**Лабораторное занятие № 27.**

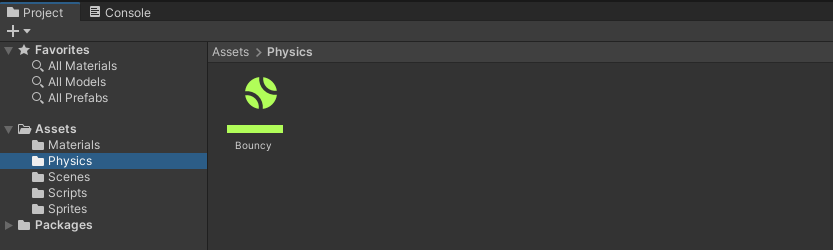
Лист

**Тема: Разработка игры “ PingPong”**

**Цель: Разработать игру “ PingPong ”**

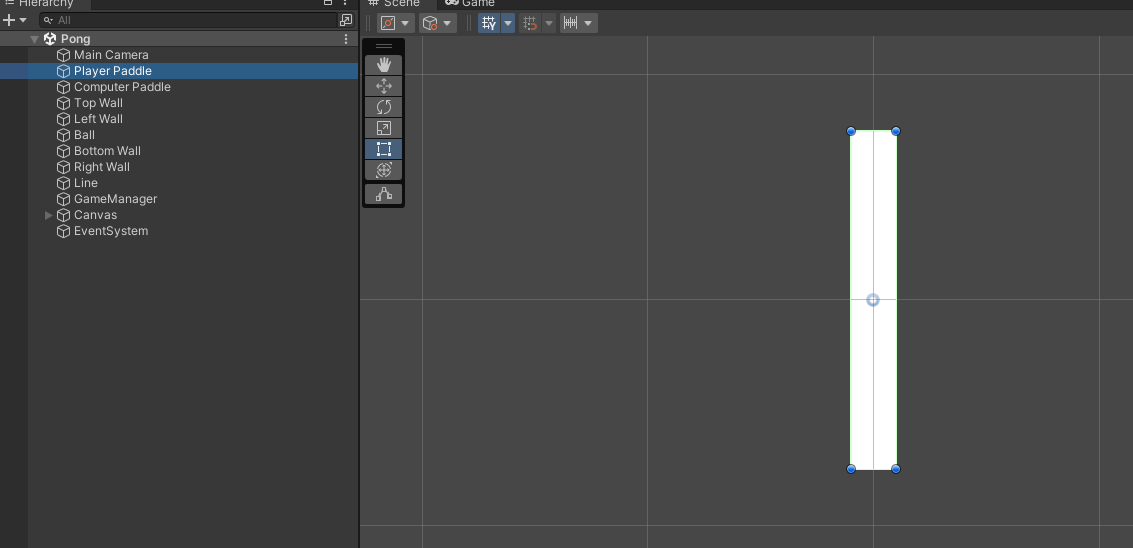
Ход работы:

1.Перекидываю ресурсы (Asset)



**Рис.27.1** – Asset.

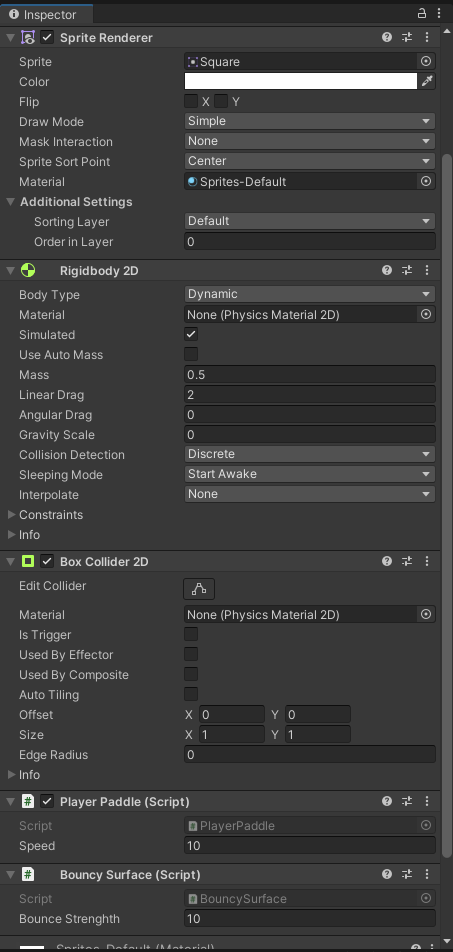
2.Создание игрока (Player).



**Рис.27.2** – Player.

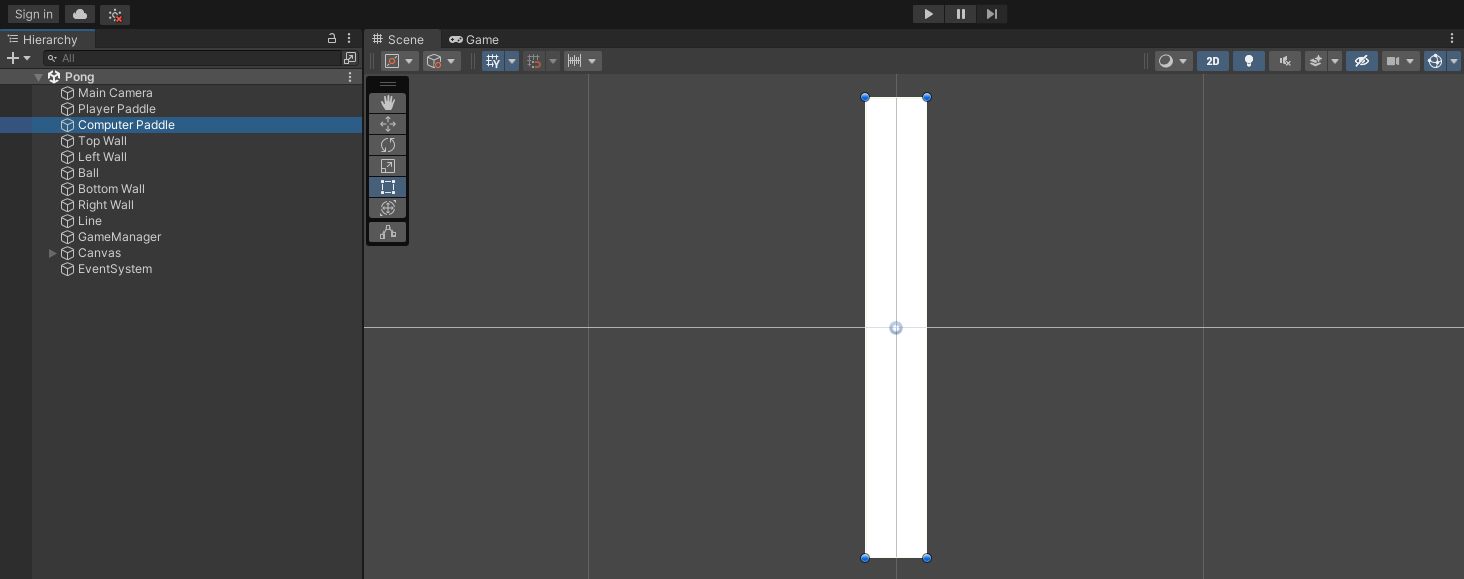
3.Inspector игрока (Player).

Лист



**Рис.27.3** – Inspector Player.

4.Создание противника.



**Рис.27.4** – Computer Paddle.

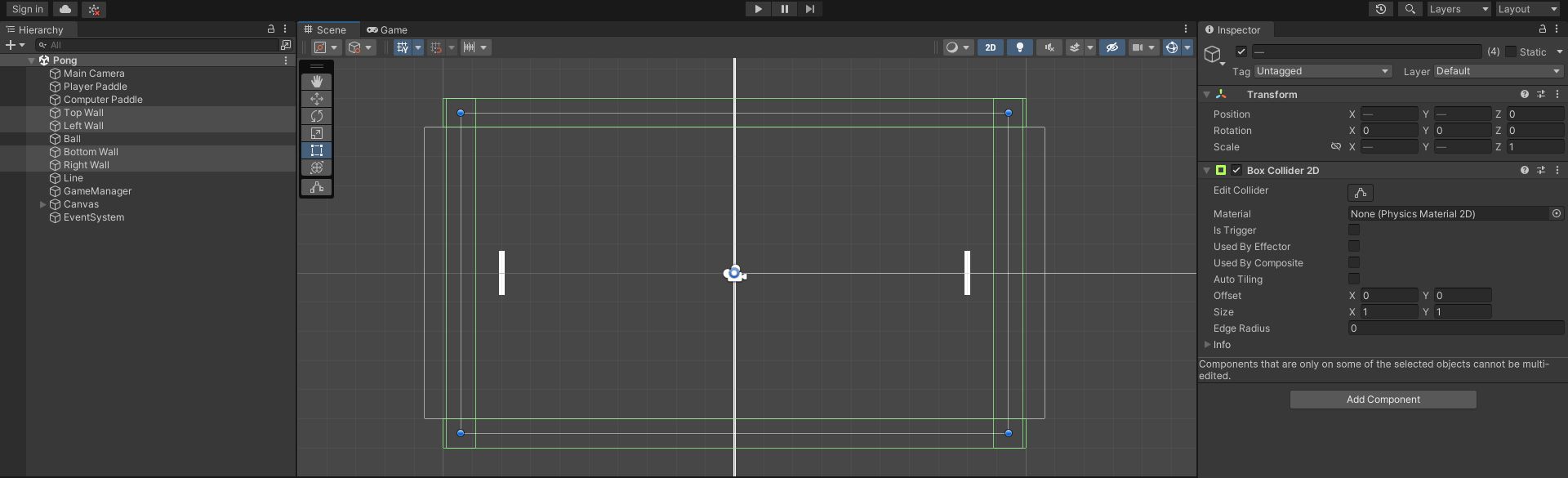
5.Inspector Противника.

Лист



**Рис.27.5** – Inspector Computer Paddle.

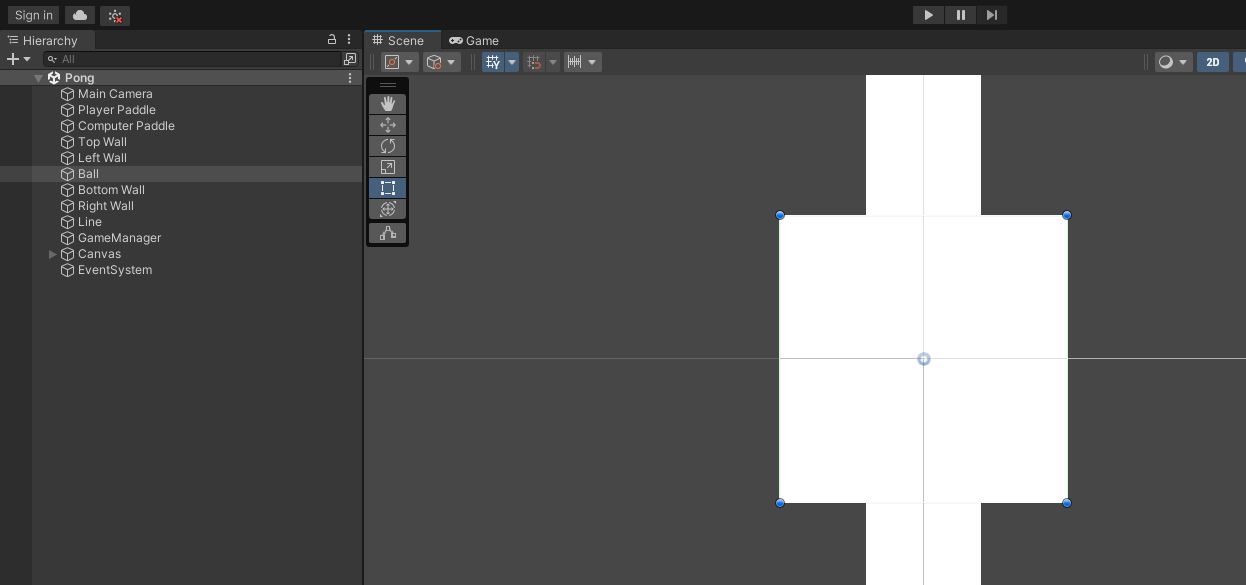
6.Создание границ поля.



**Рис.27.6** – Границы поля.

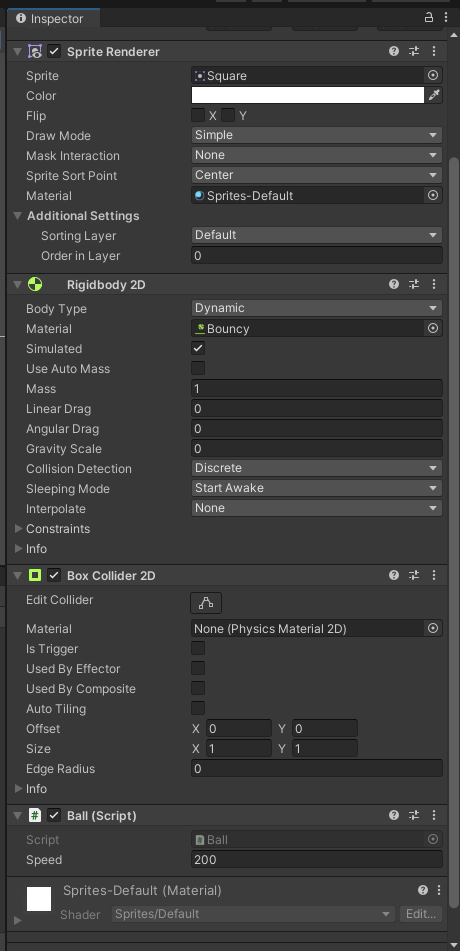
7.Создание мячика (Ball)

Лист



**Рис.27.7** – Ball.

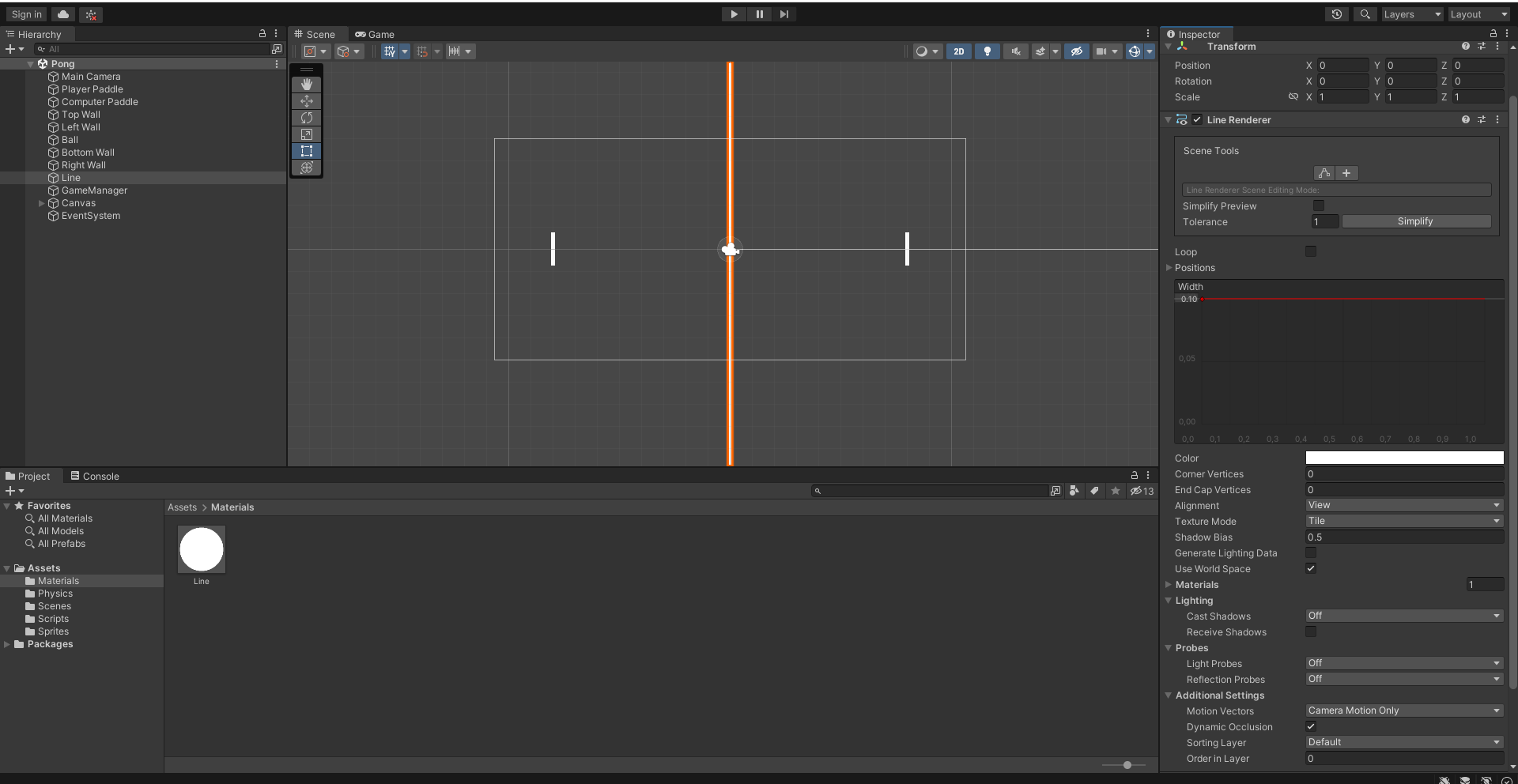
7.Inspector мячика (Ball).



**Рис.27.7** – Inspector Ball.

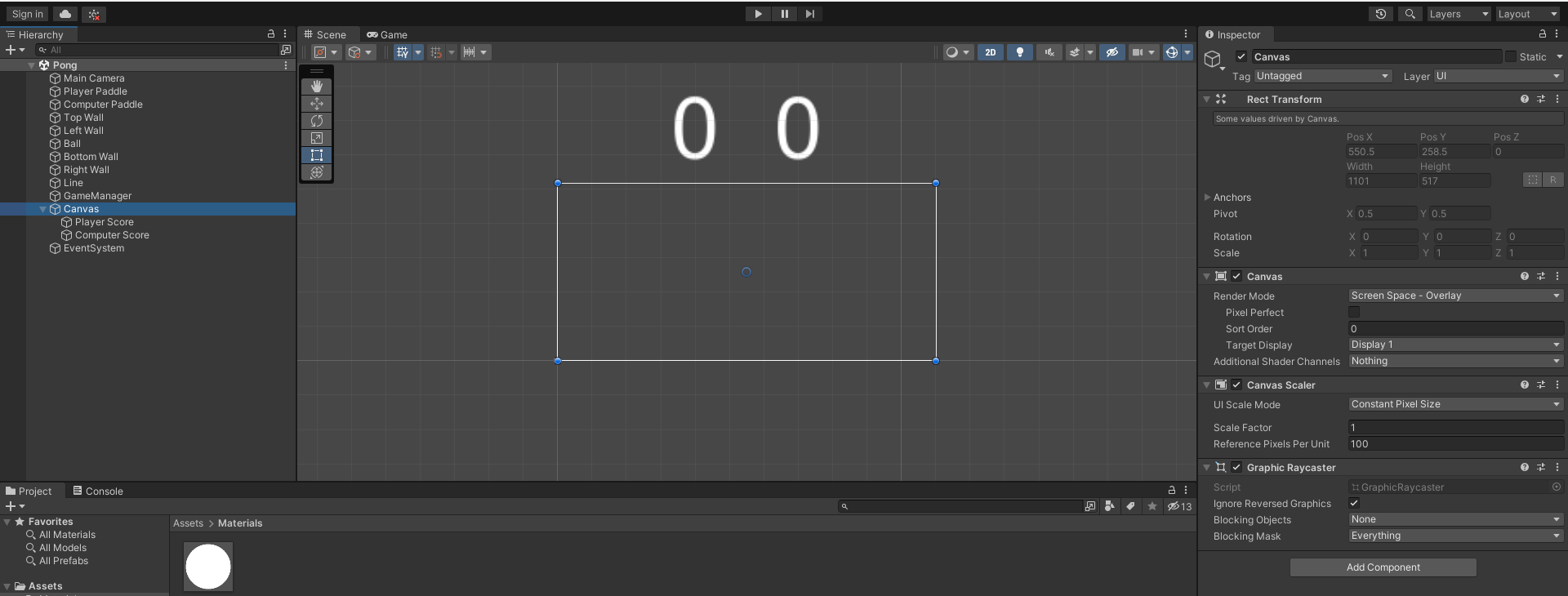
8.Разделение полей (Line).

Лист



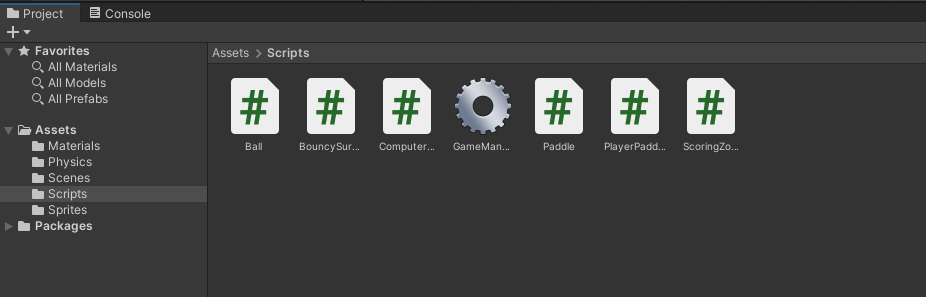
**Рис.27.8** – Line.

9.Создание счёта (Для игрока и компьютера).



**Рис.27.9** – Canvas.

10.Создание скриптов.



**Рис.27.10** – Scripts.

**Скрипты:**

Лист

Ball.cs

using UnityEngine;

public class Ball : MonoBehaviour

{

public float speed = 200.0f;

private Rigidbody2D \_rigidbody;

private void Awake()

{

\_rigidbody = GetComponent<Rigidbody2D>();

}

private void Start()

{

ResetPosition();

AddStartingForce();

}

public void ResetPosition()

{

\_rigidbody.position = Vector3.zero;

\_rigidbody.velocity = Vector3.zero;

}

public void AddStartingForce()

{

float x = Random.value < 0.5f ? -1.0f : 1.0f;

float y = Random.value < 0.5f ? Random.Range(-1.0f, -0.5f) :

Random.Range(0.5f, 1.0f);

Vector2 direction = new Vector2(x, y);

\_rigidbody.AddForce(direction \* this.speed);

}

public void AddForce(Vector2 force)

{

\_rigidbody.AddForce(force);

}

}

BouncySurface.cs

using UnityEngine;

public class BouncySurface : MonoBehaviour

{

public float bounceStrenghth;

private void OnCollisionEnter2D(Collision2D collision)

{

Ball ball = collision.gameObject.GetComponent<Ball>();

if (ball != null)

{

Vector2 normal = collision.GetContact(0).normal;

ball.AddForce(-normal \* this.bounceStrenghth);

}

}

}

Лист

ComputerPadle.cs

using UnityEngine;

public class ComputerPaddle : Paddle

{

public Rigidbody2D ball;

private void FixedUpdate()

{

if (this.ball.velocity.x > 0.0f)

{

if (this.ball.position.y > this.transform.position.y)

{

\_rigidbody.AddForce(Vector2.up \* this.speed);

}

else if (this.ball.position.y < this.transform.position.y)

{

\_rigidbody.AddForce(Vector2.down \* this.speed);

}

}

else

{

if (this.transform.position.y > 0.0f)

{

\_rigidbody.AddForce(Vector2.down \* this.speed);

}

else if (this.transform.position.y < 0.0f)

{

\_rigidbody.AddForce(Vector2.up \* this.speed);

}

}

}

}

GameManager.cs

using UnityEngine;

using UnityEngine.UI;

public class GameManager : MonoBehaviour

{

public Ball ball;

public Paddle playerPaddle;

public Paddle computerPaddle;

public Text playerScoreText;

public Text computerScoreText;

private int \_playerScore;

private int \_computerScore;

public void PlayerScores()

{

\_playerScore++;

this.playerScoreText.text = \_playerScore.ToString();

ResetRound();

}

public void ComputerScores()

Лист

{

\_computerScore++;

this.computerScoreText.text = \_computerScore.ToString();

ResetRound();

}

private void ResetRound()

{

this.playerPaddle.ResetPosition();

this.computerPaddle.ResetPosition();

this.ball.ResetPosition();

this.ball.AddStartingForce();

}

}

Paddle.cs

using UnityEngine;

public class Paddle : MonoBehaviour

{

public float speed = 10.0f;

protected Rigidbody2D \_rigidbody;

private void Awake()

{

\_rigidbody = GetComponent<Rigidbody2D>();

}

public void ResetPosition()

{

\_rigidbody.position = new Vector2(\_rigidbody.position.x, 0.0f);

\_rigidbody.velocity = Vector2.zero;

}

}

PlayerPaddle.cs

using UnityEngine;

public class PlayerPaddle : Paddle

{

private Vector2 \_direction;

private void Update()

{

if (Input.GetKey(KeyCode.W) || Input.GetKey(KeyCode.UpArrow))

{

\_direction = Vector2.up;

}

else if (Input.GetKey(KeyCode.S) || Input.GetKey(KeyCode.DownArrow))

{

\_direction = Vector2.down;

}

else

{

\_direction = Vector2.zero;

}

}

private void FixedUpdate()

{

if (\_direction.sqrMagnitude != 0)

{

\_rigidbody.AddForce(\_direction \* this.speed);

}

}

}

ScoringZone.cs

using UnityEngine;

using UnityEngine.EventSystems;

public class ScoringZone : MonoBehaviour

{

public EventTrigger.TriggerEvent scoreTrigger;

public void OnCollisionEnter2D(Collision2D collision)

{

Ball ball = collision.gameObject.GetComponent<Ball>();

if (ball != null)

{

BaseEventData eventData = new BaseEventData(EventSystem.current);

this.scoreTrigger.Invoke(eventData);

}

}

}

Лист