



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO



MANUAL TÉCNICO.

Universidad Nacional Autónoma de México.

Octavio López Rodríguez.

Semestre: 2022-2.

Materia: Temas Selectos de Ingeniería en Computación III. Realidad Virtual y Realidad Aumentada.

Fecha de entrega:

Planeación.

Para la realización de este proyecto se tuvo que hacer una planeación de actividades, las cuales incluían la búsqueda de los modelos a utilizar, buscar el material audiovisual a usar, ver lo que necesitaba para poder llevar a cabo una animación, texturizar los modelos, creación del proyecto en Unity, importar el contenido al proyecto de Unity, por el contenido me refiero a los modelos, imágenes y videos. Hacer los escenarios y las animaciones, hacer pruebas y correcciones si son necesarias y, por último, hacer la documentación del proyecto, que esta incluye dos manuales, el técnico y el de usuario.

				Semana 1.			Semana 2					
Actividad	Responsable	Inicio	Fin	V	S	D	L	M	MI	J	V	S
Crear repositorio de GitHub.	Octavio López Rodríguez.	01/04/2022	02/04/2022									
Busqueda de modelos.	Octavio López Rodríguez.	03/04/2022	05/04/2022									
Busqueda de material audiovisual.	Octavio López Rodríguez.	06/04/2022	09/04/2022									
Ver cuestiones de animaciones.	Octavio López Rodríguez.	11/04/2022	20/04/2022									
Texturizar modelos.	Octavio López Rodríguez.	21/04/2022	25/04/2022									
Crear poryecto de Unity.	Octavio López Rodríguez.	01/04/2022	02/04/2022									
Importar modelos en Unity	Octavio López Rodríguez.	26/04/2022	27/04/2022									
Importar contenido del escenario a Unity	Octavio López Rodríguez.	26/04/2022	27/04/2022									
Hacer los escenarios y animaciones.	Octavio López Rodríguez.	28/04/2022	09/05/2022									
Pruebas y correcciones del proyecto.	Octavio López Rodríguez.	10/05/2022	18/05/2022									
Documentación.	Octavio López Rodríguez.	19/05/2022	23/05/2022									

Imagen del diagrama de Gantt.

Esta planeación estaba contemplada para un tiempo de poco más de dos meses, para ser exactos del 16 de marzo del 2022 al 25 de mayo del 2022, que son casi 9 semanas. Para ver completo la planeación se puede descargar o acceder al archivo desde el siguiente repositorio de GitHub, en la carpeta de Entrega Final > Documentación > Diagrama de Gantt.

Enlace: <https://github.com/octaviolr/ProyectoFinalTS>

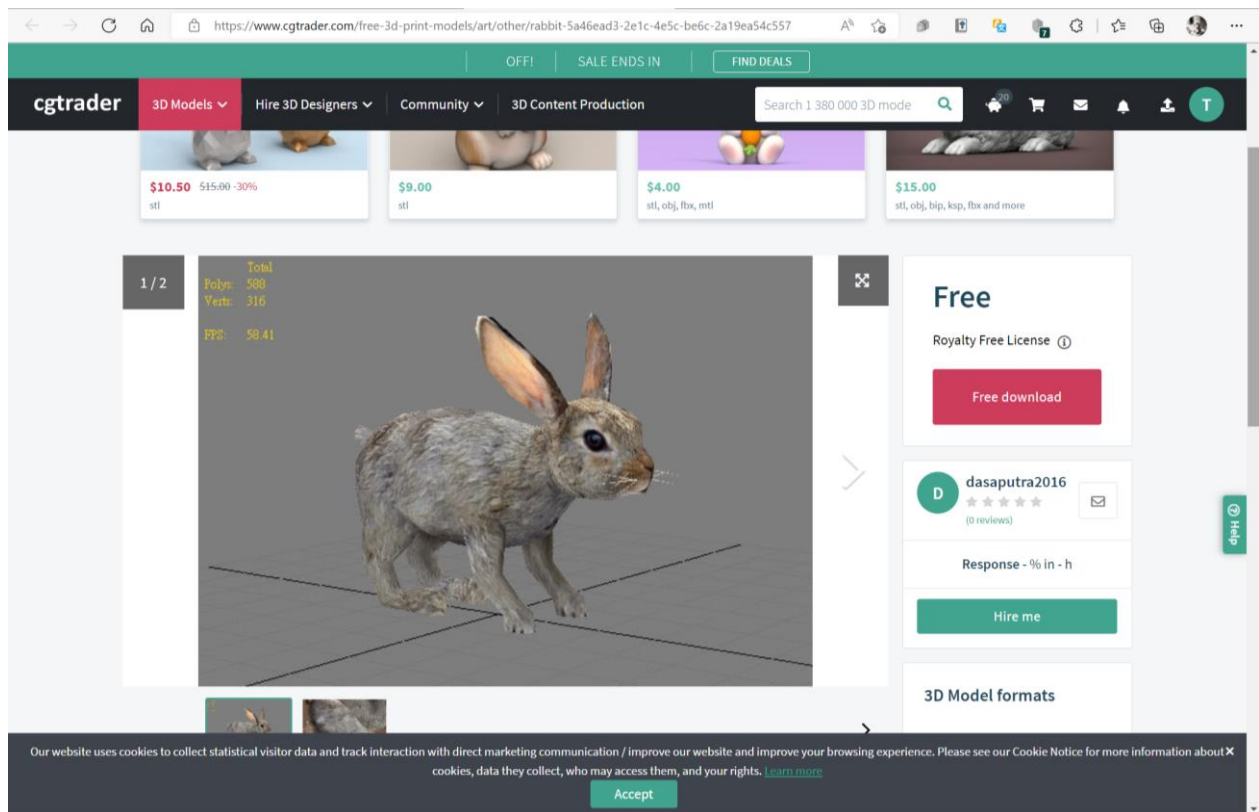
Realización del proyecto.

Para realizar este proyecto se ocuparon, principalmente, dos softwares. El primero de ellos fue Unity que es en el que se trabajó la mayor parte del tiempo, y el

Fecha de entrega:

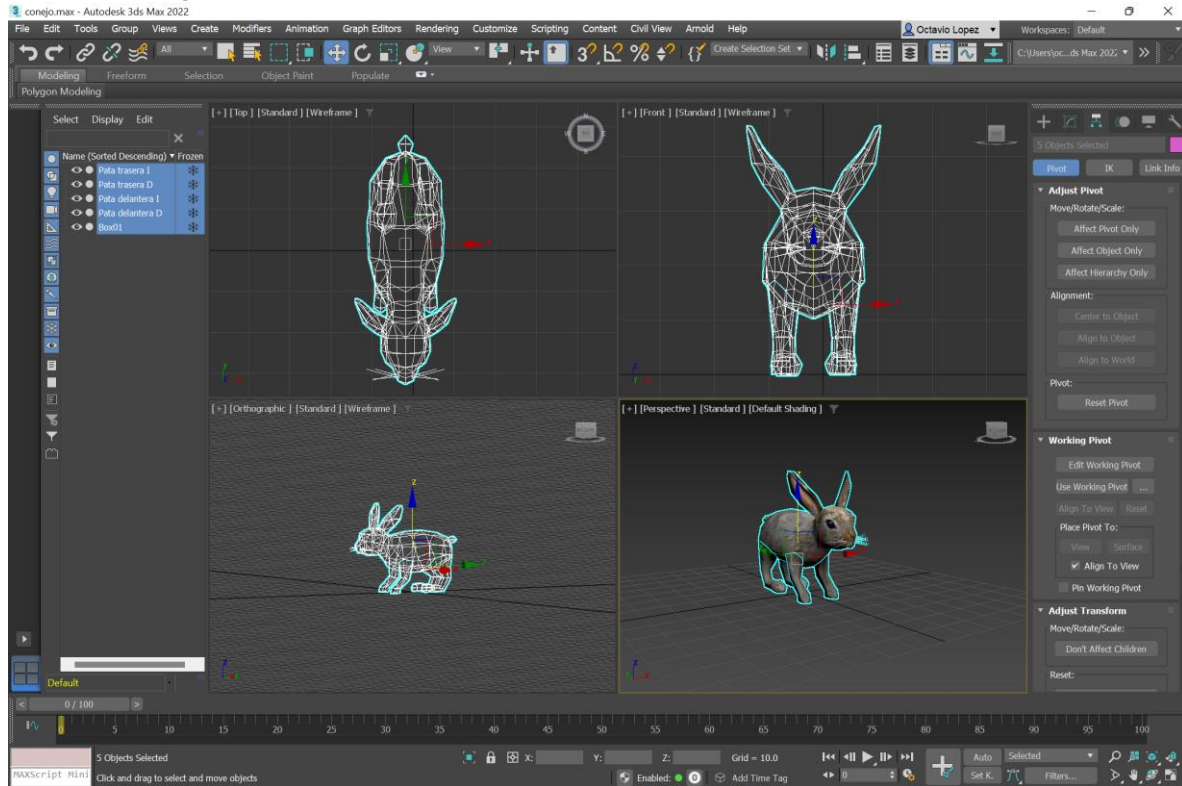
segundo, 3Ds Max que es un software que permite la manipulación y/o creación de modelos en 2D o 3D. Para el caso de Unity, tenemos que es un software gratis, pero para el caso de 3Ds Max, si es un software de paga, aunque ahorita se está usando una versión para estudiantes.

Bien, lo primero que necesitamos fueron los modelos, esos se descargaron de una página llamada cgtrader, en dicha pagina se puede encontrar una gran variedad de modelos de casi cualquier tipo, y hay tanto modelos gratis como de paga.



Ya que tenemos seleccionados los modelos que vamos a ocupar, es cuestión de descárgalos y después importarlos a 3Ds Max:

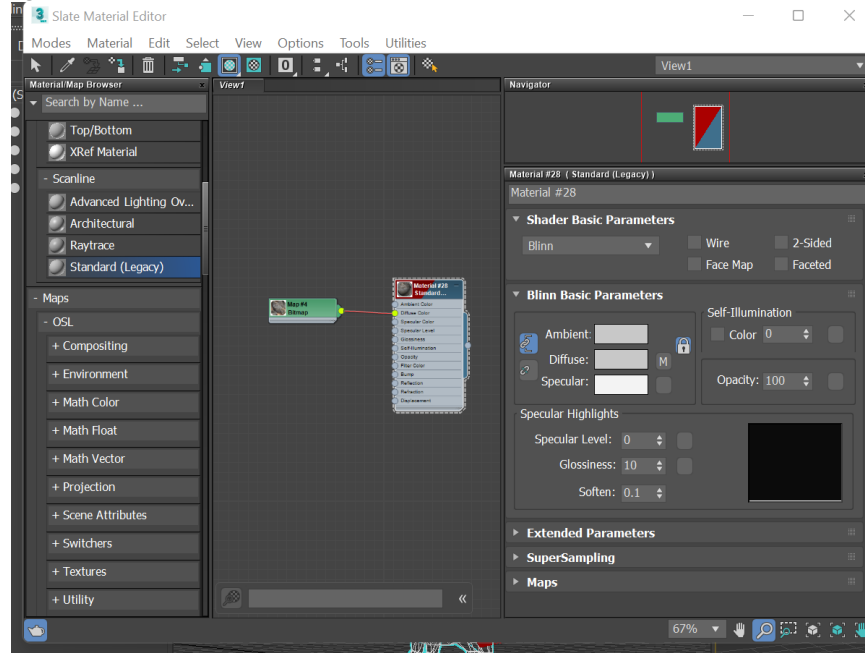
Fecha de entrega:



Ya que lo abrimos en 3Ds Max, se verá como en la imagen de arriba, aquí lo mas importante es verificar que el pivote, el centro del modelo o el que nos permite hacer las transformaciones de rotación, escala y traslación, se encuentre en el modelo, muchas veces se encuentra fuera del modelo y al intentar aplicar una transformación se vuelve complicado. Para hacer eso, del lado derecho nos aparece una opción que es Affect Pivot Only, esta opción nos permite moverlo. Ya que nos aseguramos de que el pivote está dentro del modelo, queda hacer la textura del modelo, la textura no es más que una imagen que se le va a aplicar. Al momento de descargar algunos modelos pueden incluir su textura, otros no y se tendrá que buscar una a tu gusto y/o necesidad, por suerte los modelos empleados en este proyecto si incluían su textura, por lo tanto, solo fue cuestión de agregársela.

Para agregar una textura, en 3Ds Max, podemos oprimir la tecla M y se nos abrirá una nueva ventana, en dicha ventana vamos a agregar un nuevo material de tipo Standard y posteriormente agregaremos un BitMap, y lo conectamos a la componente difusa del material.

Fecha de entrega:

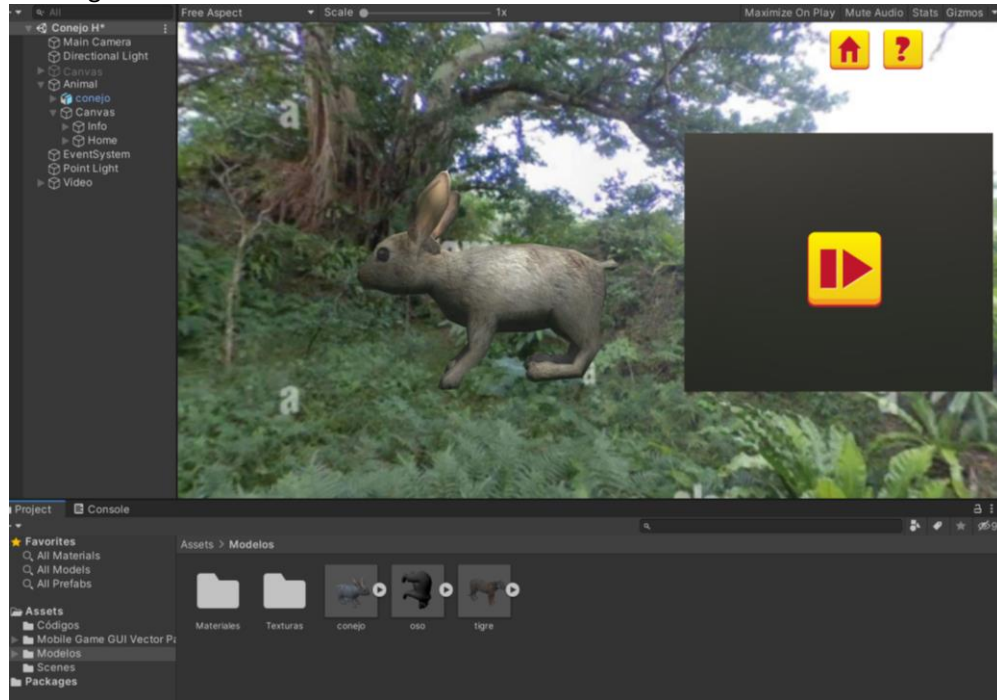


Lo siguiente es dar click en el cuadro del BitMap, y asegurarnos de que nuestra textura se encuentre en la carpeta correspondiente que tiene 3Ds Max, y simplemente es seleccionarla y aplicársela al modelo.

De esta forma ya tendríamos nuestro modelo listo para trasladarlo a nuestro ambiente en Unity. Simplemente lo exportamos en un formato de fbx.

Dentro de nuestro proyecto de Unity, sería cuestión de agregar los modelos, se recomienda tener el proyecto organizado por carpeta, para de esta forma saber dónde se encuentra cada cosa y tener un orden. Se creo una carpeta llamada “Modelos”, en la que se importaron los tres modelos que descargamos, dentro de esa carpeta se crearon otras dos, una para las texturas y otra para los materiales. En esas dos carpetas se guardarán los materiales y las texturas de los modelos, ambos no se tienen que crear ni nada, al momento de importarlos a Unity simplemente se extraen esas carpetas y cuando se trasladen al escenario ya aparecerán con su textura.

Fecha de entrega:



También se creó una carpeta para guardar los códigos necesarios dentro del proyecto, que principalmente fue uno. Es un código muy sencillo que nos permite cambiar de escenas cuando hacemos click en el botón, llamado “LogicaCambioEscena”:

```

7 public class LogicaCambioEscena : MonoBehaviour
8 {
9     public bool cambio;
10    public int indice;
11    // Start is called before the first frame update
12    @ Message de Unity | 0 referencias
13    void Start()
14    {
15    }
16
17    // Update is called once per frame
18    @ Message de Unity | 0 referencias
19    void Update()
20    {
21        if (Input.GetKeyDown(KeyCode.Space))
22        {
23            CambiarEscena(indice);
24        }
25
26        if(cambio)
27        {
28            CambiarEscena(indice);
29        }
30
31    2 referencias
32    public void CambiarEscena(int indice)
33    {
34        SceneManager.LoadScene(indice);
35    }
36
37    0 referencias
38    public void Salir()
39    {
40        Application.Quit();
41    }

```

Fecha de entrega:

Con el método `CambiarEscena`, que recibe un número entero, llamado índice, que ese número Unity se lo asigna a cada escena y ese sería el valor que le tendríamos que pasar a ese método para que nos cambie a la escena que nosotros queremos.

También se descargaron un par de Assets, uno fue `Movil Game GUI`, que este nos ayudo a darle una nueva imagen a los botones. Y el segundo fue `TextMesh Pro`, este nis ayuda a darle un formato personalizado a los textos que aparecen. Al tener muchos archivos, no los puedo subir al repositorio de GitHub, por eso recomiendo descargarlos de la tienda de Assets que tiene Unity, o bien, accediendo al siguiente enlace:

<https://assetstore.unity.com/packages/2d/gui/icons/mobile-game-gui-vector-pack-53679#description>

En el caso de `TextMesh Pro` ya venía instalado en Unity.

Costos del proyecto.

A diferencia de la propuesta inicial, hay un pequeño cambio en los costos, porque ahora el precio del internet ha subido.

Servicio.	Precio por mes.	Tiempo total del proyecto. (3 meses)
Luz.	\$150.00	\$450.00
Internet.	\$619.00	\$1,857.00
Modelos.	\$0.00	\$0.00
Licencias de Software.	\$0.00	\$0.00
Sueldo.	\$16,000.00	\$48,000.00
Total.	\$16,769.00	\$50,307.00

Servicio.	Precio por mes.	Tiempo total del proyecto. (3 meses)
Luz.	\$150.00	\$450.00
Internet.	\$717.00	\$2,151.00
Modelos.	\$940.00	\$2,820.00
Licencias de Software.	\$3,305.00	\$9,915.00
Sueldo.	\$16,000.00	\$48,000.00
Total.	\$21,014.00	\$61,162.00

Como se puede ver en la imagen de arriba, antes teníamos un costo de internet de \$619.00, ahora el precio ha subido a \$717.00, por lo tanto, la nueva tabla de costos queda de la siguiente manera:

Fecha de entrega:

Servicio.	Precio por mes.	Tiempo total del proyecto. (3 meses)
Luz.	\$150.00	\$450.00
Internet.	\$717.00	\$2,151.00
Modelos.	\$0.00	\$0.00
Licencias de Software.	\$0.00	\$0.00
Sueldo.	\$16,000.00	\$48,000.00
Total.	\$16,867.00	\$50,601.00
Servicio.	Precio por mes.	Tiempo total del proyecto. (3 meses)
Luz.	\$150.00	\$450.00
Internet.	\$717.00	\$2,151.00
Modelos.	\$940.00	\$940.00
Licencias de Software.	\$3,305.00	\$9,915.00
Sueldo.	\$16,000.00	\$48,000.00
Total.	\$21,112.00	\$61,456.00

Hay que recordar que se están haciendo dos presupuestos, uno con los modelos y licencias empleados en el proyecto, y la otra con el costo de modelos que no son gratis y el costo de las licencias para el uso de software.

Precio del producto.

Haciendo las cuentas tenemos que en total se trabajaron 225 horas, 5 horas por día durante nueve semanas, trabajando de lunes a viernes. Y tomando en cuenta que una hora de trabajo mía la estoy cobrando a \$169.17 (ciento sesenta y nueve pesos con diecisiete centavos) nos da un total de $169.17 \times 225 = 38,063.25$ pesos.

Si se deseara vender, el precio sería de 400 pesos por unidad, esto tomando en cuenta los precios de distintas aplicaciones que pude encontrar en el mercado y están rondando entre los 300 y 500 pesos.

Conclusión.

Para concluir este proyecto me gustaría decir que me llevo una buena experiencia, sinceramente no creí que disfrutaría tanto la materia, ya que computación grafica fue un tanto pesada para mí y al ver que estaban relacionadas, realmente tenía expectativas bajas. Pero al final me termino gustando, no al grado de que piense en

Fecha de entrega:

dedicarme a esto en un futuro, pero si me abrió un poco el panorama de lo que se puede hacer con estas tecnologías, ya que últimamente se ha estado hablando mucho de la realidad virtual y de la realidad aumentada, pero en lo personal, no había tenido un acercamiento así de directo con ellas y mucho menos había pensado en intentar desarrollar algo para estas tecnologías.