

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Computación Grafica Avanzada

Practica 2. Animación por Esqueletos.



Alumna: Anahi Hillary Gil González.

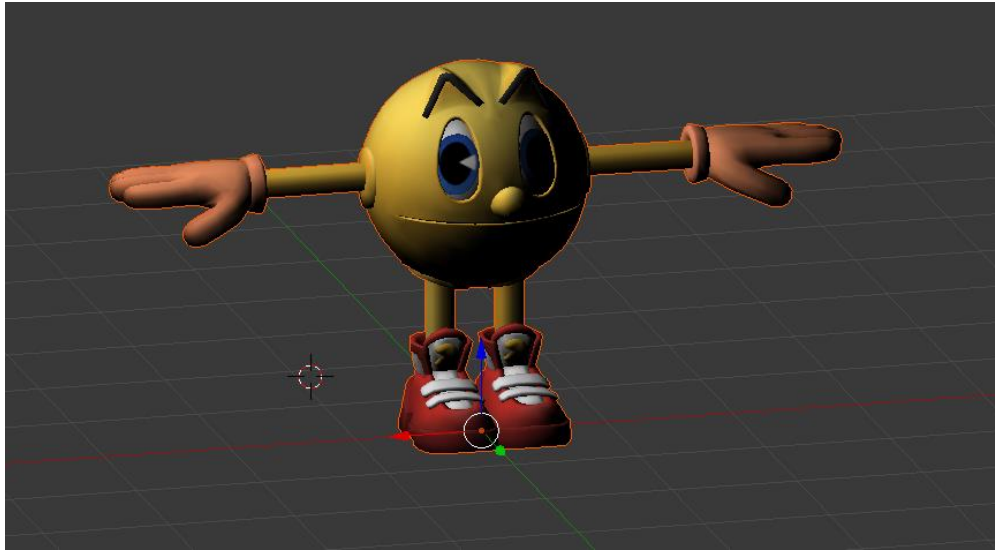
Profesor: Reynaldo Martell Ávila.

Edo. De México a 27 de octubre de 2020

Desarrollo.

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

El modelo que seleccione es la figura de un Pac-Man, al modelo original lo tuve que escalar y cambiar su posición hasta que lo pudiera visualizar de forma sencilla en Blender.

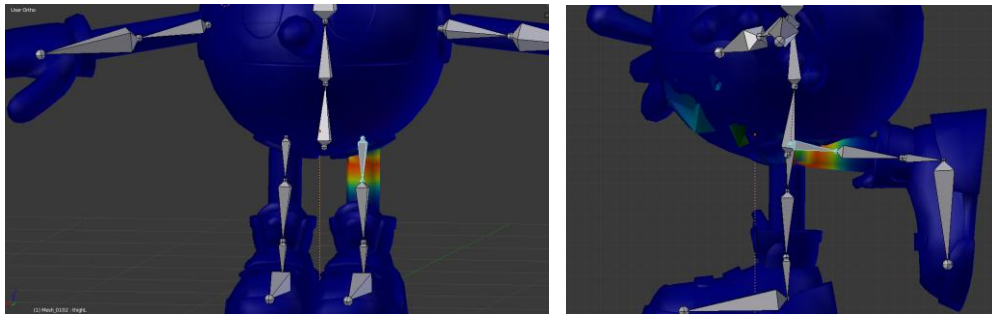


Este es el esqueleto terminado del modelo:

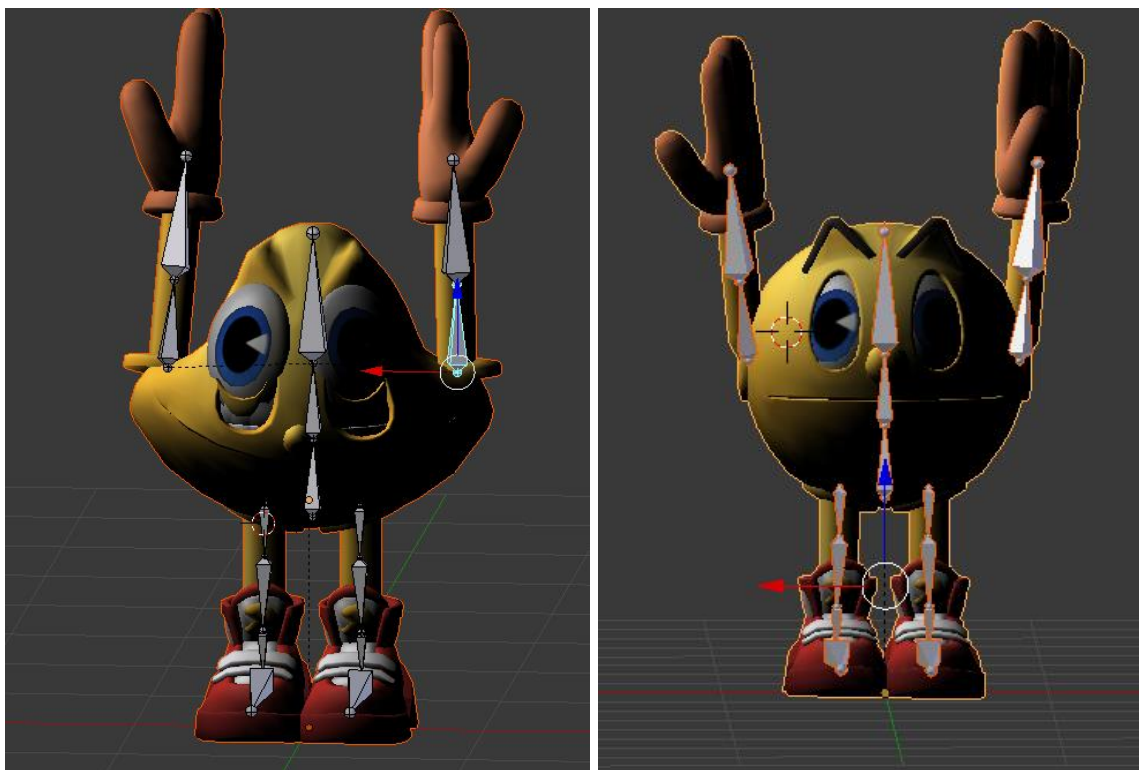


2. Ajustar los pesos para que la maya se deforme de manera correcta. (15 ptos.)

Ya que los pesos que Blender puso automáticamente en el esqueleto, no correspondían con el movimiento natural de las extremidades de Pac-Man, se procedió a modificar estos pesos, aplicándolo en brazos y piernas, torso, cabeza y pecho.

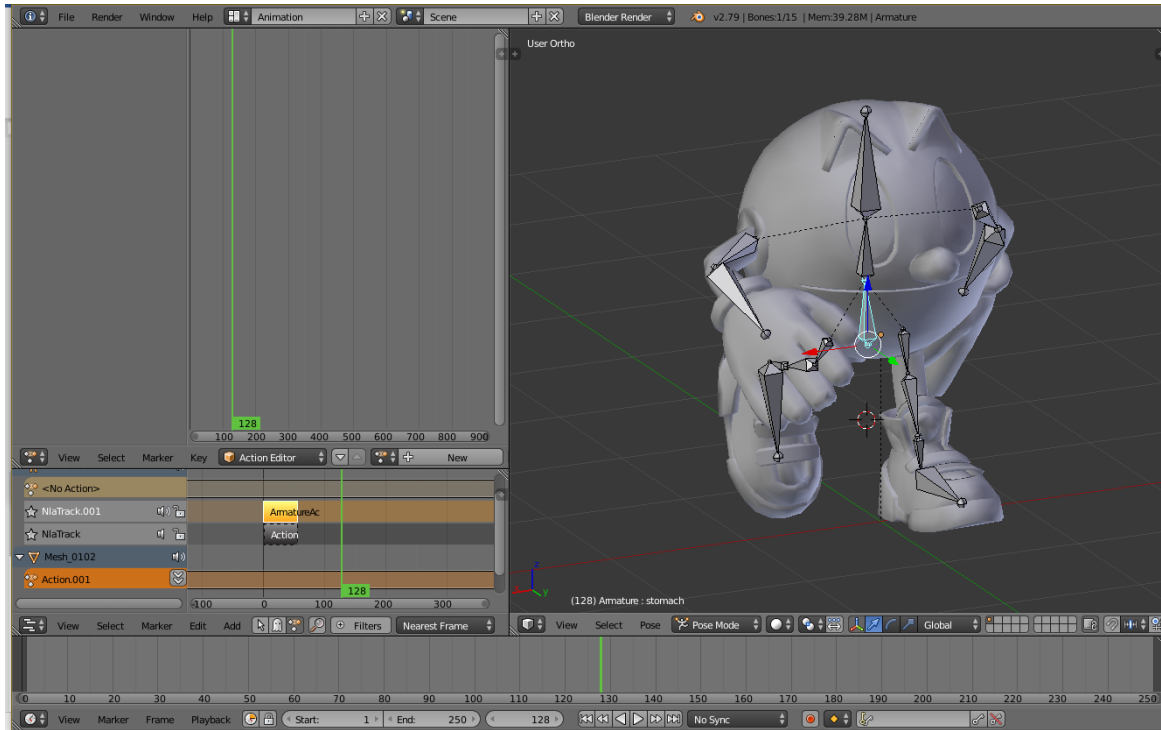


Aquí podemos ver un antes y después:



3. Realizar la animación del modelo corriendo. (15 pts.)

Para correr se sacaron 3 keyFrames del modelo para que la animación sea mas secuencias y no se vean brinco abruptos al momento de cargar la animación en OpenGL.



4. Realizar la animación del modelo en pose de descanso. (15 ptos.)

Cuando Pac-Man se encuentre en Descanso, balanceara los brazos de un lado a otro y luego levantara el brazo derecho buscando algo en el escenario.



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

Para este ejercicio fue necesario cargar cada animación por separado, así que lo primero fue declarar ambas variables del modelo cuando está descansando y corriendo:

```
86 //----->Cargando ambas animaciones de Pacman
87 Model modelPacManDescanso;
88 Model modelPacManCorriendo;
```

También se declararon dos nuevas matrices para hacer las transformaciones necesarias de ambas animaciones:

```
127 //----->Matrices para Pacman
128 glm::mat4 modelMatrixPacManDescanso = glm::mat4(1.0f);
129 glm::mat4 modelMatrixPacManCorriendo = glm::mat4(1.0f);
```

Finalmente se cargan y referencian las variables de los modelos a su correspondiente archivo fbx

```
309 //----->Cargando modelos de Pacman
310 modelPacManDescanso.loadModel("../models/PacMan/Pac-Man_Descanso.fbx");
311 modelPacManDescanso.setShader(&shaderMulLighting);
312 modelPacManCorriendo.loadModel("../models/PacMan/Pac-Man_Corriendo.fbx");
313 modelPacManCorriendo.setShader(&shaderMulLighting);
```

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

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

Para este ejercicio se declaro una nueva variable llamada banderaPacman, la cual será igual a 0 cuando Pacman esté en modo descanso y en 1 cuando Pacman este corriendo.

```
133 int banderaPacman = 0; //----->Para indicar el estado de PacMan
```

Para controlar a Pacman y poder moverlo libremente en el escenario se ocuparon las teclas flechas del teclado:

Tecla	Movimiento
↑	Adelante
↓	Atrás
→	Gira hacia la derecha
←	Gira hacia la izquierda

```
663 //----->Controles del modelo de PacMan
664 if (glfwGetKey(window, GLFW_KEY_LEFT) == GLFW_RELEASE && glfwGetKey(window, GLFW_KEY_RIGHT) == GLFW_RELEASE
665     && glfwGetKey(window, GLFW_KEY_UP) == GLFW_RELEASE && glfwGetKey(window, GLFW_KEY_DOWN) == GLFW_RELEASE)
666     banderaPacman = 0;
667 else if (glfwGetKey(window, GLFW_KEY_LEFT) == GLFW_PRESS) {
668     modelMatrixPacmanCorriendo = glm::rotate(modelMatrixPacmanCorriendo, 0.02f, glm::vec3(0, 1, 0));
669     modelMatrixPacmanDescanso = glm::rotate(modelMatrixPacmanDescanso, 0.02f, glm::vec3(0, 1, 0));
670     banderaPacman = 1;
671 }
672 else if (glfwGetKey(window, GLFW_KEY_RIGHT) == GLFW_PRESS) {
673     modelMatrixPacmanCorriendo = glm::rotate(modelMatrixPacmanCorriendo, -0.02f, glm::vec3(0, 1, 0));
674     modelMatrixPacmanDescanso = glm::rotate(modelMatrixPacmanDescanso, -0.02f, glm::vec3(0, 1, 0));
675     banderaPacman = 1;
676 }
677 else if (glfwGetKey(window, GLFW_KEY_UP) == GLFW_PRESS) {
678     modelMatrixPacmanCorriendo = glm::translate(modelMatrixPacmanCorriendo, glm::vec3(0.0, 0.0, -0.02));
679     modelMatrixPacmanDescanso = glm::translate(modelMatrixPacmanDescanso, glm::vec3(0.0, 0.0, -0.02));
680     banderaPacman = 1;
681 }
682 else if (glfwGetKey(window, GLFW_KEY_DOWN) == GLFW_PRESS) {
683     modelMatrixPacmanCorriendo = glm::translate(modelMatrixPacmanCorriendo, glm::vec3(0.0, 0.0, 0.02));
684     modelMatrixPacmanDescanso = glm::translate(modelMatrixPacmanDescanso, glm::vec3(0.0, 0.0, 0.02));
685     banderaPacman = 1;
686 }
```

Como se muestra en la imagen, mientras ninguna de las teclas antes mencionadas se presione banderaPacman será igual a 0, cuando alguna de ellas se presione banderaPacman será igual a 1.

Para poder hacer el cambio entre ambas animaciones se tomó en cuenta banderaPacman durante la renderización de los modelos de Pacman descansando y corriendo:

```
1050 //----->renderizando de Pacman según el estado de banderaPacman
1051
1052 if (banderaPacman == 1) {
1053     glm::mat4 modelMatrixPacmanCorriendoBody = glm::mat4(modelMatrixPacmanCorriendo);
1054     modelMatrixPacmanCorriendoBody = glm::scale(modelMatrixPacmanCorriendoBody, glm::vec3(0.005, 0.005, 0.005));
1055     modelPacmanCorriendo.render(modelMatrixPacmanCorriendoBody);
1056 }
1057 else {
1058     glm::mat4 modelMatrixPacmanDescansoBody = glm::mat4(modelMatrixPacmanDescanso);
1059     modelMatrixPacmanDescansoBody = glm::scale(modelMatrixPacmanDescansoBody, glm::vec3(0.005, 0.005, 0.005));
1060     modelPacmanDescanso.render(modelMatrixPacmanDescansoBody);
1061 }
```

La demostración de este ejercicio se encuentra en el siguiente enlace:

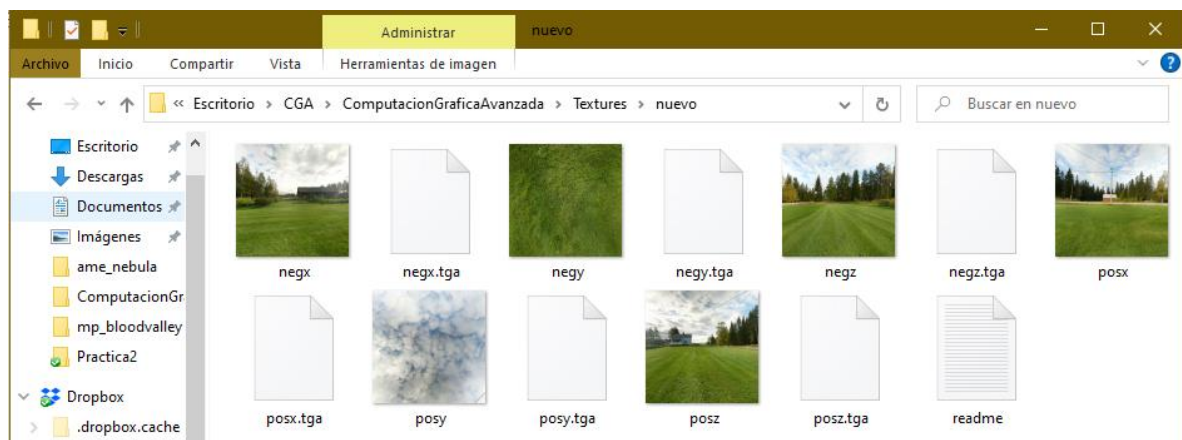
<https://youtu.be/EdouAJ6GwVo>

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

Para completar el ejercicio fue necesario descargar una skybox de la siguiente página:

<https://opengameart.org/content/skybox>

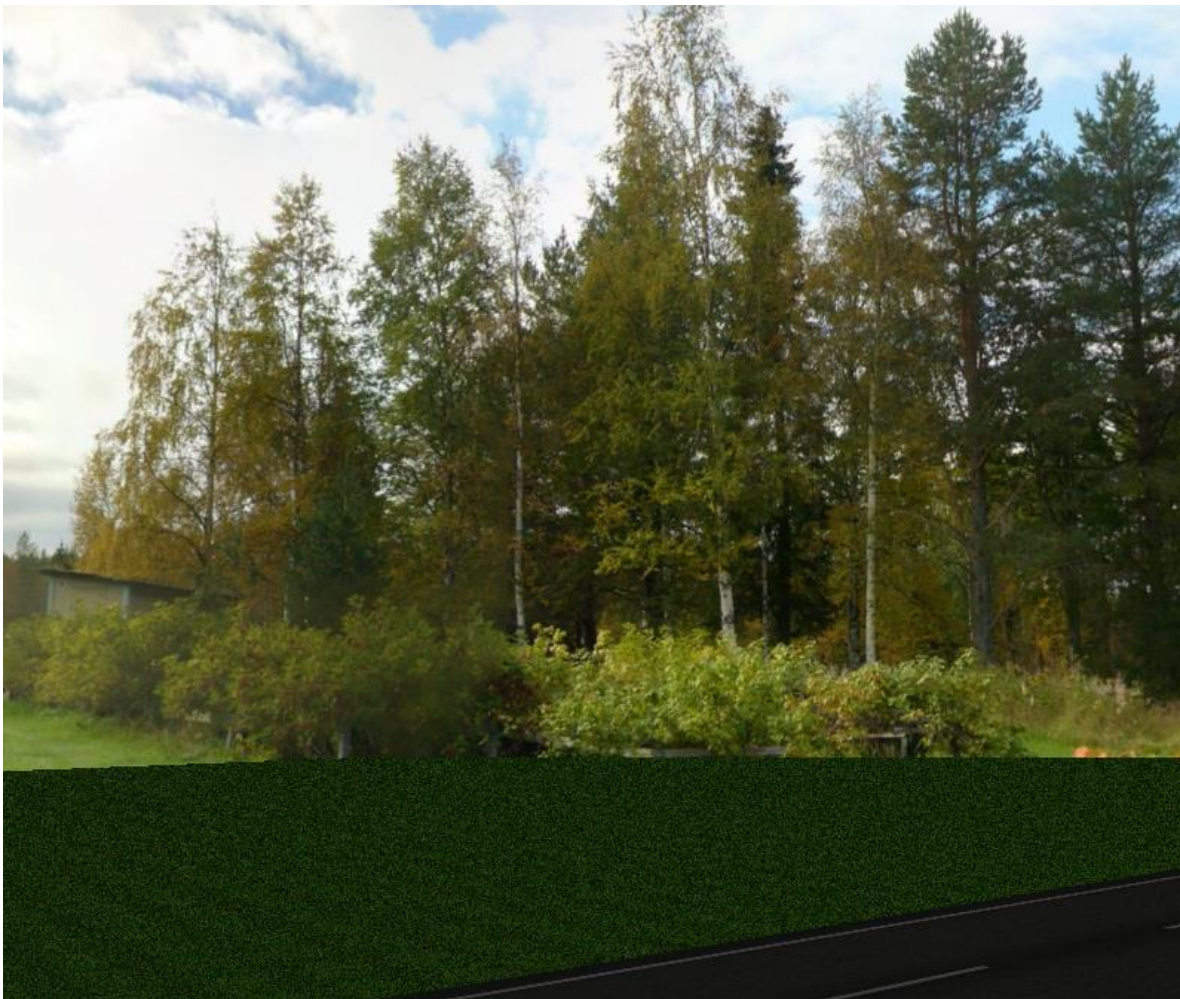
Las 6 imágenes se descargan en JPG por lo que fue necesario convertir todas en formato tga:



En el código se colocaron las direcciones correspondientes de las 6 imágenes, para que el programa pudiera cargar el skybox:

```
108 //-----Cargando Nuevo Skybox
109 std::string fileNames[6] = { "../Textures/nuevo/posx.tga",
110     "../Textures/nuevo/negx.tga",
111     "../Textures/nuevo/posy.tga",
112     "../Textures/nuevo/negy.tga",
113     "../Textures/nuevo/posz.tga",
114     "../Textures/nuevo/negz.tga" };
```

Este es el resultado:





Conclusión.

Hacer animación por esqueletos se vuelve complicado al momento de asignar pesos a los huesos, en el caso de esta practica el modelo que escogí tenia varias mayas debajo de la principal, estas mayas representaban los dientes, lengua y otras partes de la boca de Pacman y me causaron problemas ya que no podía distinguir bien a lo que le estaba asignando peso y por eso hay pequeños huecos en mi modelo.

En cuanto a la animación me parecio más sencillo hacerlo por Blender que programar toda la secuencia en OpenGL.

Enlace a repositorio

<https://github.com/HillaryGil97/ComputacionGraficaAvanzada>