



UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO



FACULTAD DE INGENIERÍA

INGENIERÍA EN COMPUTACIÓN

Practica 2

Animación por Esqueleto

ALUMNOS:

MURRIETA VILLEGAS ALFONSO

VALDESPINO MENDIETA JOAQUIN

PROFESOR:

Ávila Martell Reynaldo

GRUPO:

1

FECHA DE ENTREGA:

22/03/2021

## Reporte Practica 2

### Desarrollo

1. Escoger un modelo de un personaje y realizar la armadura con huesos para este modelo en blender. (15 pts.)

Para este apartado se eligió un modelo de SketchFab, un Venom LowPoly, mostrado a continuación



(venom)

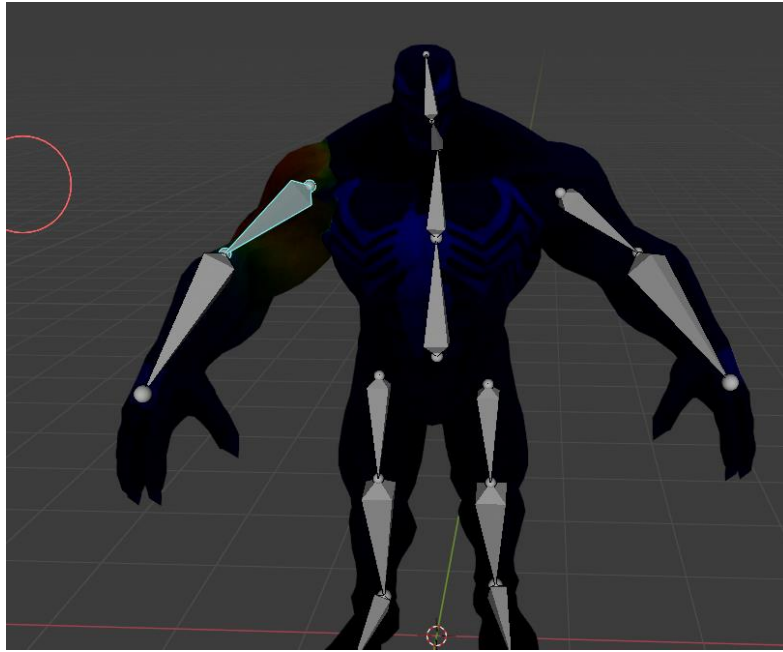
Se editó en Blender y agrego una armadura con huesos adaptados a la morfología de este modelo, a continuación, se muestra el resultado de la creación de los huesos.



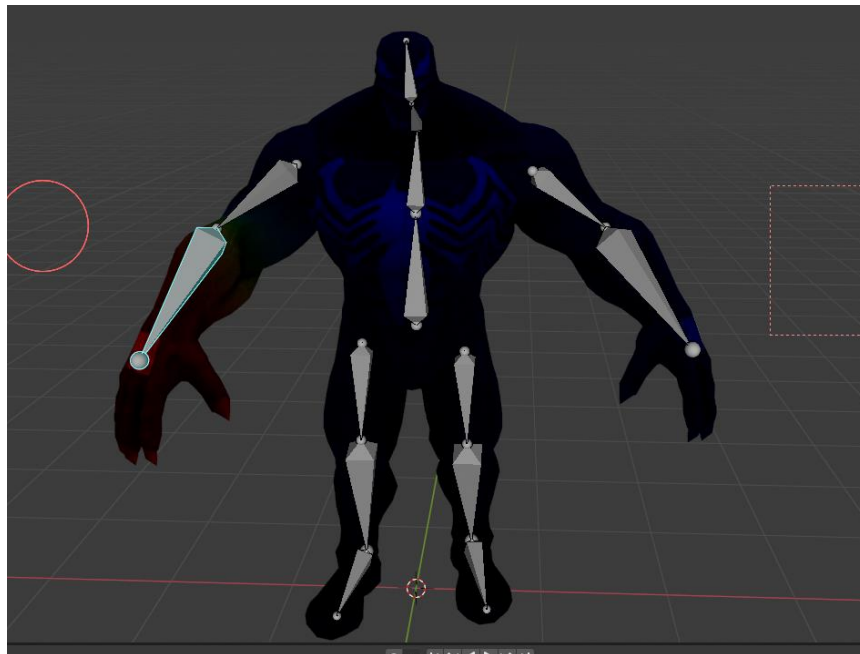
(Armadura de Huesos)

2. Ajustar los pesos para que la maya se deforme de manera correcta. (15 ptos.)

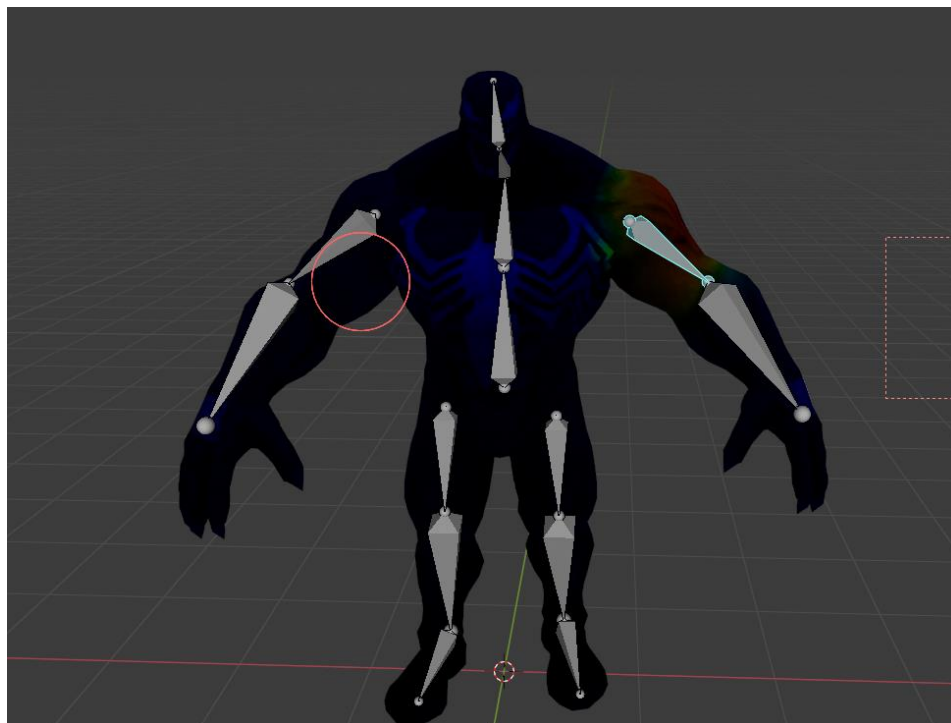
Una vez realizada la armadura, se asignaron los pesos automáticos de los huesos respecto a la maya para poder manipular la pose del modelo, para ello, se tuvieron que editar los límites de deformación de la maya mediante el pincel en modo “Subtract”, a continuación, partes resultado obtenido.



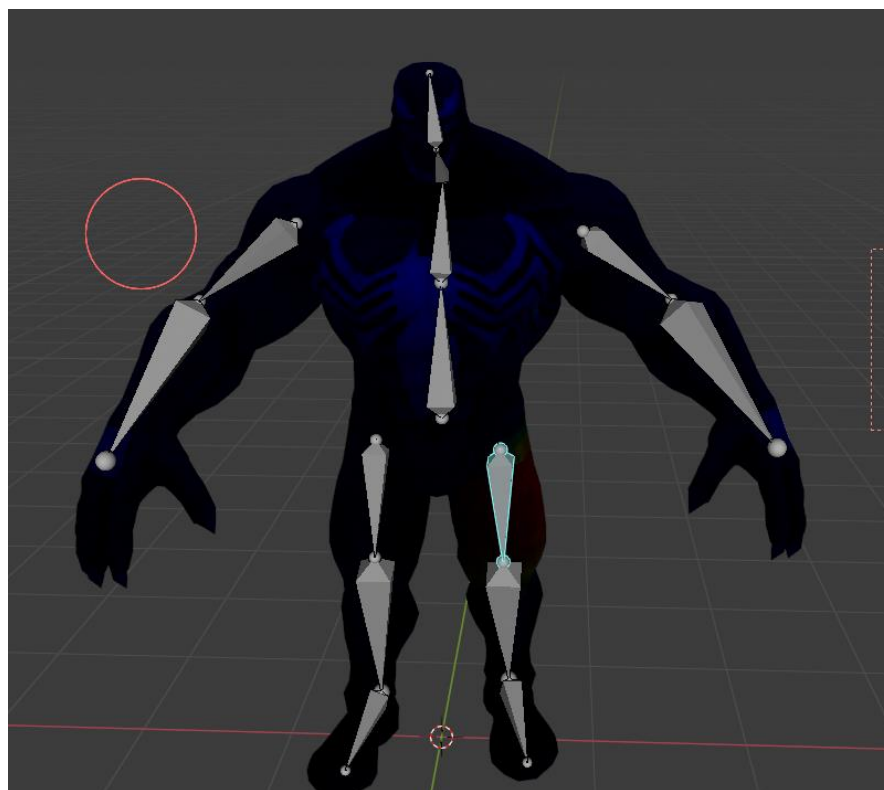
(Antebrazo – hubo mayor deformidad)



(brazo)



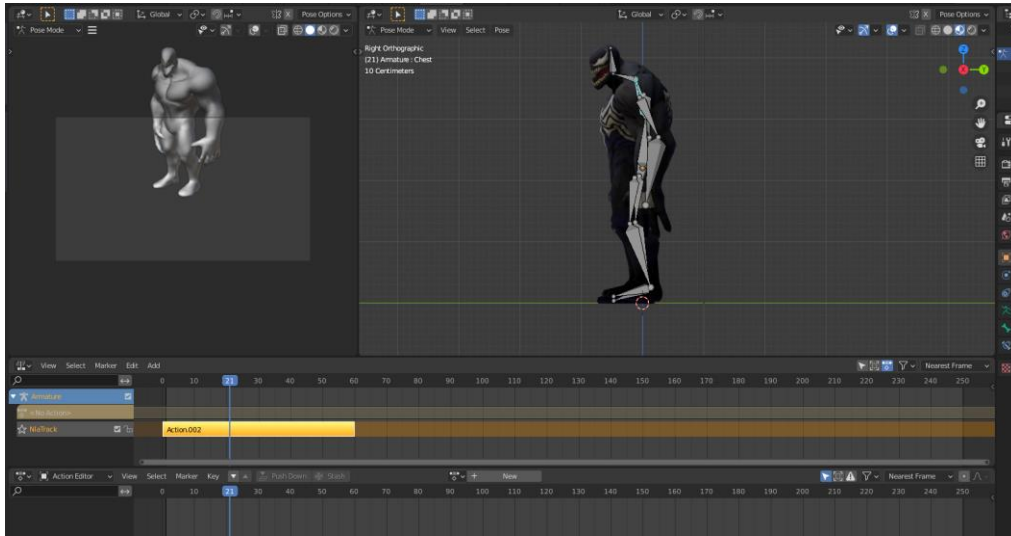
(brazo Izquierdo)



(Pierna- hubo mayor deformidad)

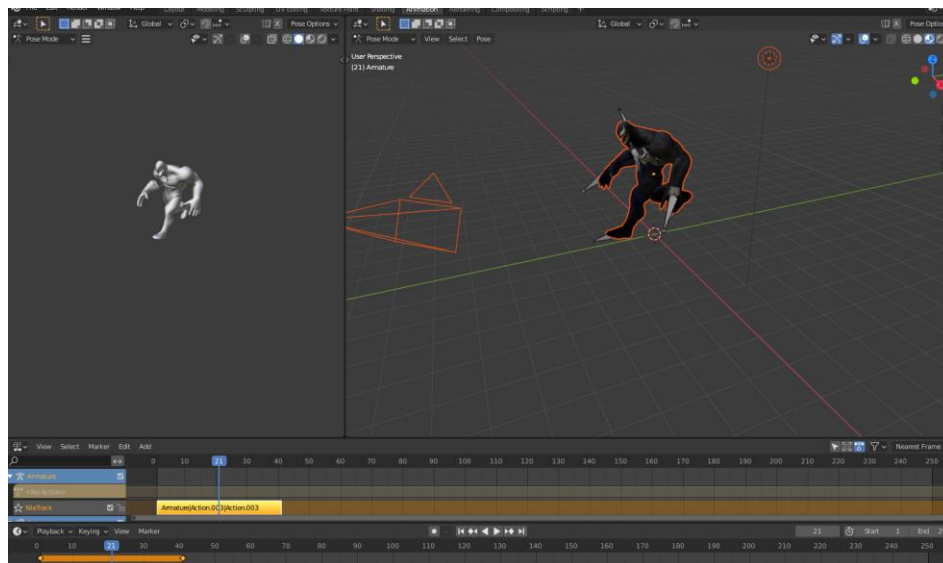
### 3. Realizar la animación del modelo en pose de descanso. (15 pts.)

Con el paso anterior, pudimos obtener una deformación correcta de la malla, para poder realizar las animaciones correspondientes. En este punto se tuvieron que guardar en keyframes las posiciones de los huesos en un determinado instante, con el modo “NLA” o No Linear Animation, se realizó de primera instancia la animación en descanso, que consiste básicamente en una inclinación del torso y una rotación de la cabeza modelo de izquierda a derecha a modo de observación de su alrededor, posterior a ello se exporto el modelo en FBX como venomS.fbx.



### 4. Realizar la animación del modelo Corriendo. (15 pts.)

Para la animación de correr se realizó lo mismo que el paso anterior, la diferencia clara fue que también se tuvieron que mover las piernas y pies para dar el movimiento representativo, además de que se tuvo que duplicar el primer KeyFrame de la animación al último para tener una animación Cíclica. Por último, se exporto como venomC.fbx



## 5. Exportar y cargar las animaciones en OpenGL. (15 ptos.)

Como se observó se crearon dos modelos, cada uno con una animación diferente, para cargarlas en openGL. Se tuvieron que hacer dos importaciones, creando dos instancias de Model para almacenar los objetos animados.

```
Model venomAnimateR;  
//corriendo  
Model venomAnimateC;
```

Después de esto se utilizó una misma matriz de modelo para modificar su posición y rotación, con el fin de cumplir con el siguiente punto.

```
glm::mat4 modelMatrixVenom = glm::mat4(1.0f);
```

Luego de agregar la matriz de transformación, procedemos a cargar los modelos correspondientes, con su respectivo shader.

```
//venom  
venomAnimateR.loadModel("../models/venom/source/venomS.fbx");  
venomAnimateR.setShader(&shaderMullighting);  
  
venomAnimateC.loadModel("../models/venom/source/venomC.fbx");  
venomAnimateC.setShader(&shaderMullighting);
```

Por último, se agregó el renderizado de los objetos, utilizando un escalado ya que el modelo era demasiado grande, además la transición entre imágenes se hará con las teclas, vistas en el siguiente punto.

```
glm::mat4 modelMatrixVenomBody = glm::mat4(modelMatrixVenom);  
modelMatrixVenomBody = glm::scale(modelMatrixVenomBody, glm::vec3(.01, .01, .01));  
if(animselect == 0)  
venomAnimateR.render(modelMatrixVenomBody);  
else if (animselect == 1)  
venomAnimateC.render(modelMatrixVenomBody);
```

Esto se realizó junto con el siguiente punto, básicamente para colocar una transición entre animaciones.

6. Agregar el control del modelo con las flechas del teclado y manejar estos controles:  
(15 pts.)

- Cuando avance se debe activar la animación de caminado.
- Cuando esté detenido que tenga la animación en reposo.

Dentro de este apartado, para realizar un control las teclas, se tuvo que aumentar el número de modelos seleccionados que se utilizan dentro del proyecto, actualizado a 5, luego se procedió a realizar una construcción de ifs anidados para la rotación y traslación del modelo para su correspondiente movimiento, quedando de la siguiente manera.

```
if (modelSelected == 5 && glfwGetKey(window, GLFW_KEY_LEFT) == GLFW_PRESS)
    modelMatrixVenom = glm::rotate(modelMatrixVenom, 0.02f, glm::vec3(0, 1, 0));
else if (modelSelected == 5 && glfwGetKey(window, GLFW_KEY_RIGHT) == GLFW_PRESS)
    modelMatrixVenom = glm::rotate(modelMatrixVenom, -0.02f, glm::vec3(0, 1, 0));

if (modelSelected == 5 && glfwGetKey(window, GLFW_KEY_UP) == GLFW_PRESS) {
    modelMatrixVenom = glm::translate(modelMatrixVenom, glm::vec3(0.0, 0.0, 0.02));
    if (animselect == 0)
        animselect = 1;
}
else if (modelSelected == 5 && glfwGetKey(window, GLFW_KEY_DOWN) == GLFW_PRESS) {
    modelMatrixVenom = glm::translate(modelMatrixVenom, glm::vec3(0.0, 0.0, -0.02));
    if (animselect == 0)
        animselect = 1;
}

else if (modelSelected == 5 && glfwGetKey(window, GLFW_KEY_UP) == GLFW_RELEASE) {
    if (animselect == 1)
        animselect = 0;
}
else if (modelSelected == 5 && glfwGetKey(window, GLFW_KEY_DOWN) == GLFW_RELEASE) {
    if (animselect == 1)
        animselect = 0;
}
```

(Control de movimiento y animación)

Como se logra observar una vez seleccionado el modelo 5 que es el personaje introducido, podemos realizar una rotación a la izquierda o derecha con las flechas laterales, sin embargo. El cambio de animación viene con el avance o retroceso del personaje con las teclas UP y DOWN, activando una variable global que se encargará de elegir qué modelo se mandará a renderizar con la siguiente sección de código.

```
if(animselect == 0)
    venomAnimateR.render(modelMatrixVenomBody);
else if (animselect == 1)
    venomAnimateC.render(modelMatrixVenomBody);
```

(selección de modelo a renderizar)

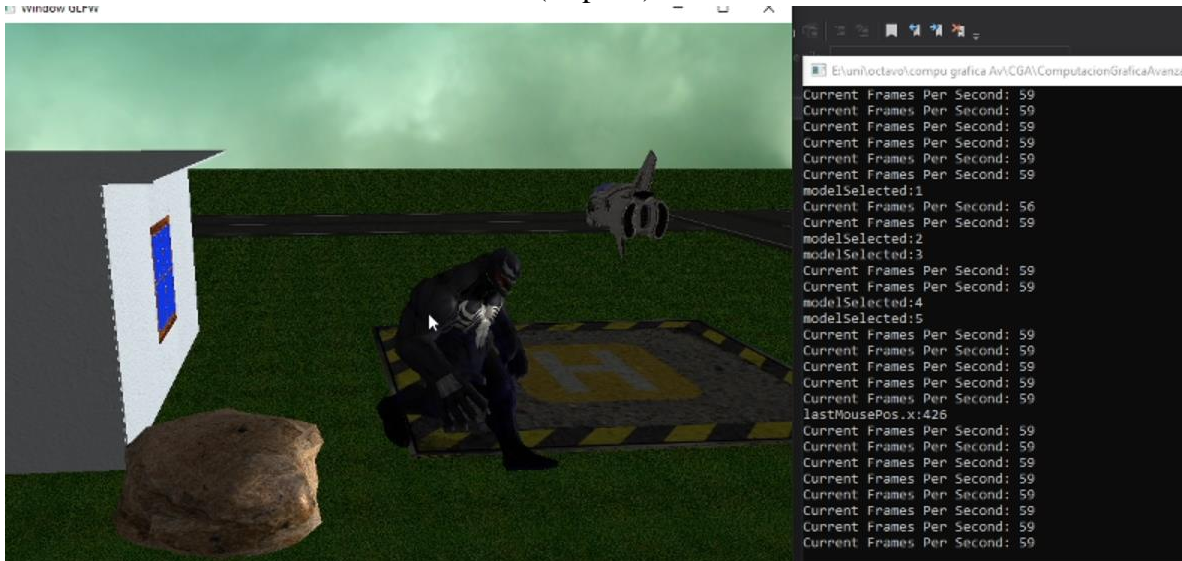


Básicamente mientras se mantenga presionado la tecla correspondiente al movimiento se cambiará a la animación 1 (animselect = 1) que es la del Venom corriendo (venomAnimateC), cuando se deje de pulsar (Release) este cambiara al estado de reposo que es la Animación 0 (VenomAnimateR).

Al final podemos observar la carga correcta de los modelos con texturas y animaciones dentro del escenario, además del uso correcto del control con las entradas del teclado.



(Reposo)



(Corriendo – Tecla UP)



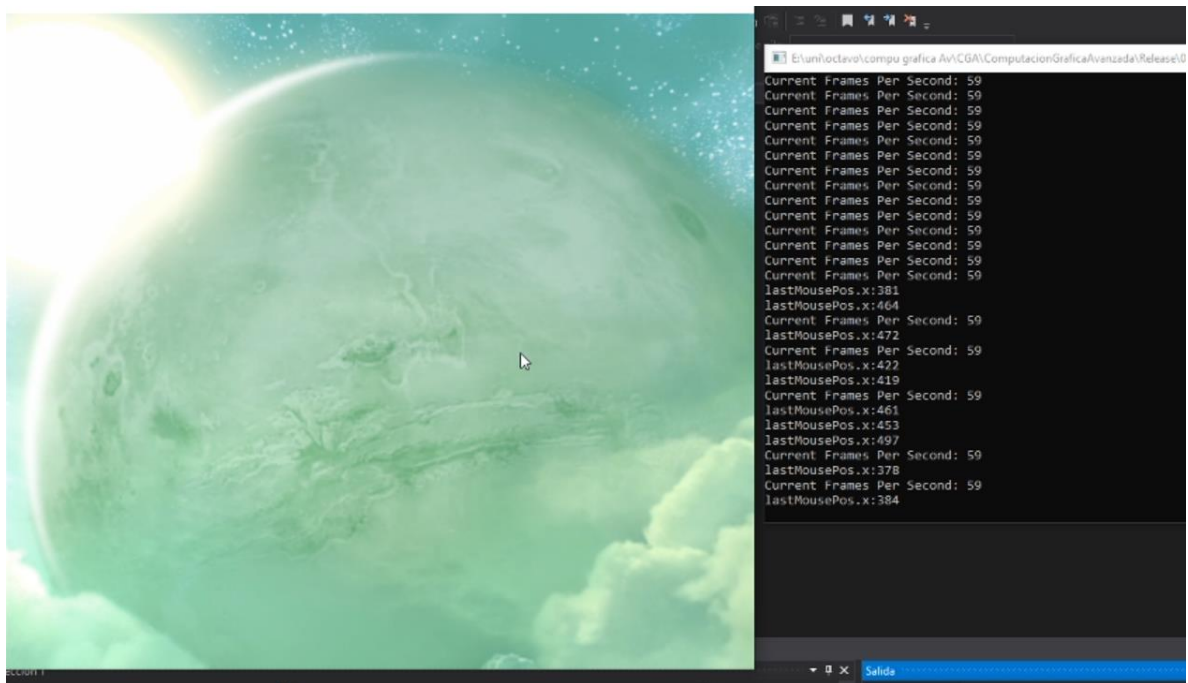
7. Cambiar el skybox con imágenes que puedan obtener de la red. (10 pts.)

Para este punto se escogió un Skybox obtenido por internet, este conjunto de archivos son básicamente imágenes con formato de imágenes pasterizadas o mapa de bits (.tga), estos son un total de 6, correspondientes a cada una de las caras del cubo, solamente se modificó la siguiente sección adecuándolo a los nombres de los archivos correspondientes en el orden ft-bk-up-dn-rt-lt.

```
std::string fileNames[6] = { "../Textures/mp_inter/interstellar_ft.tga",  
    "../Textures/mp_inter/interstellar_bk.tga",  
    "../Textures/mp_inter/interstellar_up.tga",  
    "../Textures/mp_inter/interstellar_dn.tga",  
    "../Textures/mp_inter/interstellar_rt.tga",  
    "../Textures/mp_inter/interstellar_lf.tga" };
```

(modificación de la sección de carga de imágenes del Skybox)

Obteniendo el siguiente resultado.



NOTA: para ver los resultados de una manera más clara y en vivo de esta práctica puede consultar el siguiente video: <https://youtu.be/xbxnwQJ4jxQ>

## Conclusiones

Murrieta Villegas Alfonso

En la presente práctica aprendimos sobre todo a poder modelar a través de la técnica de huesos mediante Blender, desde deformar mallas, hasta darle pesos y movimiento. Por otro lado, y como parte fundamental del curso, también aprendimos a importar y hacer funcionar estos modelos empleando OpenGL

Valdespino Mendieta Joaquin

Dentro de esta práctica pudimos realizar y comprender el uso de una de las estructuras más utilizadas dentro de la animación de objetos con articulaciones, que es el modelado por armadura o huesos, esencialmente nos permite deformaciones de las mallas de los objetos para darles movimiento y posteriormente guardarlos en keyframes usando herramientas como es el caso de Blender, con una correcta implementación en carga de estos modelos y realizar transiciones para darle vida a nuestro personaje en OpenGL para el desarrollo de entornos virtuales.

## Referencias

Video Consultados (2.1, 2.2, 2.3, 2.4, 2.4.1, 2.5, 2.6 y 2.7) de Martell, R. (2020):

<https://www.youtube.com/channel/UC1utWomTeIZx3DcawQVWxEQ/videos>

Modelo obtenido: <https://sketchfab.com/3d-models/venom-f9896507e5d747dc9b87cbd5ec055432>

Skybox: <https://opengameart.org/content/interstellar-skybox>