



Práctica 8

Fecha de Entrega: 13 de enero de 2023

Trejo Nava Ana Maritz

31515077



Universidad Nacional Autónoma de México

Facultad de ingeniería

División de ingeniería eléctrica

Temas Selectos de ingeniería en Computación III

Alumno: Trejo Nava Ana Maritza

No. Cuenta: 315150773

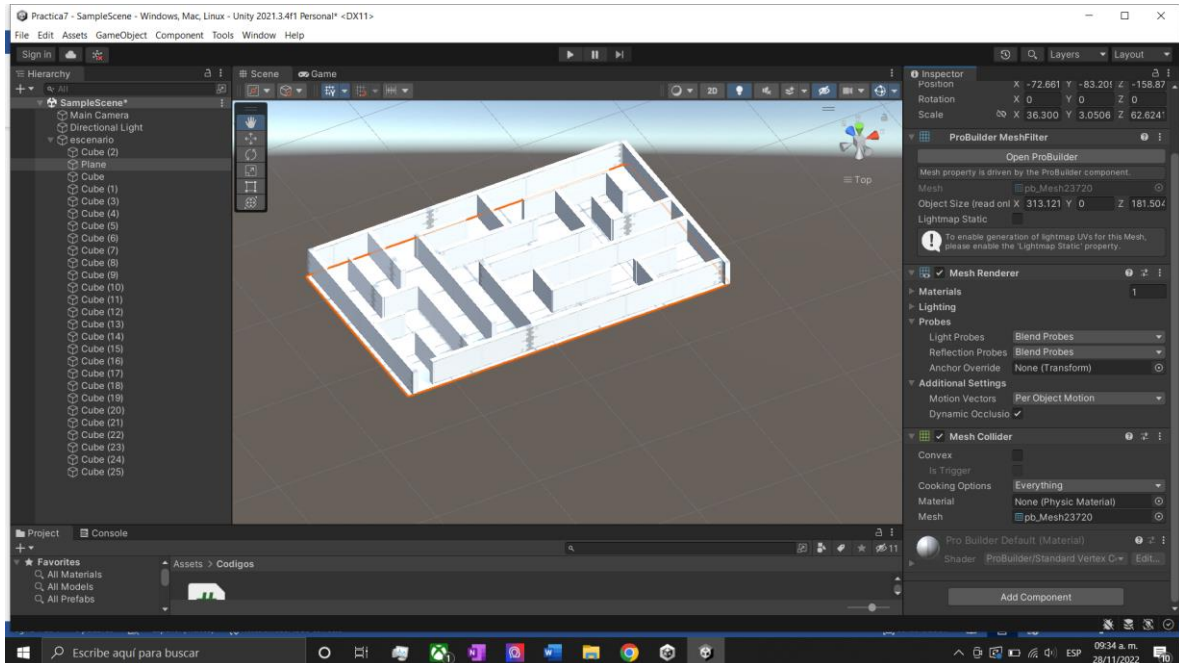
Grupo:01

Practica 8

Profesor: Ing. Arturo Pérez de la Cruz

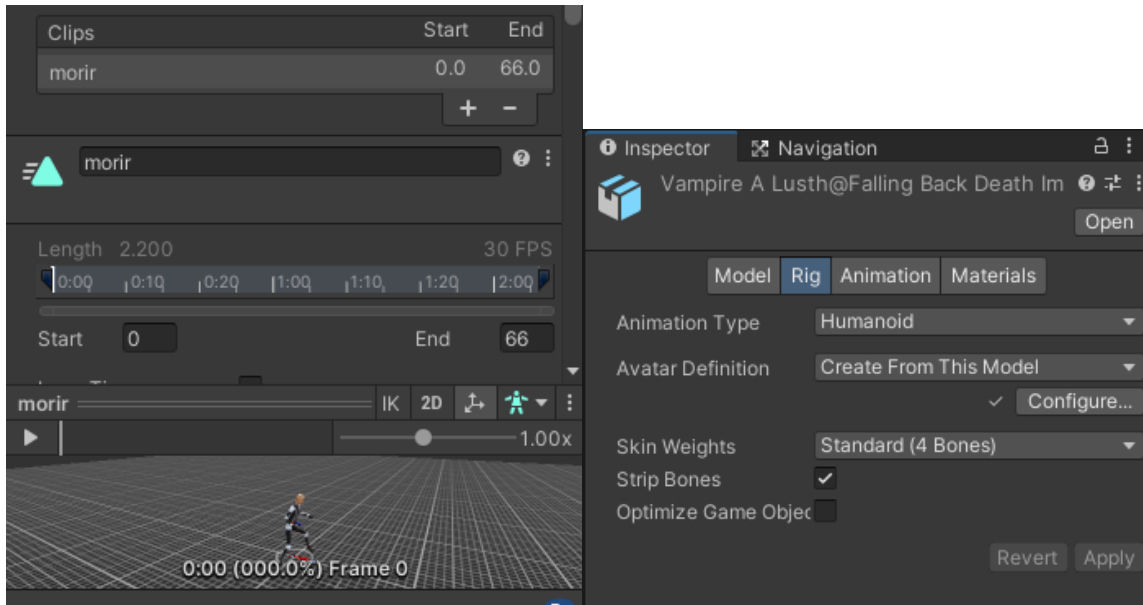
Fecha: 13/01/2023

Para realizar esta práctica configuramos nuestro proyecto como lo hemos hecho en prácticas anteriores, posteriormente, se importó el Package de Probuilder con el cual se le fue creando un laberinto como se muestra en la imagen siguiente.

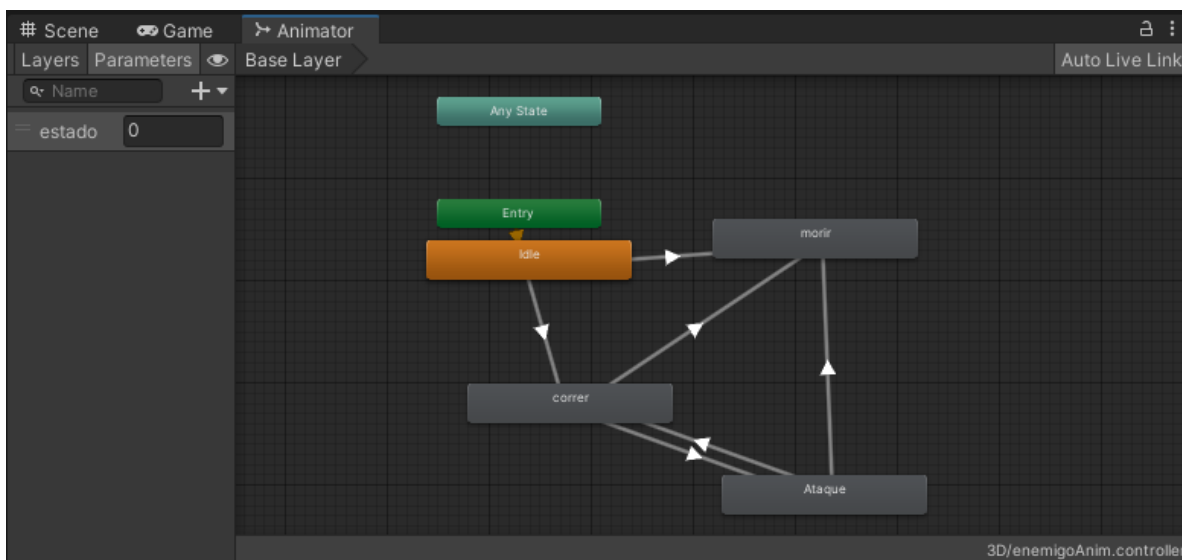


Después configuramos la cámara, de tal forma que parezca que se está en primera persona, por lo que simplemente se ajustó la cámara y le agregamos código para poder desplazarla con las teclas.

Posteriormente entramos a una página llamada Mixamo, en la cual podemos encontrar diferentes modelos 3D de personajes los cuales ya están animados o bien se le puede ir agregando nuevas animaciones y después importarlos en FBX para Unity. Y en Unity se tiene que comprobar que estén bien, sino se tiene que ajustar manualmente. Y se tiene que especificar en la sección de Rig que es tipo humanoide para que lo pueda reconocer Unity. Ya en la sección de animación especificamos si queremos que la animación sea repetitiva o solo una vez se reproduzca, la duración el nombre que se le asignara, de igual forma no se puede dejar pasar especificar la carpeta para la extracción de materiales y texturas.

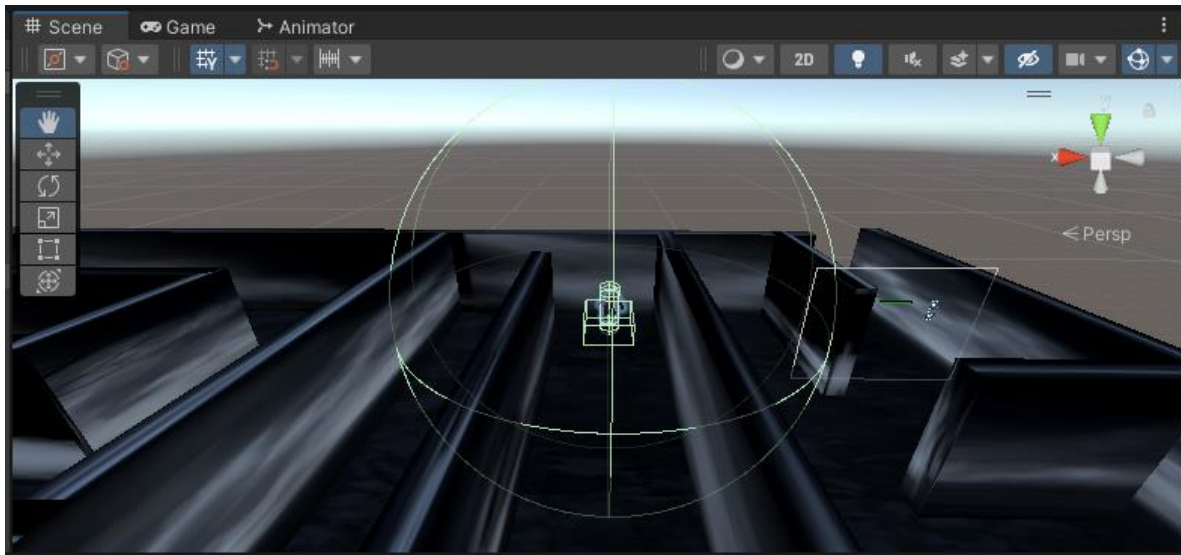


Ya que contamos con las animaciones necesarias es debemos abrir una nueva ventana llamada Animator, en la cual es una maquina de estados que nos permite configurar o bien diseñar el comportamiento que tendrá nuestro personaje, de igual forma podemos especificar parámetros en los cuales va a recibir un tipo de dato, el cual nosotros vamos a programar y especificar en cada transición para que cuando escribamos el código podamos realizar las diferentes animaciones.



Sin embargo, ya que teníamos nuestra máquina de estados tuvimos muchas complicaciones ya que la animación no se ejecutaba cuando el personaje estaba

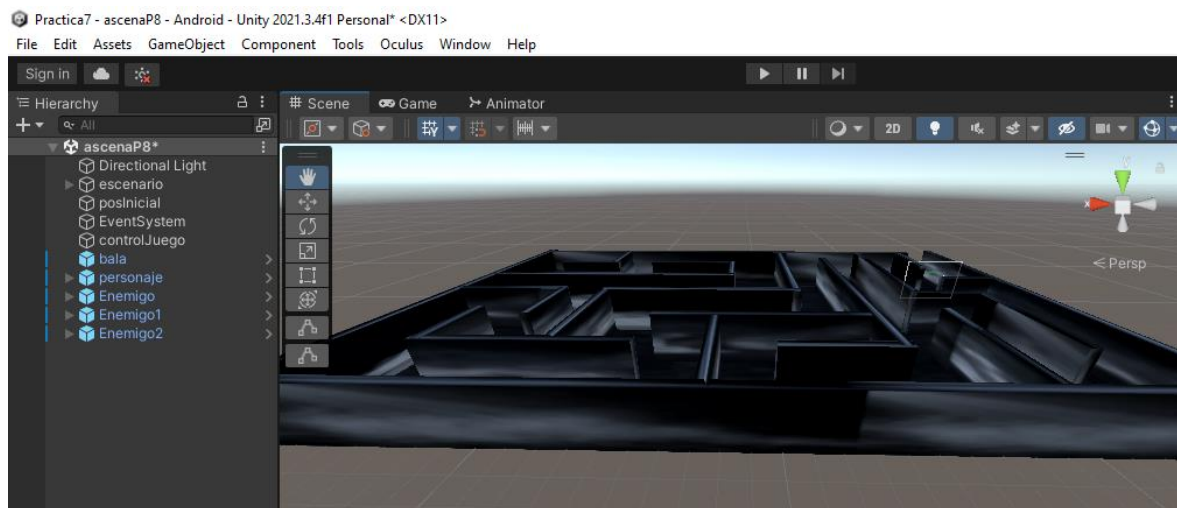
cerca por lo que fue tardamos una clase completa tratando de detectar el error, pero aun así no se logró, por lo que después se nos entrego un nuevo personaje ya con la configuración correcta para evitar no tener problemas con la animación y principalmente con los colliders que se colocaron y que no reconocía.



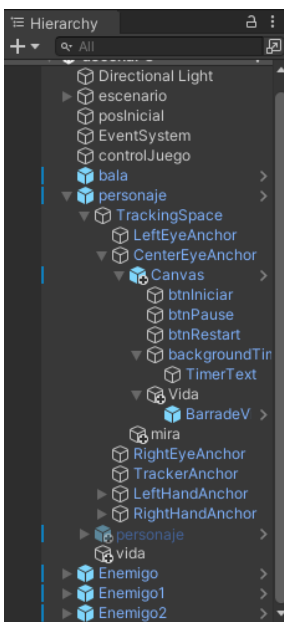
De igual forma se agregaron elementos visuales como el entorno se texturizo para que pareciera un laberinto un tanto tenebroso o algo así. Por otro lado, se agregó un menú el cual permite reiniciar el juego, pausarlo y continuar, además de mostrar un cronometro de 5 minutos una vez que termina el juego también, además de una barra de vida la cual se va reduciendo una vez que un enemigo te toca, por lo que será necesario terminar el circuito antes de que se termine la vida sino el juego se reiniciara.



Finalmente se agregaron bala para herir al enemigo y evitar ser lastimado. Para ello agregamos una esfera, con un collider y un rigidbody en cual mediante código después de 5 segundos se destruye, pero si toca a un enemigo lo mata o bien activa la animación para que muera, y después de cierto tiempo estando en el suelo el enemigo ahora si se destruye, mediante la función de Invoke hice que se destruyera después de 20 segundos. Finalmente se agregaron mas enemigos y listo quedo todo para ser configurado para el Oculus Quest 2.



Para configurar nuestro proyecto primero se accedió a la documentación



Lo primero que se pide es configurar el proyecto, pero eso ya lo realizamos desde un principio. De igual forma pide agregar al proyecto el paquete `OculusIntegration.unitypackage` el cual nos permitirá trabajar con las características del oculus. Ya que se descargó evidentemente se importo a Unity, se realizó así ya que en el package manager no lo encontré. De igual se continuo configuran en `ProjectSettings` y según se especificaba en la documentación. Y finalmente se borro la cámara que se tenía por default, agregando la que se proporciona en el paquete, la `OVRCameraRig`, respetando la jerarquía, a esta cámara se le asigno el canvas creado previamente y se configuro como `World Space`.

Finalmente se conectaron los cascos y se comprobó en Build Settings que se reconocieran, pero para esto también era necesario aceptar todos los permisos que se especifican cuando te colocas los lentes, al principio no podía por lo mismo que no los había configurado, por lo que tarde bastante en eso. Pero ya con esto se pudo continuar con el reconocimiento y la instalación del laberinto ya en los lentes. Por lo que esta practica se pudo completar de forma satisfactoria.

Links de la Documentación:

<https://developer.oculus.com/documentation/unity/book-unity-gsg/>

<https://developer.oculus.com/documentation/native/android/ts-adb/>

Enlace al video: <https://youtu.be/KzX4Ql5kGu4>