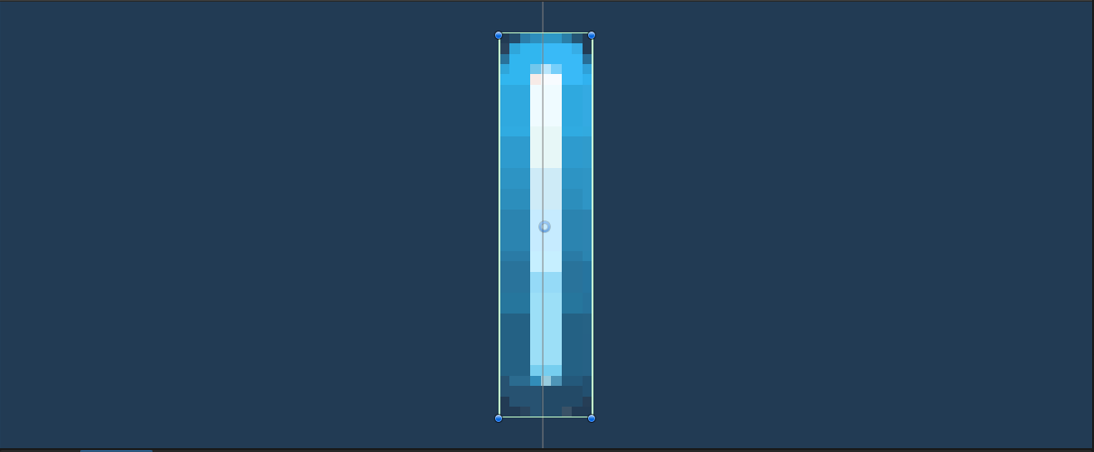
**Лабораторное занятие №12-14.**

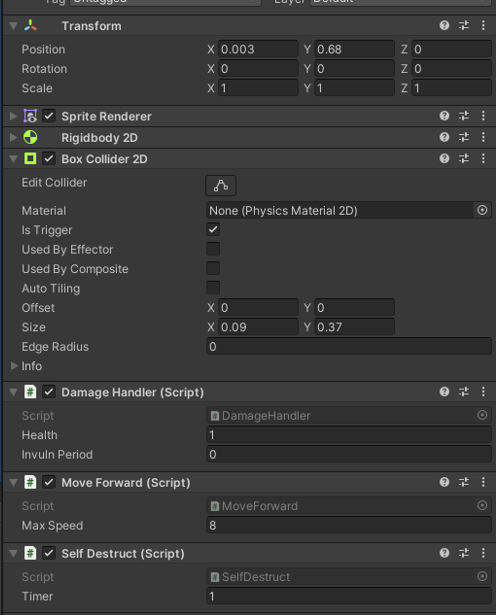
**Тема: разработать игру «Space Shooter»**

**Цель: разработать игру «Space Shooter»**

Ход работы:



**Рис.12.1.** Prefab Лазера.



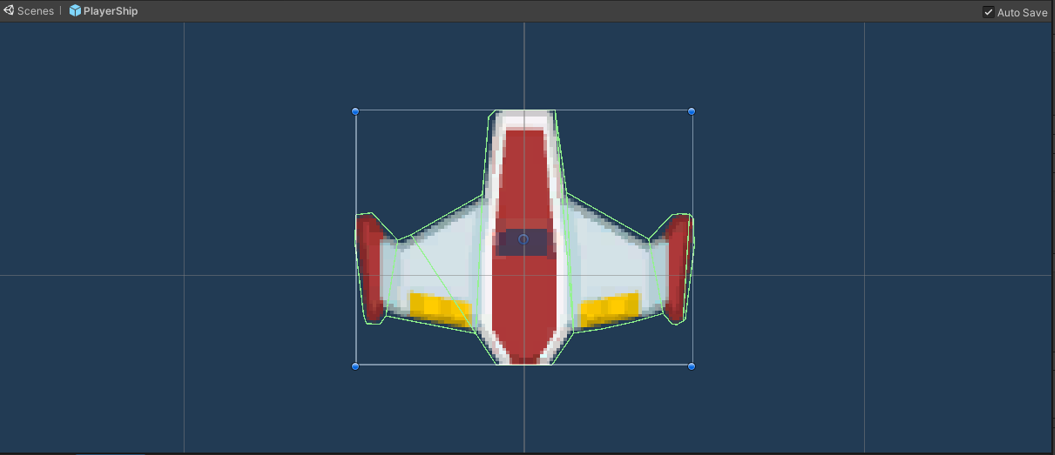
**Рис.12.2.** Компоненты Prefab Лазера.



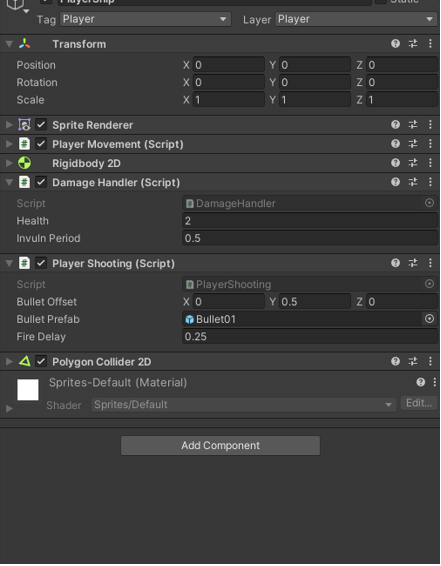
**Рис.12.3.** Prefab Противника.



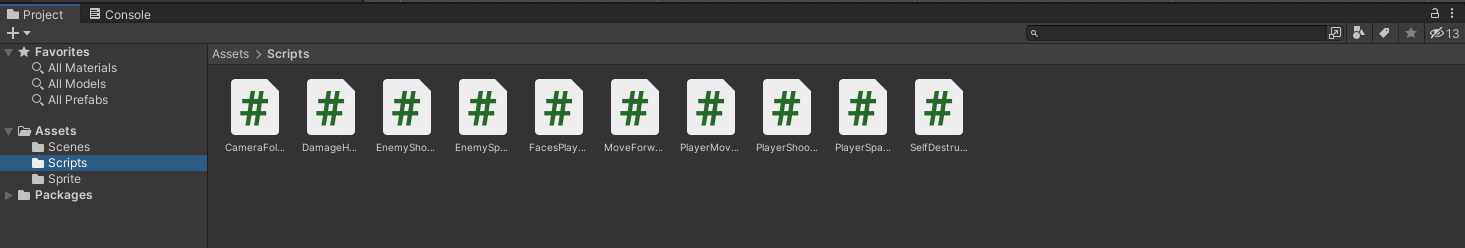
**Рис.12.4.** Компоненты Prefab Противника.



**Рис.12.5.** Prefab Игрока.



**Рис.12.6.** Компоненты Prefab Игрока.



**Рис.12.7.** Создание Скриптов.

**Листинг CameraFollow**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class CameraFollow : MonoBehaviour

{

public Transform myTarget;

// Update is called once per frame

void Update()

{

if(myTarget != null)

{

Vector3 targPos = myTarget.position;

targPos.z = transform.position.z;

transform.position = targPos;

}

}

}

**Листинг DamageByColision**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class DamageHandler : MonoBehaviour

{

public int health = 1;

public float invulnPeriod = 0;

float invulnTimer = 0;

int correctLayer;

SpriteRenderer spriteRend;

void Start ()

{

correctLayer = gameObject.layer;

spriteRend = GetComponent<SpriteRenderer>();

if (spriteRend == null)

{

spriteRend = transform.GetComponentInChildren<SpriteRenderer>();

if (spriteRend == null)

{

Debug.LogError("Object'" + gameObject.name + "' has no sprite renderer.");

}

}

}

private void OnTriggerEnter2D()

{

health--;

if (invulnPeriod > 0)

{

invulnTimer = invulnPeriod;

gameObject.layer = 10;

}

}

void Update()

{

if (invulnTimer > 0)

{

invulnTimer -= Time.deltaTime;

if (invulnTimer <= 0)

{

gameObject.layer = correctLayer;

if(spriteRend != null)

{

spriteRend.enabled = true;

}

}

else

{

if(spriteRend != null)

{

spriteRend.enabled = !spriteRend.enabled;

}

}

}

if (health <= 0)

{

Die();

}

}

void Die()

{

Destroy(gameObject);

}

}

**Листинг EnemyShooting**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class EnemyShooting : MonoBehaviour

{

public Vector3 bulletOffset = new Vector3(0, 0.5f, 0);

public GameObject bulletPrefab;

int bulletLayer;

public float fireDelay = 0.50f;

float cooldownTimer = 0;

void Start()

{

bulletLayer = gameObject.layer;

}

void Update()

{

cooldownTimer -= Time.deltaTime;

if (cooldownTimer <= 0)

{

cooldownTimer = fireDelay;

Vector3 offset = transform.rotation \* new Vector3(0, 0.5f, 0);

GameObject bulletGO = (GameObject) Instantiate(bulletPrefab, transform.position + offset, transform.rotation);

bulletGO.layer = bulletLayer;

}

}

}

**Листинг EnemySpawner**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class EnemySpawner : MonoBehaviour

{

public GameObject enemyPrefab;

float spawnDistance = 12f;

float enemyRate = 5;

float nextEnemy = 1;

void Update()

{

nextEnemy -= Time.deltaTime;

if (nextEnemy <= 0)

{

nextEnemy = enemyRate;

enemyRate \*= 0.9f;

if (enemyRate < 2)

enemyRate = 2;

Vector3 offset = Random.onUnitSphere;

offset.z = 0;

offset = offset.normalized \* spawnDistance;

Instantiate(enemyPrefab, transform.position + offset, Quaternion.identity);

}

}

}

**Листинг FacesPlayer**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class FacesPlayer : MonoBehaviour

{

public float rotSpeed = 90f;

Transform player;

void Update()

{

if(player == null)

{

GameObject go = GameObject.FindWithTag ("Player");

if(go != null)

{

player = go.transform;

}

}

if (player == null)

return;

Vector3 dir = player.position - transform.position;

dir.Normalize();

float zAngle = Mathf.Atan2(dir.y, dir.x) \* Mathf.Rad2Deg - 90;

Quaternion desiredRot = Quaternion.Euler(0, 0, zAngle);

transform.rotation = Quaternion.RotateTowards(transform.rotation, desiredRot, rotSpeed \* Time.deltaTime);

}

}

**Листинг MoveForward**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class MoveForward : MonoBehaviour

{

public float maxSpeed = 5f;

void Update()

{

Vector3 pos = transform.position;

Vector3 velocity = new Vector3(0, maxSpeed \* Time.deltaTime, 0);

pos += transform.rotation \* velocity;

transform.position = pos;

}

}

**Листинг PlayerMovement**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class PlayerMovement : MonoBehaviour

{

public float maxSpeed = 5f;

public float rotSpeed = 180f;

float shipBoundaryRadius = 0.5f;

void Start()

{

}

// Update is called once per frame

void Update()

{

Quaternion rot = transform.rotation;

float z = rot.eulerAngles.z;

z -= Input.GetAxis("Horizontal") \* rotSpeed \* Time.deltaTime;

rot = Quaternion.Euler(0, 0, z);

transform.rotation = rot;

Vector3 pos = transform.position;

Vector3 velocity = new Vector3(0, Input.GetAxis("Vertical") \* maxSpeed \* Time.deltaTime, 0);

pos += rot \* velocity;

if(pos.y + shipBoundaryRadius > Camera.main.orthographicSize)

{

pos.y = Camera.main.orthographicSize - shipBoundaryRadius;

}

if (pos.y - shipBoundaryRadius < -Camera.main.orthographicSize)

{

pos.y = -Camera.main.orthographicSize + shipBoundaryRadius;

}

float screenRatio = (float)Screen.width / (float)Screen.height;

float width0rtho = Camera.main.orthographicSize \* screenRatio;

if (pos.x + shipBoundaryRadius > width0rtho)

{

pos.x = width0rtho - shipBoundaryRadius;

}

if (pos.x - shipBoundaryRadius < -width0rtho)

{

pos.x = -width0rtho + shipBoundaryRadius;

}

transform.position = pos;

}

}

**Листинг PlayerShooting**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class PlayerShooting : MonoBehaviour

{

public Vector3 bulletOffset = new Vector3(0, 0.5f, 0);

public GameObject bulletPrefab;

int bulletLayer;

public float fireDelay = 0.25f;

float cooldownTimer = 0;

void Start()

{

bulletLayer = gameObject.layer;

}

void Update()

{

cooldownTimer -= Time.deltaTime;

if (Input.GetButton("Fire1") && cooldownTimer <= 0)

{

cooldownTimer = fireDelay;

Vector3 offset = transform.rotation \* new Vector3(0, 0.5f, 0);

GameObject bulletGO = (GameObject)Instantiate(bulletPrefab, transform.position + offset, transform.rotation);

bulletGO.layer = gameObject.layer;

}

}

}

**Листинг PlayerSpawner**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class PlayerSpawner : MonoBehaviour

{

public GameObject playerPrefab;

GameObject playerInstance;

public int numLives = 4;

float respawnTimer = 1;

void Start()

{

SpawnPlayer();

}

void SpawnPlayer()

{

numLives--;

respawnTimer = 1;

playerInstance = (GameObject)Instantiate(playerPrefab, transform.position, Quaternion.identity);

}

void Update()

{

if(playerInstance == null && numLives > 0)

{

respawnTimer -= Time.deltaTime;

if (respawnTimer <= 0)

{

SpawnPlayer();

}

}

}

void OnGUI()

{

if (numLives > 0 || playerInstance != null)

{

GUI.Label(new Rect(0, 0, 100, 50), "Lives Left: " + numLives);

}

else

{

GUI.Label(new Rect(Screen.width/2 - 50, Screen.height/2 - 25, 100, 50), "Game Over, Man!");

}

}

}

**Листинг SelfDestruct**

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class SelfDestruct : MonoBehaviour

{

public float timer = 1f;

void Update()

{

timer -= Time.deltaTime;

if(timer <= 0)

{

Destroy(gameObject);

}

}

}