Лабораторная работа №33-35

Тема: разработка механики 2D рогалика

Цель: приобрести навыки в разработке механики 2D рогалика

Ход работы

1. Выполнение работы

Создание игрового проекта 2D рогалик

1. Создал анимации



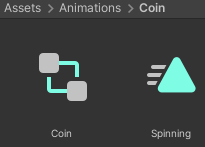


Рис. 33.2 Анимации

1. Создал префабы



Рис. 33.2 Префабы

1. Создал 2 сцены

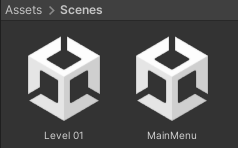


Рис. 33.2 Сцены

1. Сцена MainMenu

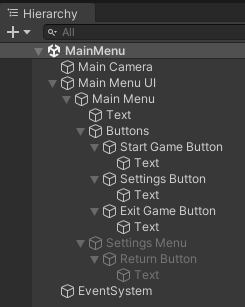


Рис. 33.2 Иерархия сцены MainMenu

1. Сцена Level 01

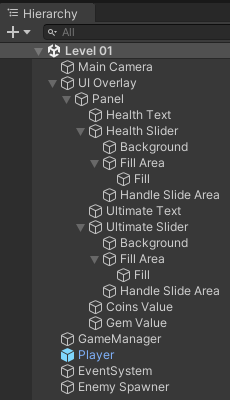


Рис. 33.2 Иерархия сцены Level 01

1. Создал скрипт MainMenu, который отвечает за действия при нажатии кнопок в главном меню.

Листинг MainMenu.cs

using UnityEngine;

using UnityEngine.SceneManagement;

public class MainMenu : MonoBehaviour

{

public GameObject settingsPanel;

public void StartGame()

{

SceneManager.LoadScene(1);

}

public void OpenSettings()

{

settingsPanel.SetActive(true);

}

public void CloseSettings()

{

settingsPanel.SetActive(false);

}

public void ExitGame()

{

Application.Quit();

}

}

1. Создал скрипт CameraFollowPlayer, перемещающий камеру за игроком.

Листинг CameraFollowPlayer.cs

using UnityEngine;

public class CameraFollowPlayer : MonoBehaviour

{

public Transform player;

public float smoothing;

public Vector3 offset;

void FixedUpdate()

{

if (player != null)

{

Vector3 newPosition = Vector3.Lerp(transform.position, player.transform.position + offset, smoothing);

transform.position = newPosition;

}

}

}

1. Создал скрипт CurrencyPickup, отвечающий за подбор монет и самоцветов игроком.

Листинг CurrencyPickup.cs

using UnityEngine;

public class CurrencyPickup : MonoBehaviour

{

public enum PickupObject { COIN, GEM }

public PickupObject currentObject;

public int pickupQuantity;

private void OnTriggerEnter2D(Collider2D collision)

{

if (collision.name == "Player")

{

PlayerStats.playerStats.AddCurrency(this);

Destroy(gameObject);

}

}

}

1. Создал скрипт Enemy, отвечающий за поведение врага при лечении, получении урона и смерти.

Листинг Enemy.cs

using UnityEngine;

using UnityEngine.UI;

public class Enemy : MonoBehaviour

{

public float health;

public float maxHealth;

public GameObject coin;

public GameObject healthBar;

public Slider healthBarSlider;

private void Start()

{

health = maxHealth;

}

public void DealDamage(float damage)

{

healthBar.SetActive(true);

health -= damage;

CheckDeath();

healthBarSlider.value = CalculateHealthPercentage();

}

public void HealCharacter(float heal)

{

health += heal;

CheckOverheal();

healthBarSlider.value = CalculateHealthPercentage();

}

public void CheckOverheal()

{

if (health > maxHealth)

health = maxHealth;

}

private void CheckDeath()

{

if (health <= 0)

{

Instantiate(coin, transform.position, Quaternion.identity);

Destroy(gameObject);

}

}

private float CalculateHealthPercentage()

{

return health / maxHealth;

}

}

1. Создал скрипт EnemyAttack, ищущий игрока по компоненту.

Листинг EnemyAttack.cs

using UnityEngine;

public class EnemyAttack : MonoBehaviour

{

protected GameObject player;

public virtual void Start()

{

player = FindObjectOfType<PlayerMovement>().gameObject;

}

}

1. Создал скрипт TestEnemyShooting, отвечающий за стрельбу врага по игроку.

Листинг TestEnemyShooting.cs

using System.Collections;

using UnityEngine;

public class TestEnemyShooting : EnemyAttack

{

public GameObject projectile;

public float minDamage;

public float maxDamage;

public float projectileForce;

public float cooldown;

public override void Start()

{

base.Start();

StartCoroutine(ShootPlayer());

}

IEnumerator ShootPlayer()

{

yield return new WaitForSeconds(cooldown);

if (player != null)

{

GameObject spell = Instantiate(projectile, transform.position, Quaternion.identity);

Vector2 myPos = transform.position;

Vector2 targetPos = player.transform.position;

Vector2 direction = (targetPos - myPos).normalized;

spell.GetComponent<Rigidbody2D>().velocity = direction \* projectileForce;

spell.GetComponent<TestEnemyProjectile>().damage = Random.Range(minDamage, maxDamage);

StartCoroutine(ShootPlayer());

}

}

}

1. Создал скрипт TestEnemyProjectile, наносящий урон игроку при столкновении с ним.

Листинг TestEnemyProjectile.cs

using UnityEngine;

public class TestEnemyProjectile : MonoBehaviour

{

public float damage;

private void OnTriggerEnter2D(Collider2D collision)

{

if (collision.tag != "Enemy")

if (collision.tag == "Player")

PlayerStats.playerStats.DealDamage(damage);

Destroy(gameObject);

}

}

1. Создал скрипт EnemySpawner, создающий врагов в случайной точке сцены.

Листинг EnemySpawner.cs

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class EnemySpawner : MonoBehaviour

{

public List<GameObject> Enemies = new List<GameObject>();

public float spawnRate;

private float x, y;

private Vector3 spawnPos;

private void Start()

{

StartCoroutine(SpawnTestEnemy());

}

IEnumerator SpawnTestEnemy()

{

x = Random.Range(-1, 1);

y = Random.Range(-1, 1);

spawnPos.x += x;

spawnPos.y += y;

Instantiate(Enemies[0], spawnPos, Quaternion.identity);

yield return new WaitForSeconds(spawnRate);

StartCoroutine(SpawnTestEnemy());

}

}

1. Создал скрипт FloatToPlayer, притягивающий предмет к игроку.

Листинг FloatToPlayer.cs

using UnityEngine;

public class FloatToPlayer : MonoBehaviour

{

public GameObject player;

public float speed;

void Start()

{

player = GameObject.Find("Player");

}

void Update()

{

if (player != null)

transform.position = Vector3.MoveTowards(transform.position, player.transform.position, speed \* Time.deltaTime);

}

}

1. Создал скрипт PlayerMovement, отвечающий за передвижение игрока и его анимацию.

Листинг PlayerMovement.cs

using UnityEngine;

public class PlayerMovement : MonoBehaviour

{

public float dashRange;

public float speed;

private Vector2 direction;

private Animator animator;

private Vector2 targetPos;

private enum Facing { UP, DOWN, LEFT, RIGHT }

private Facing FacingDir = Facing.DOWN;

private void Start()

{

animator = GetComponent<Animator>();

}

void Update()

{

TakeInput();

Move();

}

private void Move()

{

transform.Translate(direction \* speed \* Time.deltaTime);

if (direction.x != 0 || direction.y != 0)

SetAnimatorMovement(direction);

else

animator.SetLayerWeight(1, 0);

}

private void TakeInput()

{

direction = Vector2.zero;

if (Input.GetKey(KeyCode.W))

{

direction += Vector2.up;

FacingDir = Facing.UP;

}

if (Input.GetKey(KeyCode.A))

{

direction += Vector2.left;

FacingDir = Facing.LEFT;

}

if (Input.GetKey(KeyCode.S))

{

direction += Vector2.down;

FacingDir = Facing.DOWN;

}

if (Input.GetKey(KeyCode.D))

{

direction += Vector2.right;

FacingDir = Facing.RIGHT;

}

if (Input.GetKeyDown(KeyCode.Space))

{

Vector2 currentPos = transform.position;

targetPos = Vector2.zero;

if (FacingDir == Facing.UP)

targetPos.y = 1;

else if (FacingDir == Facing.DOWN)

targetPos.y = -1;

else if (FacingDir == Facing.LEFT)

targetPos.x = -1;

else if (FacingDir == Facing.RIGHT)

targetPos.x = 1;

transform.Translate(targetPos \* dashRange);

}

}

private void SetAnimatorMovement(Vector2 direction)

{

animator.SetLayerWeight(1, 1);

animator.SetFloat("xDir", direction.x);

animator.SetFloat("yDir", direction.y);

}

}

1. Создал скрипт PlayerStats, содержащий информацию о показателях игрока и их изменение.

Листинг PlayerStats.cs

using UnityEngine;

using UnityEngine.UI;

public class PlayerStats : MonoBehaviour

{

public static PlayerStats playerStats;

public GameObject player;

public Text healthText;

public Slider healthSlider;

public float health;

public float maxHealth;

public int coins;

public int gems;

public Text coinsValue;

public Text gemsValue;

private void Awake()

{

if (playerStats != null)

Destroy(playerStats);

else

playerStats = this;

DontDestroyOnLoad(this);

}

private void Start()

{

health = maxHealth;

SetHealthUI();

}

public void DealDamage(float damage)

{

health -= damage;

CheckDeath();

SetHealthUI();

}

public void HealCharacter(float heal)

{

health += heal;

CheckOverheal();

SetHealthUI();

}

private void SetHealthUI()

{

healthSlider.value = CalculateHealthPercentage();

healthText.text = Mathf.Ceil(health).ToString() + " / " + Mathf.Ceil(maxHealth).ToString();

}

public void CheckOverheal()

{

if (health > maxHealth)

health = maxHealth;

}

private void CheckDeath()

{

if (health <= 0)

{

health = 0;

Destroy(player);

}

}

private float CalculateHealthPercentage()

{

return health / maxHealth;

}

public void AddCurrency(CurrencyPickup currency)

{

if (currency.currentObject == CurrencyPickup.PickupObject.COIN)

{

coins += currency.pickupQuantity;

coinsValue.text = "Gold: " + coins;

}

else if (currency.currentObject == CurrencyPickup.PickupObject.GEM)

{

gems += currency.pickupQuantity;

gemsValue.text = "Gems: " + gems;

}

}

}

1. Создал скрипт TestProjectile, отвечающий за нанесение урона врагам при соприкосновении.

Листинг TestProjectile.cs

using UnityEngine;

public class TestProjectile : MonoBehaviour

{

public float damage;

private void OnTriggerEnter2D(Collider2D collision)

{

if (collision.name != "Player")

if (collision.GetComponent<Enemy>() != null)

collision.GetComponent<Enemy>().DealDamage(damage);

Destroy(gameObject);

}

}

1. Создал скрипт TestSpell, отвечающий за стрельбу игрока снарядами в направлении курсора.

Листинг TestSpell.cs

using UnityEngine;

public class TestSpell : MonoBehaviour

{

public GameObject projectile;

public float minDamage;

public float maxDamage;

public float projectileForce;

void Update()

{

if (Input.GetMouseButtonDown(1))

{

GameObject spell = Instantiate(projectile, transform.position, Quaternion.identity);

Vector2 mousePos = Camera.main.ScreenToWorldPoint(Input.mousePosition);

Vector2 myPos = transform.position;

Vector2 direction = (mousePos - myPos).normalized;

spell.GetComponent<Rigidbody2D>().velocity = direction \* projectileForce;

spell.GetComponent<TestProjectile>().damage = Random.Range(minDamage, maxDamage);

}

}

}

1. Вывод

В ходе проделанной работы были приобретены навыки в разработке механики 2D рогалика.