
                        self.map.curr_room.enemies = []
                        hay_things_death.set_volume(2)
                        hay_things_death.play()
                        particles_pos = (enemy.get_position()[0][0] * TILE_SIZE + TILE_SIZE,
                                        enemy.get_position()[0][1] * TILE_SIZE + TILE_SIZE)
                        particles_rect = pygame.Rect(enemy.get_position()[0][0] * TILE_SIZE,
                                                    enemy.get_position()[0][1] * TILE_SIZE,
                                                    2 * TILE_SIZE, 3 * TILE_SIZE)
```

```
        self.create_particles(particles_pos, particle="hay", particles_rect, count=30)
        pygame.mixer.music.load("data/sound/boss_defeated.mp3")
        pygame.mixer.music.set_volume(0.65)
        pygame.mixer.music.play()
        self.score += 1000 * player.hp
        self.boss_defeated = True
    else:
        particles_pos = (enemy.get_position()[0] * TILE_SIZE + TILE_SIZE,
                        enemy.get_position()[1] * TILE_SIZE + TILE_SIZE)
        particles_rect = pygame.Rect(enemy.get_position()[0] * TILE_SIZE,
                                    enemy.get_position()[1] * TILE_SIZE,
                                    2 * TILE_SIZE, 2 * TILE_SIZE)
        if enemy.__class__.__name__ == "Crow":
            crow_death.play()
            self.create_particles(particles_pos, particle="crow", particles_rect, count=10)
        else:
            hay_things_death.play()
            self.create_particles(particles_pos, particle="hay", particles_rect, count=10)
        self.map.curr_room.enemies.remove(enemy)
```

Изображение



Всё, что вы увидите в этой игре отрисовано с нуля с любовью :)

Почти в каждом классе есть метод `render()`, который отвечает за выведение чего-либо на экран

```
def render(self, screen):  
    screen.blit(self.info_im, (833, 300))
```

```
def render(self, screen):  
    self.curr_room.render(screen)  
    self.block_edges()  
    self.treasure_taken()  
    map_net = pygame.image.load("data/images/screen/map_net.png")  
    clearCheck = {False: (70, 173, 105),  
                  True: (103, 239, 146)}  
    for y in range(3):  
        for x in range(4):  
            rect = pygame.Rect(x * self.map_tile_size + 852,  
                               y * self.map_tile_size + 80,  
                               self.map_tile_size, self.map_tile_size)  
            screen.fill(clearCheck[self.map[y][x].clear], rect)  
    screen.blit(map_net, (849, 77))  
    icon = pygame.image.load(f"{PLAYER_DIR}/player_front.png")  
    delta = (icon.get_width() - self.map_tile_size) // 2  
    screen.blit(icon, ((self.cr_yx[1] * self.map_tile_size - delta + 852,  
                        self.cr_yx[0] * self.map_tile_size - delta + 80)))
```

Размеры методов `render()`
Сильно варьируется

Звук и музыка



Pizza Tower – Mort's farm



**Undertale Yellow – Fever Pitch,
Main Menu 3 Snowdin**



Hollow Knight - Dirtmouth

**В игре очень много звуков
для различных событий
и действий
Например, урон по игроку**

Прочее

Тutorials, Экран проигрыша



! ЕСЛИ ВЫ ВЫШЛИ ИЗ КОМНАТЫ, НЕ ЗАЧИСТИВ ЕЁ ДО КОНЦА, ТО ВСЕ ВРАГИ ВОССТАНОВЯТСЯ!

⚔️ - СКОРОВИЩНИЦА С ОРУЖИЕМ

♥️ - СКОРОВИЩНИЦА СО ЗДОРОВЬЕМ

🗡️ - КОМНАТА С БОССОМ

ВОЗМОЖНО, ЭТИ КОМНАТЫ ОТКРОУТСЯ ЧУТЬ-ЧУТЬ ПОЗЖЕ...



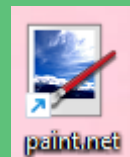
Счёт считается следующим образом

После зачистки комнаты прибавляется кол-во врагов, которые были в этой комнате, умноженное на 1000

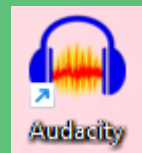
Если игрок получает урон, то из счёта отнимается 500 очков

После победы над боссом к счёту прибавляется оставшееся кол-во очков здоровья, умноженное на 1000

Другие использованные программы:



Рисование абсолютно всех изображений и спрайтов



Обработка звуковых эффектов

Игра «Курица-воин» В жанре Roguelike



Кан Артур