Лабораторная работа № 23

**Тема:** разработка игрового проекта “ hop”

**Цель:** приобрести навыки в разработке игрового проекта “ hop”

Ход работы

Выполнение работы

Импортирование ресурсов игры

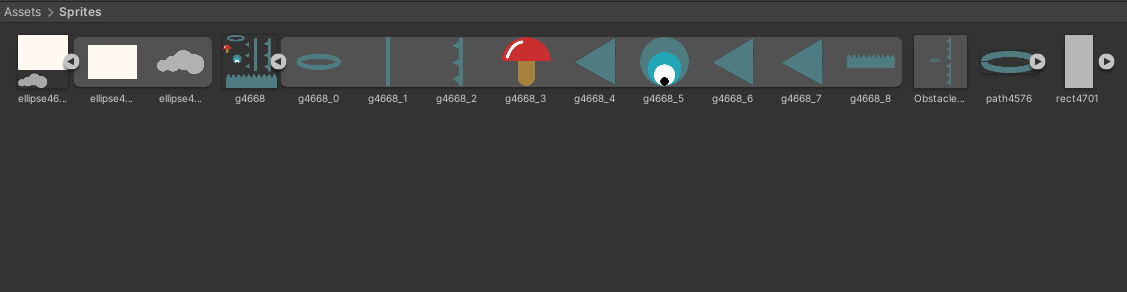


Рис. 23.1 – Папка Sprites

Организация на сцене препятствий с импортируемыми спрайтами

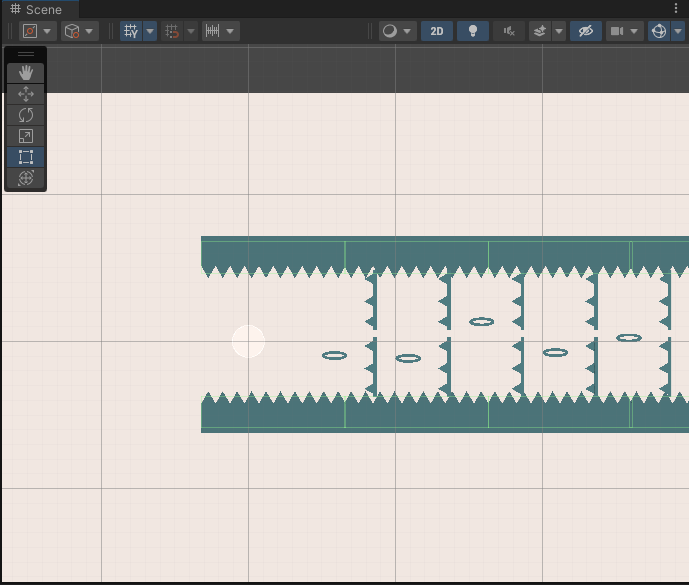


Рис. 23.2 – Окно Scene

Создаем игровой объект Player с соответствующими компонентами и спрайтом

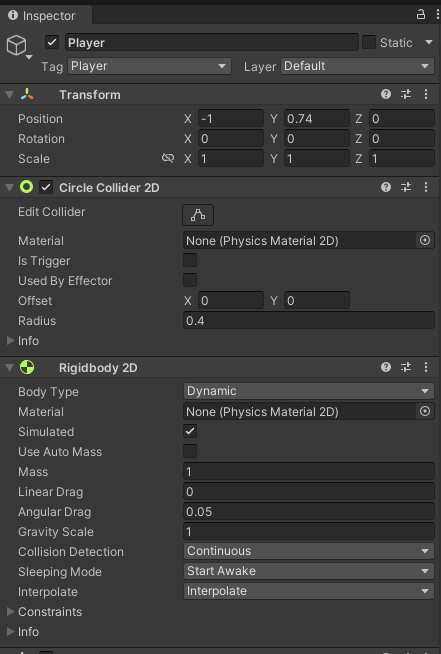


Рис. 23.3 – Inspector Player

Через Package Manager добавляем пакет Cinemachine. Через это же пакет создаем и настраиваем камеры



Рис. 23.4 – Окно Hierarchy Camera с дочерними объектами

Создаем анимацию смерти игрока и в Animator соединяем с конкретным действием

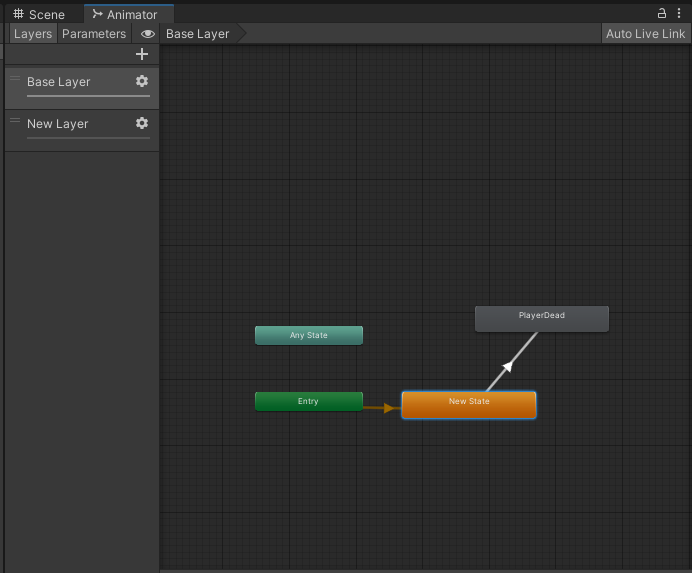


Рис. 23.5– Окно Animator

Листинг 23.1 Player.cs

|  |
| --- |
| using UnityEngine;  public class Player : MonoBehaviour  {  public Vector2 jumpDirection;  public int jumpForce;  Rigidbody2D rgbd;  Vector2 currentSpeed;  public Vector2 maxSpeed;  public bool takePlayerInput;  Animator animator;  GameManager gameManager;  // Start is called before the first frame update  void Start()  {  rgbd = GetComponent<Rigidbody2D>();  rgbd.gravityScale = 0;  animator = GetComponent<Animator>();  gameManager = FindObjectOfType<GameManager>();  }  void Update()  { if (takePlayerInput) return;  if (Input.GetMouseButtonDown(0))  {  if (rgbd.gravityScale != 1) { rgbd.gravityScale = 1; }  rgbd.AddForce(jumpDirection \* jumpForce \* Time.deltaTime);  ControlSpeed();  }  }  //control the speed of ball  void ControlSpeed()  {  currentSpeed = rgbd.velocity;  if (currentSpeed.x != maxSpeed.x) { currentSpeed.x = maxSpeed.x; }  if (currentSpeed.y != maxSpeed.y) { currentSpeed.y = maxSpeed.y; }  rgbd.velocity = currentSpeed;  }  public void PlayerDead()  {  //PlayDead Animation  //Restart Scene  Invoke("RestartScene", 2);  animator.SetTrigger("Dead");  }  public void RestartScene()  {  gameManager.RestartScene();  }  } |

Листинг 23.2 Obstacle.cs

|  |
| --- |
| using UnityEngine;  public class Obstacle : MonoBehaviour  {  public void OnCollisionEnter2D(Collision2D collision)  {  if (collision.collider.tag == "Player")  {  FindObjectOfType<Player>().PlayerDead();  }  }  public void RemoveObstacle()  {  //ball hit hoot remove it  Destroy(gameObject);  }  } |

Листинг 23.3 HoopHolder.cs

|  |
| --- |
| using UnityEngine;  public class HoopHolder : MonoBehaviour  {  public Transform obstacle;  bool obstacleCompleted;  public void OnCollisionEnter2D(Collision2D collision)  {  if (collision.collider.tag == "Player")  {  //take control of player  FindObjectOfType<Player>().takePlayerInput = true;  }  }  public void OnTriggerEnter2D(Collider2D collision)  {  if (obstacleCompleted) return;  if (collision.tag == "Player")  {  FindObjectOfType<Player>().takePlayerInput = false;  obstacle.GetComponent<Obstacle>().RemoveObstacle();  DestroyGameObject();  }  }  public void DestroyGameObject()  {  GetComponentInParent<ObstacleHolder>().PlayDeadAnimation();  Destroy(gameObject);  }  } |

Листинг 23.4 ObstacleHolder.cs

|  |
| --- |
| using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class ObstacleHolder : MonoBehaviour  {  Animator animator;  // Start is called before the first frame update  void Start()  {  animator = GetComponent<Animator>();  }  public void PlayDeadAnimation()  {  animator.SetTrigger("Destroy");  Destroy(gameObject, 2f);  }  } |

Листинг 23.5 GameManager.cs

|  |
| --- |
| using UnityEngine.SceneManagement;  using UnityEngine;  public class GameManager : MonoBehaviour  {  // Start is called before the first frame update  void Start()  {    }  public void RestartScene()  {  SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);  }  } |

Листинг 23.6 ObjectHolderInstantiator.cs

|  |
| --- |
| using UnityEngine;  public class ObjectHolderInstantiator : MonoBehaviour  {  public GameObject obstacle;  public int distanceBetweenObstacle;  int currentObstaclePosition;  public Vector2 minMaxYValue;  // Start is called before the first frame update  void Start()  {  InstantiateObstacle();  }    public void InstantiateObstacle()  {  for (int i = 0; i < 50; i++)  {  currentObstaclePosition += distanceBetweenObstacle;  GameObject GO = Instantiate(obstacle, new Vector3(currentObstaclePosition, 0, 0),  Quaternion.identity) as GameObject;  GO.transform.GetChild(1).transform.position =  new Vector2(GO.transform.GetChild(1).position.x, Random.Range(minMaxYValue.x, minMaxYValue.y));  }  }  } |

Вывод: Приобрел навыки разработки игры “ hophophop”