



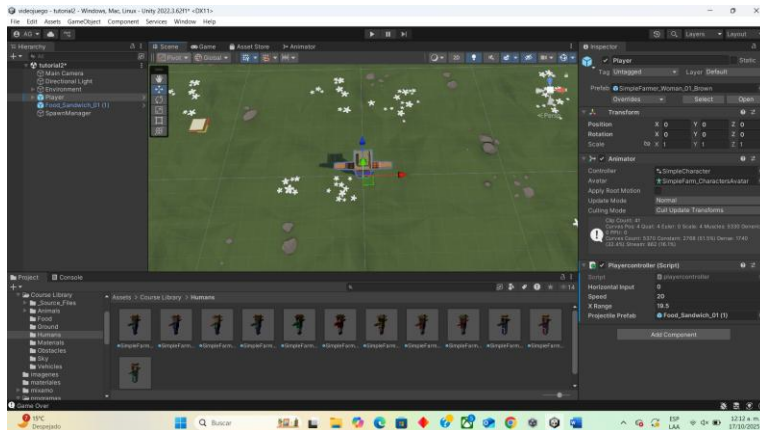
Universidad Tecnológica del Norte de Guanajuato
Entornos virtuales y Negocios digitales

TUTORIAL 2
Alejandro Hernández González- 1223100434
GIEVN0371
Programación de Videojuegos
17 de Octubre de 2025

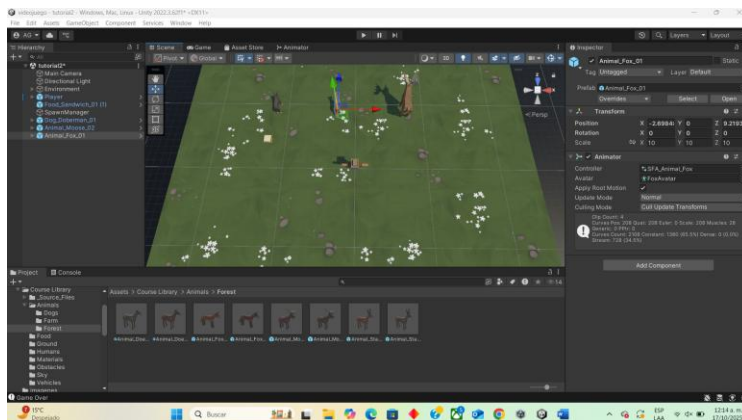
Tutorial 2

Para este segundo tutorial el objetivo es crear un juego que trata de controlar un granjero y lanzar comida a animales que irán spawnando a lo largo del mapa

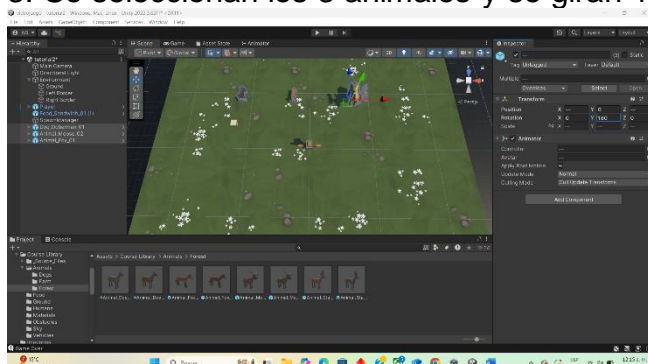
1. Primero se descarga el proyecto de la página de unity learn, y se coloca sobre el plano que está por defecto un personaje que será el que vamos a controlar, y le cambiamos el nombre a player



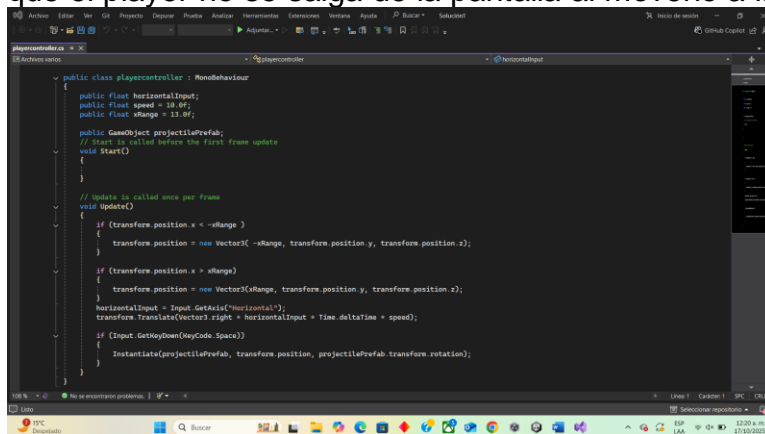
2. Se coloca sobre el plano comida de la carpeta Food y 3 animales de la carpeta Animals



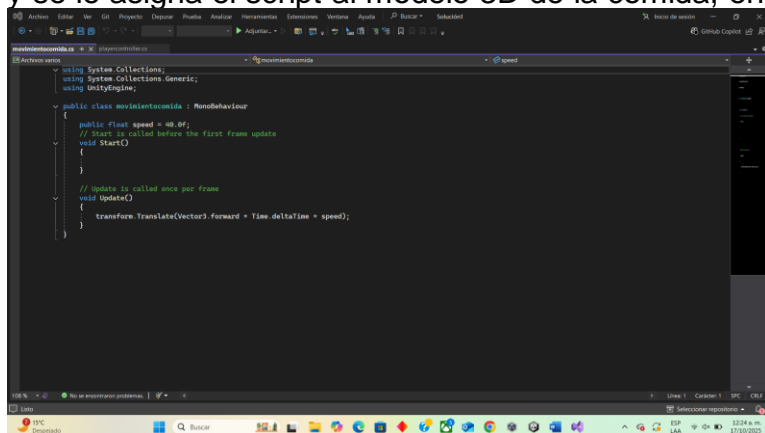
3. Se seleccionan los 3 animales y se giran 180 grados en el eje y



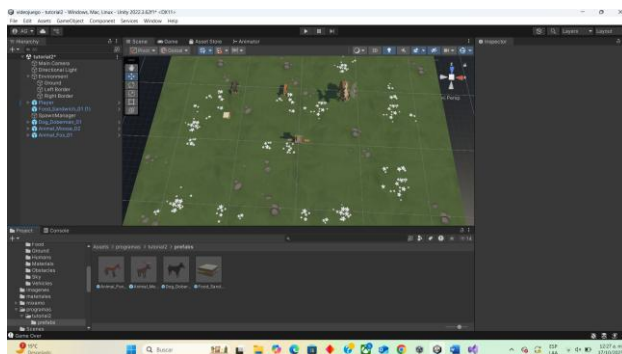
4. Se crea un nuevo script, con el fin de darle al player un movimiento horizontal, de izquierda a derecha, y usando un método if junto a la variable xRange, para asegurarnos que el player no se salga de la pantalla al moverlo a la izquierda o derecha en el eje X



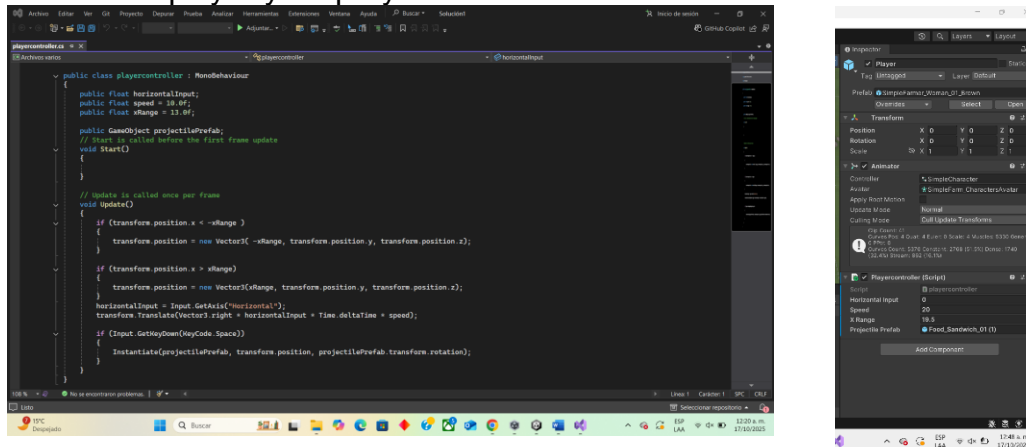
5. Se crea un nuevo script llamado movimientoComida, con el fin de implementar la función de que la comida salga disparada en el eje Z, asignándosele una velocidad de 40, y se le asigna el script al modelo 3D de la comida, en este caso al sandwich



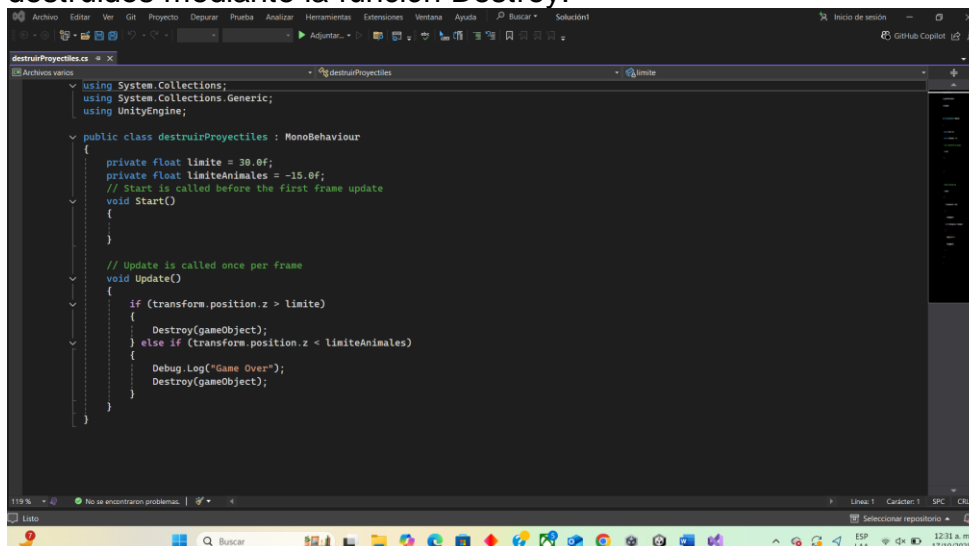
6. Se crean prefabs, los cuales sirven como clones de nuestros modelos, para así poder utilizarlos en el momento que queramos dependiendo de las funciones que queramos integrar en nuestro juego, para crearlo se arrastra del hierarchy nuestro modelo a una carpeta que creamos llamada prefabs, se arrastraron los 3 animales y el sandwich.



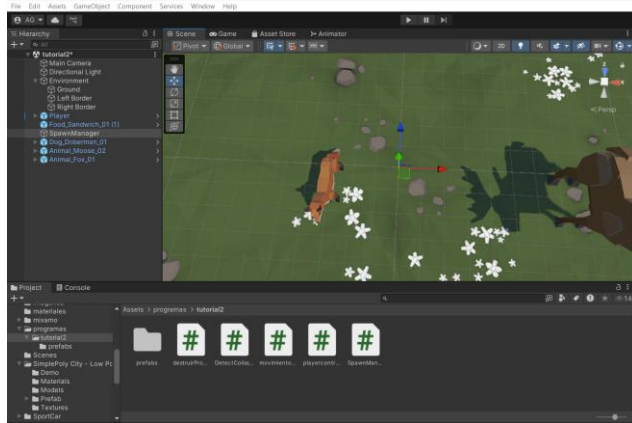
7. A nuestro script playerController le agregamos la función de que al presionar la barra espaciadora lance el proyectil que en este caso es el sándwich, nos metemos al inspector de nuestro player y en proyectilPrefab arrastramos el Prefab del sandwich



8. Creamos un nuevo script, llamado destruirProyectiles y se lo asignamos a nuestro proyectil que en este caso es el sándwich, se declaran las variables tipo float limite, que será para controlar hasta donde llega el sándwich antes de ser destruido, y la variable limiteAnimales, que controla lo mismo que el sándwich, ambas en el eje Z, y el método if para definir que cuando los animales o los proyectiles salgan de cuadro, estos sean destruidos mediante la función Destroy.

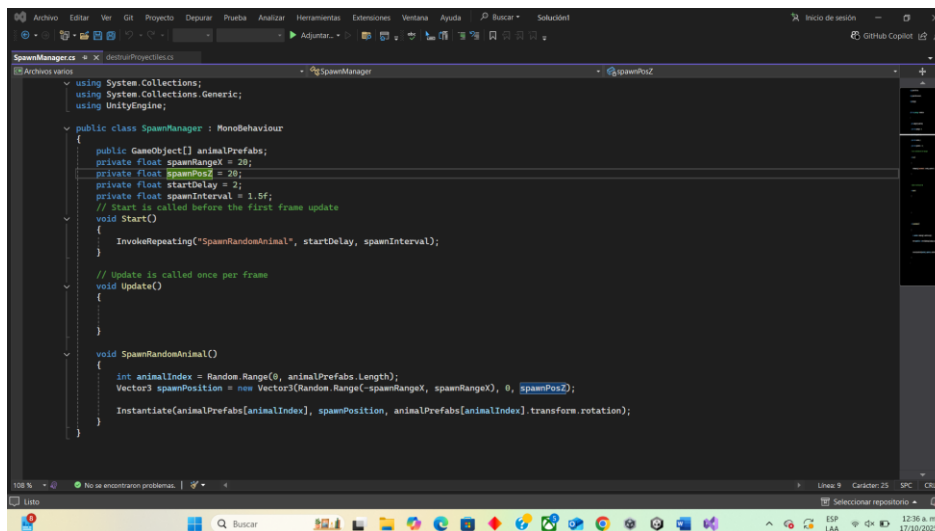


9. Creamos en hierarchy un empty object, y lo nombramos spawn Manager

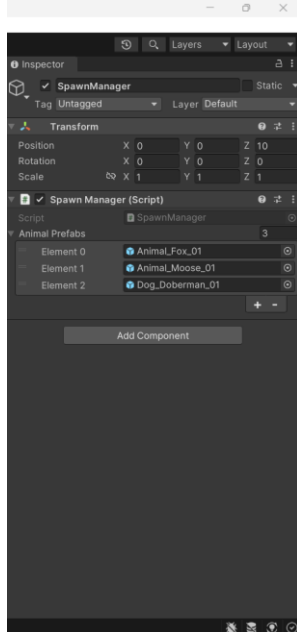


10. Creamos un nuevo script llamado SpawnManager, el cual se asignará a nuestro objeto vacío con el mismo nombre, declaramos las variables spawnRangeX y spawnPosZ para definir el límite de spawn de los animales en los ejes X,Z y las variables startDelay para que los animales comiencen a spawnear 2 segundos después de iniciar el juego, mientras que la variable spawnInterval define que los animales se irán spawnando cada 1.5 segundos, y se mandan llamar los prefabs de los animales, a los cuales se les agrega el método random.Range para que los animales spawnen en posiciones al azar dentro del límite permitido que es 20.

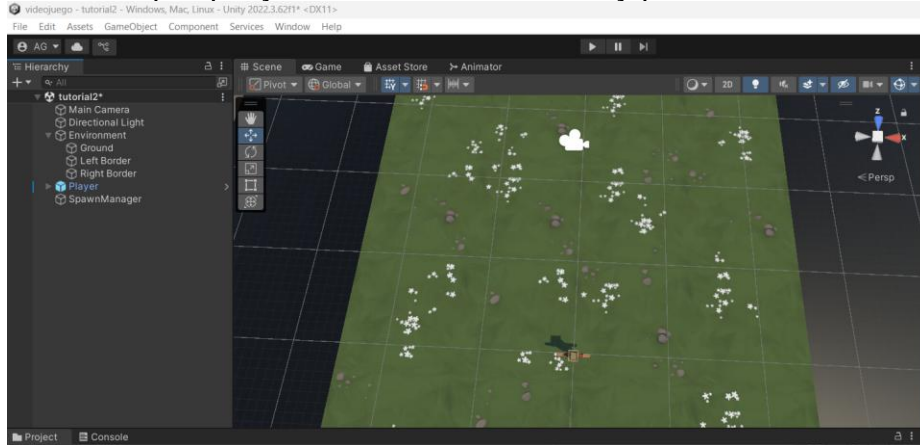
La siguiente línea de código se utiliza para mandar llamar uno de los 3 animales al azar
`int animalIndex = Random.Range(0, animalPrefabs.Length);`



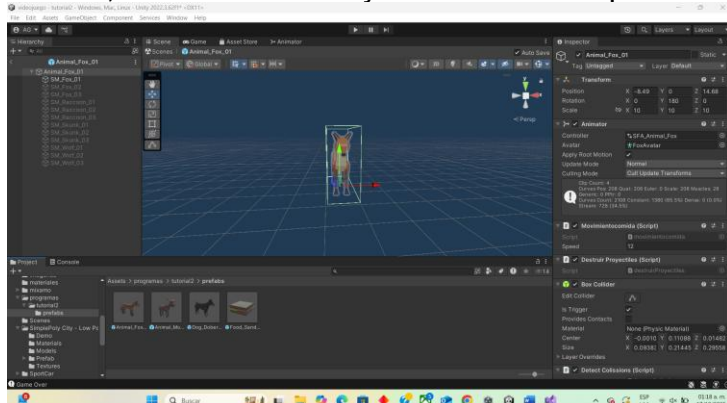
11. Seleccionamos nuestro empty llamado Spawn Manager, le agregamos el script y en Animal Prefab arrastramos cada Prefab de nuestros 3 animales

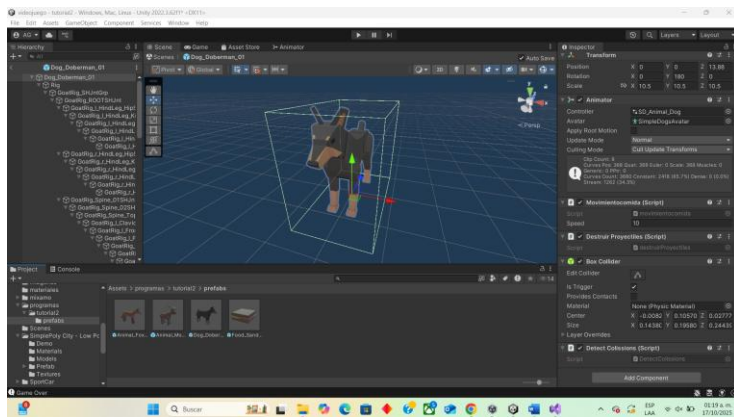
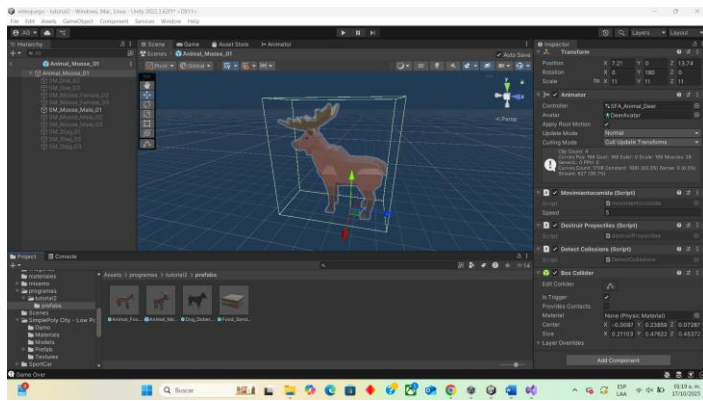


12. Eliminamos del hierarchy los 3 animales y el sándwich, ya que no los necesitamos en la escena porque ya tenemos sus Prefabs y podemos llamarlos cuando queramos

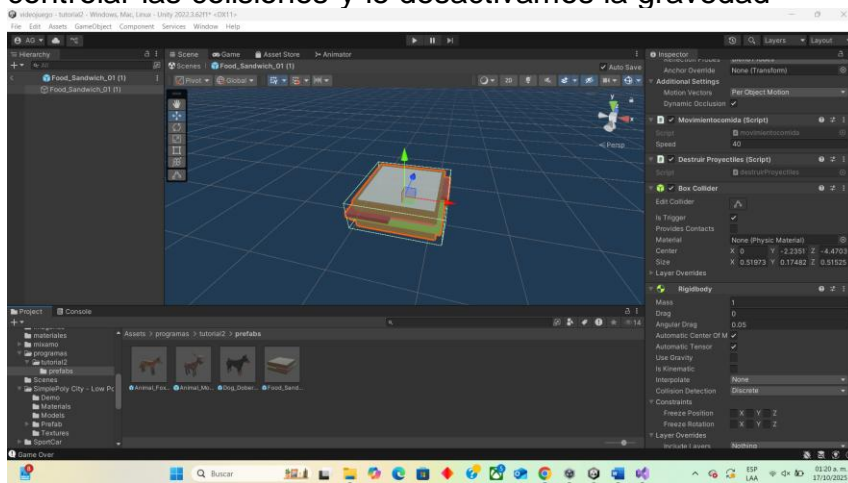


13. Hacemos doble click sobre los Prefabs (animales y proyectil) agregamos un box collider, lo acomodamos y marcamos con palomita "IsTrigger"

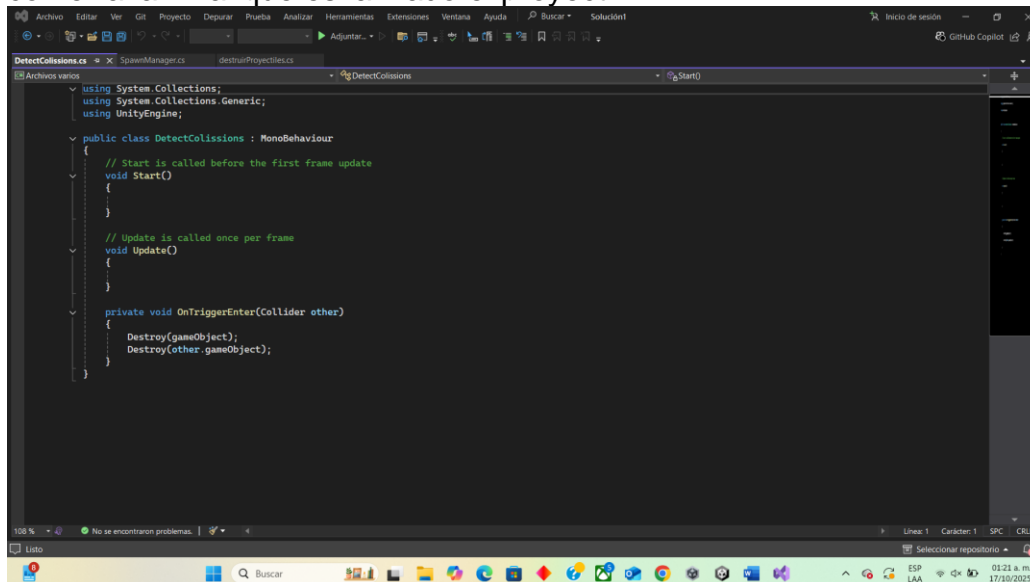




A nuestro proyectil, además de agregarle el box collider se le agrega un rigidbody, para controlar las colisiones y le desactivamos la gravedad



14. Creamos un nuevo script llamado DetectCollissions, para implementar lo que queremos que pase cuando nuestro proyectil colisione con los animales, en este caso queremos que al colisionar con otro objeto sea eliminado, lo que crea el efecto de darle de comer al animal que es lanzado el proyectil



15. Finalmente al script llamado destruiProyectil se le agrega en el último if, la función que hace al detectar si algún animal sobrepasa el límite de la pantalla, el cuál esta en -15, es decir, si el animal logra avanzar hasta -16 aparte de eliminarse saldrá un mensaje en consola el cual dirá “Game over”, ya que no se cumplió el objetivo de darle de comer.

