Pontificia Universidad Católica Madre y Maestra

Alexander Ramírez

2017-5706

Programación Aplicada

11/2/2021

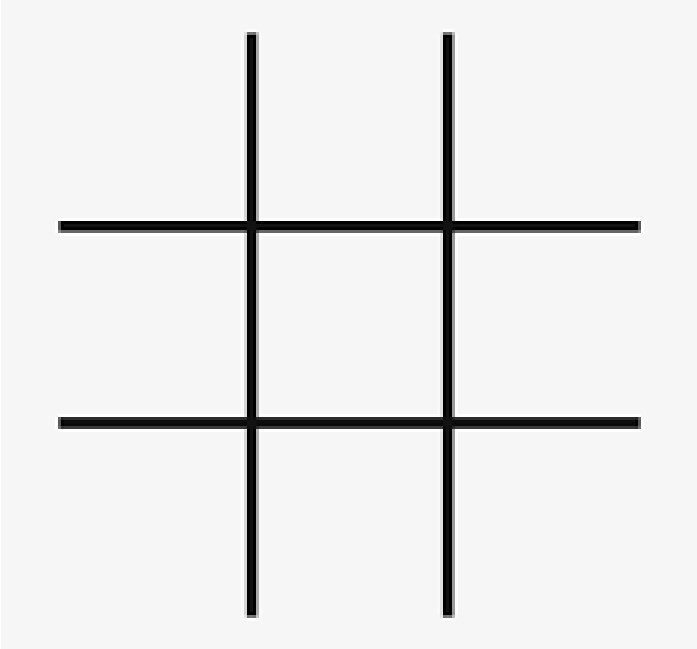
Introducción

En la clase de programación aplicada, se les ha pedido a los estudiantes realizar un juego de Tic Tac Toe con loa integración de Websockets. Este juego debe asegurar que cuando un jugador hace una jugada, el Websocket debe bloquear las actividades de ese jugador para que no juegue 2 veces consecutivas. El juego debió tener dos modos de juego: Humano vs. Computadora y Humano vs Humano.

Desarrollo:

Para esta tarea, decidí utilizar el programa Unity ya que este provee una gran variedad de funciones y utilidades de una manera simple de comprender e implementar.

Este es el fondo que decidí utilizar para el proyecto:



Al conseguir el fondo, procedí a conseguir las figuras que iba a utilizar para este proyecto:

Luego procedí a las funcionalidades (Back End) de las figuras.

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

public class turnscript : MonoBehaviour
{
    SpriteRenderer sr;
    public Sprite[] images;
    GameObject gs;

private void Start(){
    sr.sprite = null;
}

    private void OnMouseDown(){
        int index = gs.GetComponent<gamescript>().PlayerTurn();
        sr.sprite = images[index];
    }

    private void Awake(){
        sr = GetComponent<SpriteRenderer>();
        gs = GameObject.Find("Board");

    }
}

```

En las funcionalidades de las figuras, se implementó una lista donde se puedan guardar las figuras. En la posición 0 esta la figura de la x y en la posición 1 está la figura de la o

A continuación, se creó las funcionalidades para que el juego alterne entra las figuras x y o:

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class gamescript : MonoBehaviour
6   {
7       int SpriteIndex = -1;
8
9       public int PlayerTurn()
10      {
11
12          SpriteIndex++;
13          return SpriteIndex % 2;
14      }
15  }
16
```

Lo que se procedió a hacer fue crear la función PlayerTurn la cual tiene una contadora SpriterIndex la cual se le pide su modulo. Así, cuando el numero es par, el módulo es 0, así, va el jugador x. cuando el jugador de x hace su jugada, la contadora suma 1. Al sumar i, ahora el numero es impar y su modulo es 1. Al ser 1, se llama el elemento o. Así sigue el juego hasta que gane un jugador.