|  |  |  |
| --- | --- | --- |
| **Nombres**: Ricardo Alfonso Pulido Leñero | | **Matrículas**: 2871165 |
| **Nombre del curso**: Programación y prototipo de Videojuegos | **Nombre del profesor**: Yahvé Isaías Solís Aranda | |
| **Módulo**: 2 | **Actividad**: 9 | |
| **Fecha**: 15 de octubre del 2022 | | |
| **Referencias:** | | |

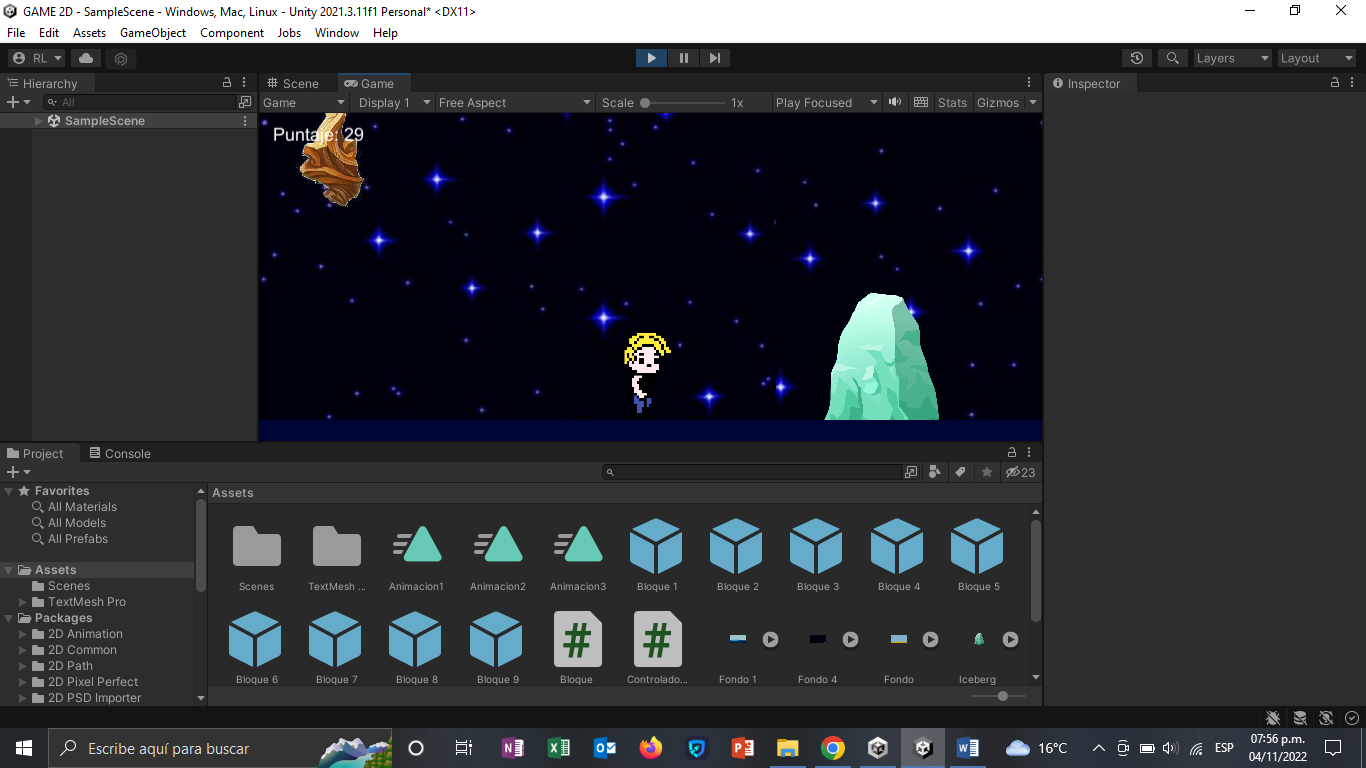
Actividad 9

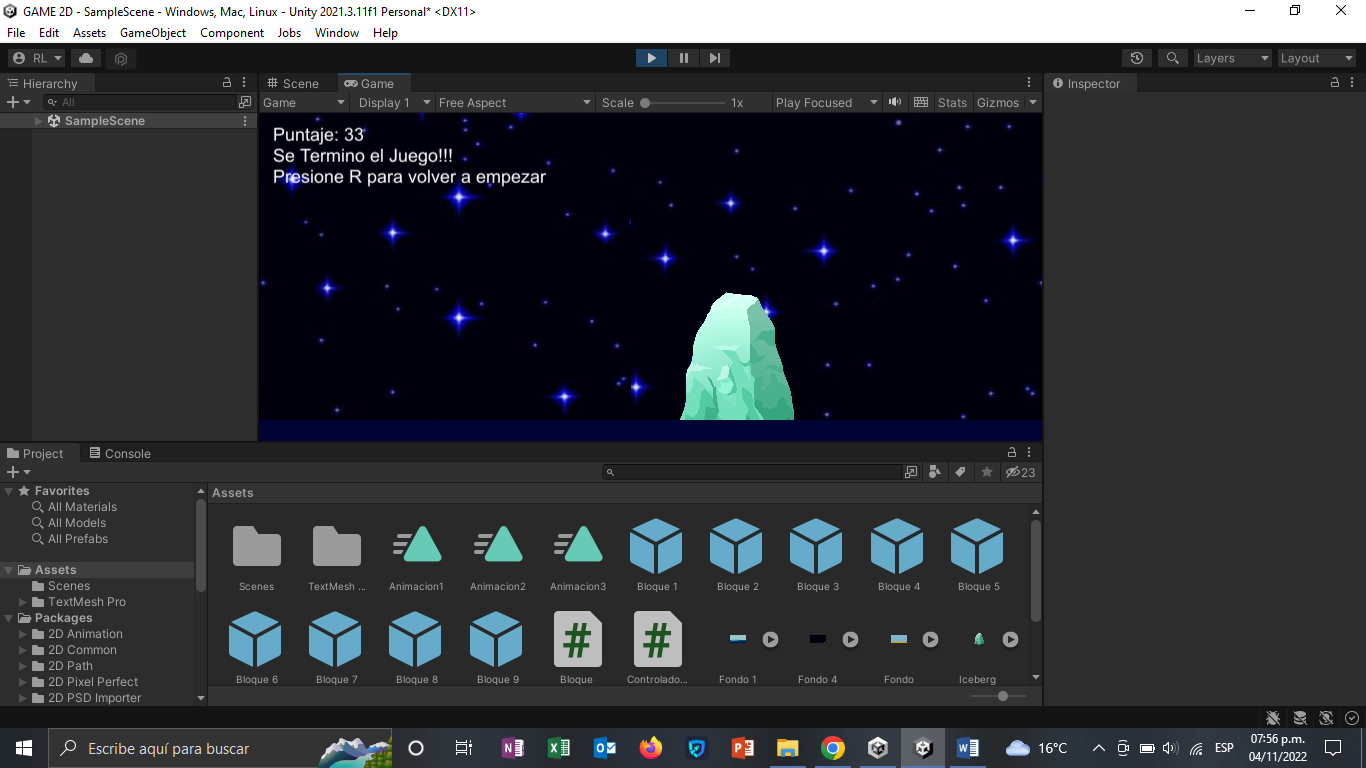
# Controllers e Input

**De acuerdo a lo visto en la sesión, entrega la escena completa del juego en 2D con la movilidad mínima (idle, walk, run, y jump).**

**Entregables:**

**1.Documento en PDF con portada y datos de los integrantes del equipo, en donde se incluya capturas de la escena, de la vista game, así como la máquina de estados diseñada en el animator. Incluir captura de la vista project mostrando las carpetas usadas. Agregar en texto el/los scripts usados.**





Codigos:

|  |
| --- |
| Bloques  using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Bloque : MonoBehaviour {  public int tamaño;  // Start is called before the first frame update  void Start() {    }  // Update is called once per frame  void Update() {    }  } |

|  |
| --- |
| Controlador de Escenario  using System.Collections;  using System.Collections.Generic;  using UnityEngine;  using UnityEngine.UI;  using UnityEngine.SceneManagement;  public class ControladorDeEscenario : MonoBehaviour {  public GameObject Jugador;  public Camera CamaraDeJuego;  public GameObject[] BloquePreFab;  public float PunteroDeJuego;  public float LugarSeguroDeGeneracion = 12;  public Text TextoDeJuego;  public bool Perdiste;  // Start is called before the first frame update  void Start() {  PunteroDeJuego = -7;  Perdiste = false;  }  // Update is called once per frame  void Update() {  if(Jugador!= null) {  CamaraDeJuego.transform.position = new Vector3(  Jugador.transform.position.x,  CamaraDeJuego.transform.position.y,  CamaraDeJuego.transform.position.z  );  TextoDeJuego.text = "Puntaje: " + Mathf.Floor(Jugador.transform.position.x);  }  else  {  if (!Perdiste)  {  Perdiste = true;  TextoDeJuego.text += "\nSe Termino el Juego!!! \nPresione R para volver a empezar";  }  if (Perdiste)  {  if (Input.GetKeyDown("r"))  {  SceneManager.LoadScene(SceneManager.GetActiveScene().name);  }  }  }  while (Jugador != null && PunteroDeJuego < Jugador.transform.position.x + LugarSeguroDeGeneracion)  {  int indiceBloque = Random.Range(0, BloquePreFab.Length - 1);  if (PunteroDeJuego < 0)  {  indiceBloque = 8;  }  GameObject ObjetoBloque = Instantiate(BloquePreFab[indiceBloque]);  ObjetoBloque.transform.SetParent(this.transform);  Bloque bloque = ObjetoBloque.GetComponent<Bloque>();  ObjetoBloque.transform.position = new Vector2(  PunteroDeJuego + bloque.tamaño / 2,  0  );  PunteroDeJuego += bloque.tamaño;  }  }  } |

|  |
| --- |
| Jugador  using System.Collections;  using System.Collections.Generic;  using UnityEngine;  public class Jugador : MonoBehaviour {  public int FuerzaDeSalto;  public int VelocidadDeMov;  bool EnElPiso = false;  // Start is called before the first frame update  void Start() {    }  // Update is called once per frame  void Update() {  if (Input.GetKeyDown("space") && EnElPiso)  {  this.GetComponent<Rigidbody2D>().AddForce(new Vector2(0, FuerzaDeSalto));  }  this.GetComponent<Rigidbody2D>().velocity= new Vector2(VelocidadDeMov,  this.GetComponent<Rigidbody2D>().velocity.y  );  }  private void OnTriggerEnter2D(Collider2D c1)  {  EnElPiso = true;  if(c1.tag == "Obstaculo")  {  GameObject.Destroy(this.gameObject);  }  }  private void OnTriggerExit2D(Collider2D collision)  {  EnElPiso = false;  }  } |

**2.Archivo comprimido del proyecto**

Nota: La entrega total, con movimiento del personaje