PROGRESS REPORT

MICROGAME #1: Ping Pong

Andrew Adame

ID: 007516100

LEGEND: COMPLETED – UNFINISHED – WIP - FIX

1. Created new project Ping Pong
2. Create folders containing important assets (Scripts, prefabs, etc)
3. Create basic Ping Pong Game
    a. Design Level
        i. Create walls/borders
            1. Use Box Collider 2D
        ii. Create players/paddles
            1. Use Box Collider 2D
            2. Use Rigidbody 2D
        iii. Create ball
            1. Use Circle Collider 2D
            2. Use Rigidbody 2D
        iv. Create Goal
    b. Start
        i. Game Start
            1. Ball Launches random direction
    c. Paddles
        i. Movement
        ii. Two Player Control
            1. Controller script
    d. Ball
        i. Movement
        ii. Physics
            1. BallController script
            2. ERRORS/PROBLEMS
                a. Ball stagnates (bounces back and fourth in same spot)
                    i. Temporary solution; add reset button?
                b. Ball clips through objects at certain high speeds
    e. Gameplay
        i. Bounce back and fourth
        ii. Speed up over time

1. Utilizes AddForce to multiply speed

iii. GameController script

f. Score Tracking

i. Max score condition

g. End

i. Game over screen

ii. Rematch option

h. **EXTRA**

i. Audio

ii. Player 1 & Player 2 ready condition?

iii. Create controlled speed multiplier

iv. Choose amount of score to end game?

v. Timer option?

vi. Ball speed option?

vii. Restart option?

viii. Quit game option?

ix. Win tracker?

x. Colorful visuals, chill-music focused (inspired by TETRIS EFFECT)

1. Soundtrack (likely can't make own)

a. Chill-step, Lo-Fi, calm electronica, etc…

2. Music synced?

3. Learn particle system

4. Arching/Movement change based on music

i. **SCRIPTS**

i. <u>BallController</u>

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class BallController : MonoBehaviour
{

    public float speed;
    public float randomUp;

    Rigidbody2D ballRigidbody;

    GameController cont;

    void Start()
    {
        ballRigidbody = GetComponent<Rigidbody2D>();
        cont = FindObjectOfType<GameController>();
    }

    private void OnEnable()
    {
```

```
        Invoke("PshBall", 1f);

    }

    private void PshBall()
    {
        int dir = Random.Range(0, 2);   //return 0, 1
        float x, y;
        if(dir == 0)
        {
            x = speed;
        }
        else
        {
            x = -speed;
        }

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

        ballRigidbody.AddForce(new Vector2(x, y));

    }

    // Update is called once per frame
    void Update()
    {

    }

    private void OnCollisionEnter2D(Collision2D collision)
    {
        if(collision.gameObject.CompareTag("Player"))
        {
            Vector2 vel;
            vel.x = ballRigidbody.velocity.x;
            vel.y = ballRigidbody.velocity.y / 2 +
(collision.collider.attachedRigidbody.velocity.y / 2);

            ballRigidbody.velocity = vel;

        }
    }

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if(collision.gameObject.CompareTag("Goal"))
        {

            if(ballRigidbody.velocity.x > 0)
            {
                cont.Score(true);
            }
            else if (ballRigidbody.velocity.x < 0)
            {
                cont.Score(false);
            }

            ballRigidbody.velocity = Vector2.zero;
```

```
            transform.position = Vector3.zero;  // equivalent to Vector3(0,0,0)
            Invoke("PshBall", 2f);
        }
    }
    }
```

ii. GameController

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class GameController : MonoBehaviour
{

    int P1Score, P2Score;
    public int maxScore = 3;

    public Text scoreText;

    public GameObject gameOverUI;
    bool gameOver = false;


    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if (gameOver)
            if (Input.anyKeyDown)
                Restart();
    }

    public void Score(bool P1GetScore)
    {
        if (P1GetScore)
            P1Score++;
        else
            P2Score++;

        if(P1Score >= maxScore)
        {
            scoreText.text = "Player 1 Wins!";
            gameOver = true;
            GameOver();
        }
        else if (P2Score >= maxScore)
        {
            scoreText.text = "Player 2 Wins!";
```

```
            gameOver = true;
            GameOver();
        }
        else
        {
            scoreText.text = P1Score + " : " + P2Score;
        }

    }
    void GameOver()
    {
        gameOver = true;
        gameOverUI.SetActive(true);
        Time.timeScale = 0f;
    }

    void Restart()
    {
        SceneManager.LoadScene("PongV1");
        Time.timeScale = 1f;
    }

}
```

iii. Controller

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Controller : MonoBehaviour
{

    public bool Plyer1;
    public float speed;
    int leftUp, rightUp;
    Rigidbody2D rigidbody;


    void Awake()
    {
        rigidbody = GetComponent<Rigidbody2D>();

    }

    // Update is called once per frame
    void Update()
    {
        if(Plyer1)
        {
            if (Input.GetKey(KeyCode.W))
                leftUp = 1;
            else if (Input.GetKey(KeyCode.S))
                leftUp = -1;
            else
                leftUp = 0;

            // When not pressing key, stop moving
            if (leftUp == 0)
```

```
                rigidbody.velocity = new Vector2(0, rigidbody.velocity.x);

            rigidbody.AddForce(Vector2.up * leftUp * speed * Time.deltaTime);  //(0.1)
* 1 * speed * 0.003
          //(new Vector2(0, leftUp * speed * Time.deltaTime))                    (0 *
1 * speed * 0.03, 1 * 1 * speed * 0.03)

        }
        else   //Plyer2
        {
            if (Input.GetKey(KeyCode.UpArrow))
                rightUp = 1;
            else if (Input.GetKey(KeyCode.DownArrow))
                rightUp = -1;
            else
                rightUp = 0;

            if (rightUp == 0)
                rigidbody.velocity = new Vector2(0, rigidbody.velocity.x);

            rigidbody.AddForce(Vector2.up * rightUp * speed * Time.deltaTime);

        }

    }
}
```