

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего профессионального образования
«Южно-Уральский государственный университет (национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

Разработка редактора локаций для компьютерной 2D-игры в жанре Survival RPG на платформе Godot Engine

Научный руководитель:
ст. преподаватель кафедры СП
Я.А. Краева

Автор:
Студент группы КЭ-401
К.И. Слитиков

Челябинск, 2024 г

АКТУАЛЬНОСТЬ

- Роль игровой индустрии в мире
- Развитие инструментов для разработки игр
- Значимость игровых редакторов и основные аспекты их актуальности

ЦЕЛЬ И ЗАДАЧИ

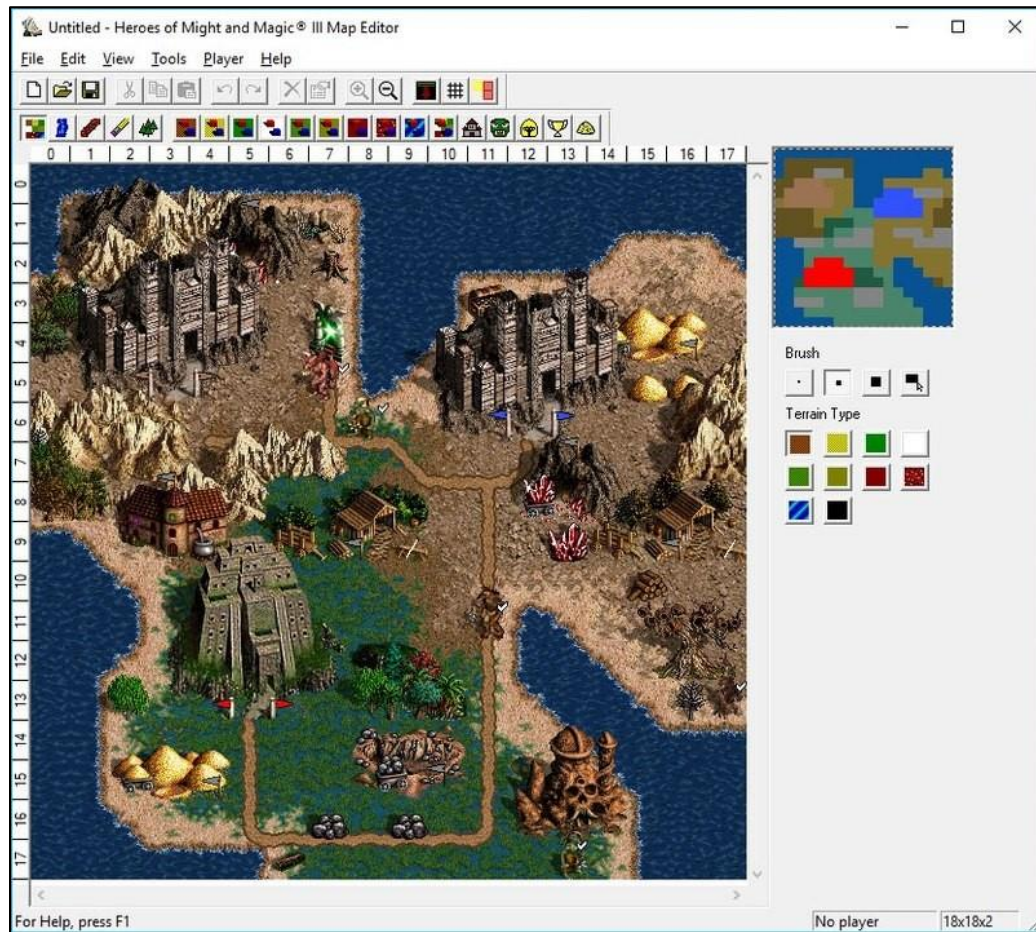
Цель:

Разработка редактора локаций для компьютерной 2D-игры в жанре Survival RPG на платформе Godot Engine

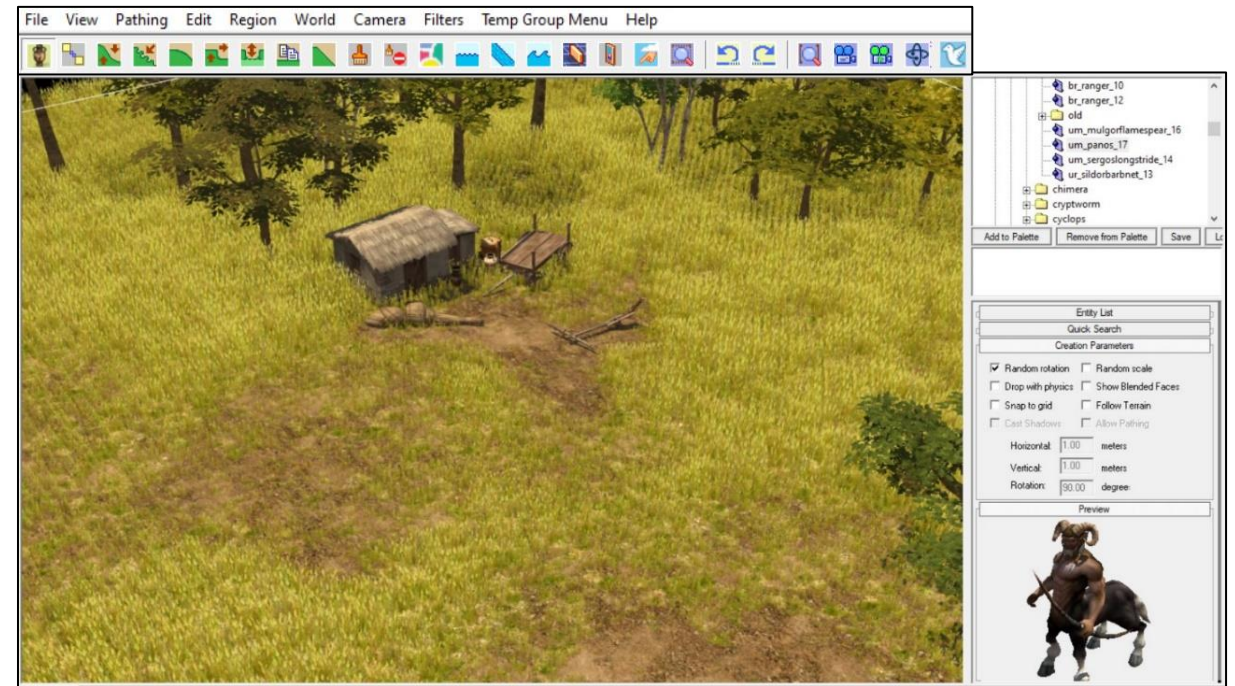
Задачи:

1. Провести анализ предметной области и обзор аналогов
2. Спроектировать приложение
3. Реализовать и протестировать приложение

ОБЗОР АНАЛОГОВ



Heroes Of Might And Magic 3



Titan Quest

ЦЕЛЬ И ВОЗМОЖНОСТИ РЕДАКТОРА

Цель:

Возможность создания игровых локаций

Возможности редактора:

- Создание 7 типов локаций с возможностью указать размеры
- Выбор из 3 типов объектов для размещения: декоративные, обычные, персонажи
- Сохранение локации
- Загрузка локации
- Тестирование локации

МОДУЛИ ДЛЯ РАЗРАБОТКИ

Разрабатываемые модули:


- Модуль отрисовки локации
- Модуль ограничений
- Модуль перемещения по редактору
- Модули персонажей
- Модули объектов
- Модули панелей управления
- Модули загрузки и сохранения

МОДУЛЬ ПЕРСОНАЖА ИГРОКА

```
func _physics_process(delta):
    var velocity = Vector2.ZERO
    if Input.is_action_pressed("right_walk"):
        velocity.x += 1.0
    if Input.is_action_pressed("left_walk"):
        velocity.x -= 1.0
    if Input.is_action_pressed("down_walk"):
        velocity.y += 1.0
    if Input.is_action_pressed("up_walk"):
        velocity.y -= 1.0
    velocity = velocity.normalized()

    if velocity == Vector2.ZERO:
        $AnimationTree.get("parameters/playback").travel("Idle")
    else:
        $AnimationTree.get("parameters/playback").travel("Walk")
        $AnimationTree.set("parameters/Idle/blend_position", velocity)
        $AnimationTree.set("parameters/Walk/blend_position", velocity)
        move_and_slide(velocity * speed)

    position.x = clamp(position.x, 10, $Camera2D.limit_right-10)
    position.y = clamp(position.y, 10, $Camera2D.limit_bottom-10)
```



Скрипт «Player.gd»

МОДУЛЬ ПЕРСОНАЖА ИГРОКА

```
func add_item(item_name, item_quantity):
    for item in inventory:
        if inventory[item][0] == item_name:
            var stack_size = int(Jsondata.item_data[item_name]["StackSize"])
            var able_to_add = stack_size - inventory[item][1]
            if able_to_add >= item_quantity:
                inventory[item][1] += item_quantity
                return
            else:
                inventory[item][1] += able_to_add
                item_quantity = item_quantity - able_to_add
        # item doesn't exist in inventory yet, so add it to an empty slot
    for i in range(NUM_INVENTORY_SLOTS):
        if inventory.has(i) == false:
            inventory[i] = [item_name, item_quantity]
    return
```

Реализация функции проверки и добавления предметов
в слот инвентаря скрипта «PlayerInventory.gd»

МОДУЛЬ ПЕРСОНАЖА ИГРОКА

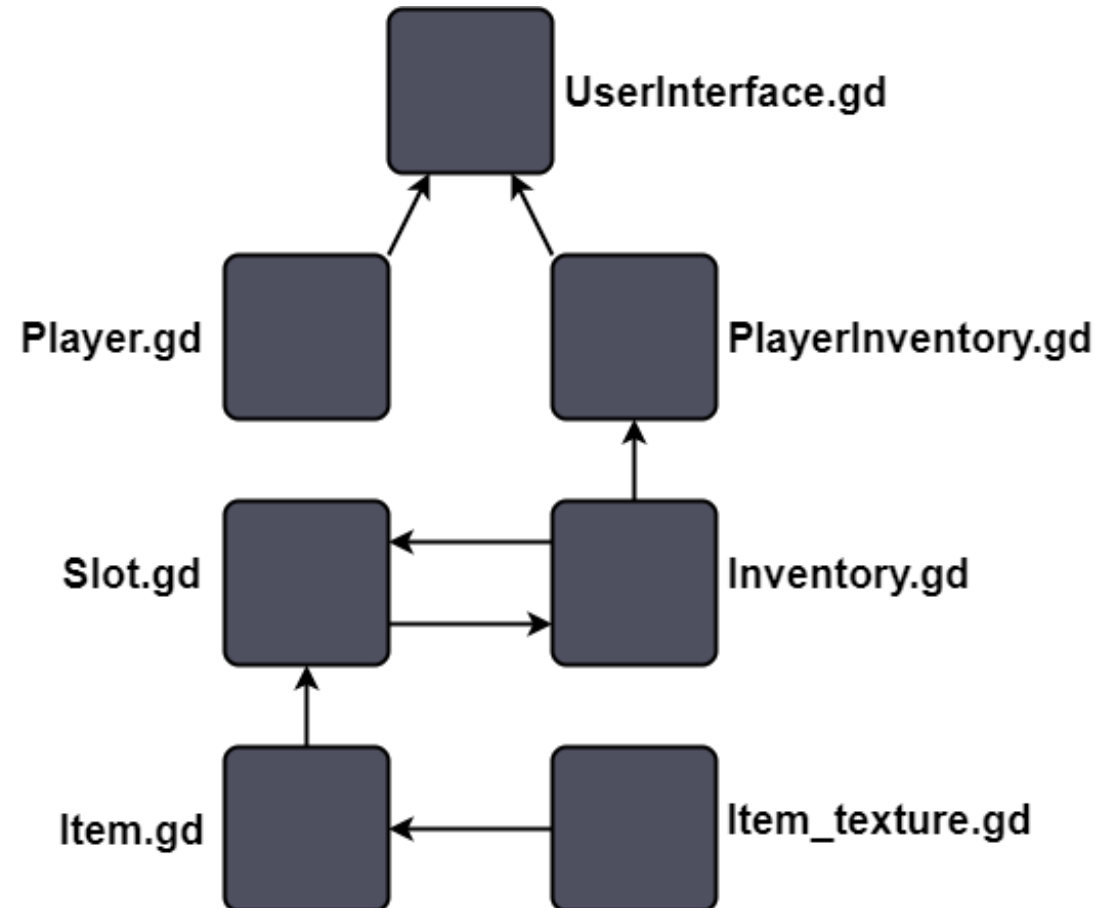


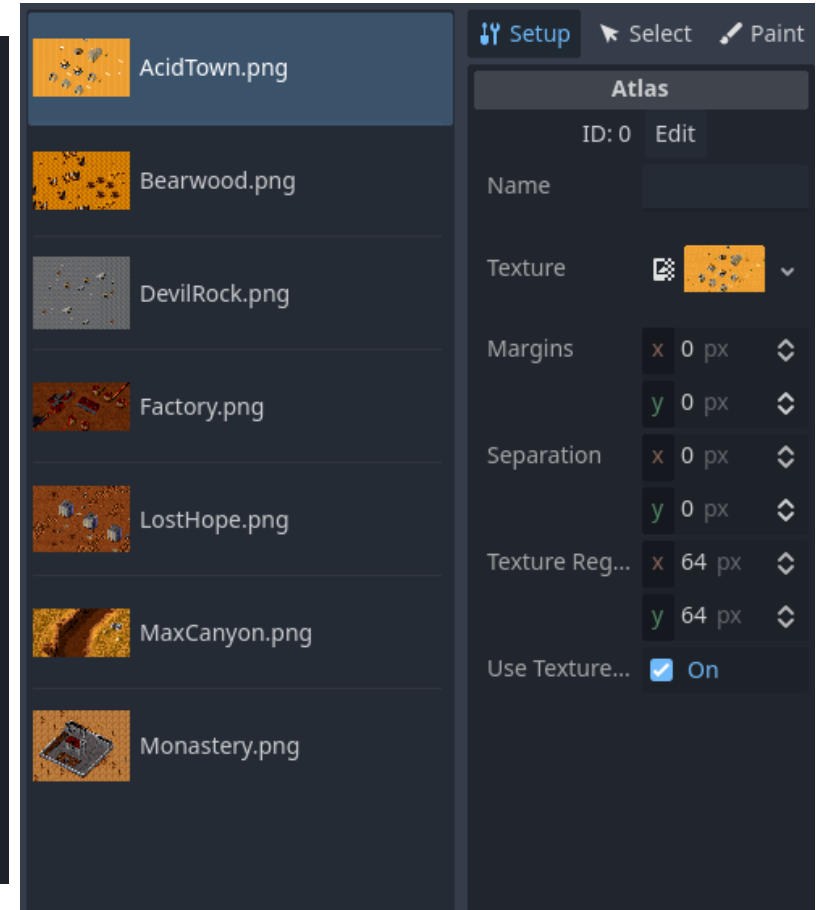
Схема взаимодействия модулей реализации персонажа

МОДУЛЬ ОТРИСОВКИ ЛОКАЦИЙ

```
func _on_create_button_pressed():
>| print("you pressed")
>| var lineWidth = width_input.text
>| var lineHeight = height_input.text
>| var lineId = tile_id_input.text
>|
>| # Cheking not null's
>| MAP_WIDTH = int(lineWidth) if lineWidth != "" and int(lineWidth) <= 30 else 10
>| MAP_HEIGHT = int(lineHeight) if lineHeight != "" and int(lineHeight) <= 30 else 10
>| BASE_TILE_ID = int(lineId) if lineId != "" and int(lineId) <= 6 else 0
>| # Clear the map before creating a new one
>| tile_map.clear()
>| create_map()

func create_map():
>| var layer = 0 # Default layer, you can change this if you have multiple layers
>| var source_id = BASE_TILE_ID # Use the same base tile for all cells
>| var atlas_coords = Vector2i(0, 0) # Use the same atlas coordinates for all cells
>| var alternative_tile = 0 # Default alternative tile
>| for x in range(MAP_WIDTH):
>| >| for y in range(MAP_HEIGHT):
>| >| >| set_cell(layer, Vector2i(x, y), source_id, atlas_coords, alternative_tile)
>| >| >| print("Set tile at position: ", Vector2i(x, y), " with source ID: ", source_id)
```

Фрагмент скрипта «TileMap.gd»



Окно настройки «TileMap»

МОДУЛЬ ОГРАНИЧЕНИЙ

```
func place_object(position: Vector2):
>| print("obj placed")
>| var new_object = load(selected_scene_path).instantiate()
>| add_child(new_object)
>| new_object.call("load_texture", selected_texture_path)
>| new_object.position = position + Vector2(GRID_SIZE / 2, GRID_SIZE / 2)
>| new_object.add_to_group(PLACED_OBJECT_GROUP) # Add to the group
>| occupied_cells[snap_to_grid(position)] = new_object # Mark the cell as occupied
>|
func is_within_boundaries(position: Vector2) -> bool:
>| var cell_x = position.x / GRID_SIZE
>| var cell_y = position.y / GRID_SIZE
>| return cell_x >= 0 and cell_x < GRID_WIDTH and cell_y >= 0 and cell_y < GRID_HEIGHT
>|
func is_cell_occupied(position: Vector2) -> bool:
>| return occupied_cells.has(position)
>|
func _on_create_button_pressed():
>| var lineWidth = width_input.text
>| var lineHeight = height_input.text
>| delete_all_objects()
>| GRID_WIDTH = int(lineWidth) if lineWidth != "" and int(lineWidth) <= 30 else 10
>| GRID_HEIGHT = int(lineHeight) if lineHeight != "" and int(lineHeight) <= 30 else 10
```

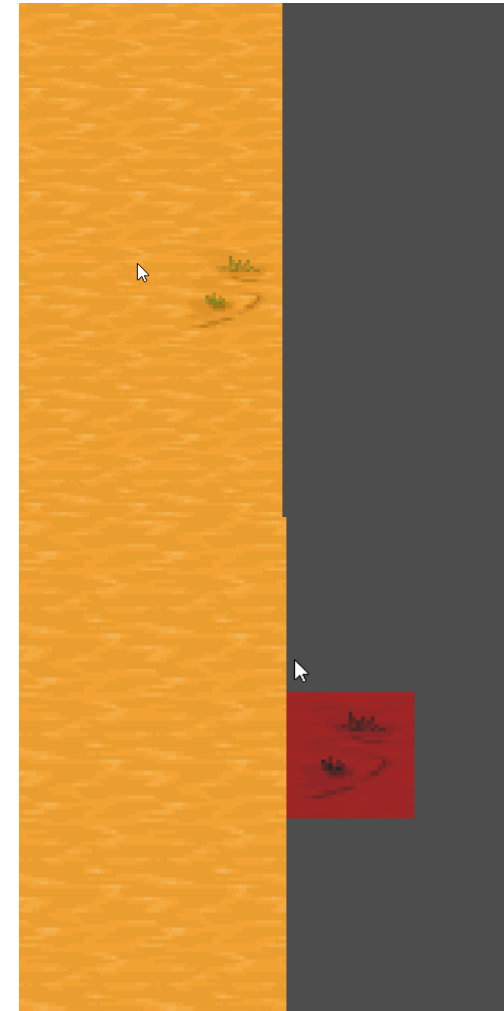
Фрагмент скрипта «main_scene.gd»

МОДУЛЬ ОГРАНИЧЕНИЙ

```
▼ func create_preview_object():
  >| print("preview check")
▼ >| if preview_object:
  >| >| print("queue freed")
  >| >| preview_object.queue_free()
▼ >| if selected_scene_path:
  >| >| preview_object = load(selected_scene_path).instantiate()
  >| >| add_child(preview_object)
  >| >| preview_object.modulate = Color(1, 1, 1, 0.5) # Make the preview object semi-transparent

▼ func _process(delta):
  >| update_preview_position()
  >|
▼ func update_preview_position():
▼ >| if preview_object and can_place:
  >| >| var grid_position = snap_to_grid(get_global_mouse_position())
  >| >|
▼ >| >| if is_within_boundaries(grid_position):
  >| >| >| preview_object.modulate = Color(1, 1, 1, 0.5)
▼ >| >| else:
  >| >| >| preview_object.modulate = Color(1, 0, 0, 0.5)
  >| >| >| preview_object.position = grid_position + Vector2(GRID_SIZE / 2, GRID_SIZE / 2)
```

Фрагмент скрипта «main_scene.gd»



Демонстрация превью объекта

ПЕРЕМЕЩЕНИЕ ПО РЕДАКТОРУ

```
func _input(event):
>|  if event is InputEventMouseButton:
>|  >|  if event.is_pressed():
>|  >|  >|  if event.button_index == MOUSE_BUTTON_WHEEL_DOWN:
>|  >|  >|  >|  if zoom > zoom_min:
>|  >|  >|  >|  >|  zoom -= zoom_speed
>|  >|  >|  >|  >|  if is_equal_approx(zoom.x, 0.0):
>|  >|  >|  >|  >|  >|  zoom.x = zoom_min.x
>|  >|  >|  >|  >|  if is_equal_approx(zoom.y, 0.0):
>|  >|  >|  >|  >|  >|  zoom.y = zoom_min.y
>|  >|  >|  if event.button_index == MOUSE_BUTTON_WHEEL_UP:
>|  >|  >|  >|  if zoom < zoom_max:
>|  >|  >|  >|  >|  zoom += zoom_speed
>|  >|  >|  >|  >|  if is_equal_approx(zoom.x, 0.0):
>|  >|  >|  >|  >|  >|  zoom.x = zoom_max.x
>|  >|  >|  >|  >|  if is_equal_approx(zoom.y, 0.0):
>|  >|  >|  >|  >|  >|  zoom.y = zoom_max.y
>|  if event is InputEventMouseMotion:
>|  >|  if event.button_mask == MOUSE_BUTTON_MASK_MIDDLE:
>|  >|  >|  position -= event.relative / zoom
>|  pass
```

Фрагмент скрипта камеры редактора

МОДУЛЬНОЕ ТЕСТИРОВАНИЕ

Модули	Тесты
Модули персонажа	Управление персонажем и проверка инвентаря
Модуль отрисовки локаций, Модуль перемещения по редактору	Проверка настройки отрисовки с разными параметрами и перемещение
Модуль ограничений	Проверка корректности отображения ограничений после отрисовки
Модуль панелей управления, Модуль объектов	Проверка работоспособности панелей управления и корректности размещения объектов на локации

ЮЗАБИЛИТИ-ТЕСТИРОВАНИЕ

В тестировании участвовало 5 человек

Обнаруженные проблемы:

- Некорректная загрузка состояния карты
- При наложении панелей управления на карту, при перемещении камеры, возможность расставлять объекты не блокируется.
- Некорректное перемещение по локациям внутри игры.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ

- Выполнен анализ предметной области и обзор аналогов
- Спроектировано 7 модулей для редактора локаций
- Модули были реализованы и протестированы

Направление дальнейших исследований:

- Доработка существующих модулей
- Упрощение импорта созданных локаций
- Корректирование объектов и перемещения на локациях