

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Computación Grafica Avanzada

Practica 3. Terrenos.



Alumna: Anahi Hillary Gil González.

Profesor: Reynaldo Martell Ávila.

Edo. De México a 29 de octubre de 2020

Desarrollo.

1. Utilizando el modelo animados con las dos poses (descanso y corriendo), colocar la animación en la práctica de terreno, también debe agregar los movimientos para mover al personaje (girar y avanzar) **(20 Puntos)**.

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:

```
84 //----->Cargando ambas animaciones de Pacman
85 Model modelPacManDescanso;
86 Model modelPacManCorriendo;
```

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

```
121 //----->Matrices para Pacman
122 glm::mat4 modelMatrixPacmanDescanso = glm::mat4(1.0f);
123 glm::mat4 modelMatrixPacmanCorriendo = glm::mat4(1.0f);
```

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

```
287 //----->Cargando modelos de Pacman
288 modelPacManDescanso.loadModel("../models/PacMan/Pac-Man_Descanso.fbx");
289 modelPacManDescanso.setShader(&shaderMullighting);
290 modelPacManCorriendo.loadModel("../models/PacMan/Pac-Man_Corriendo.fbx");
291 modelPacManCorriendo.setShader(&shaderMullighting);
```

Se declaró una nueva variable llamada `banderaPacman`, la cual será igual a 0 cuando Pacman esté en modo descanso y en 1 cuando Pacman esté corriendo.

```
127 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

```

695 //----->Controles del modelo de PacMan
696 if (glfwGetKey(window, GLFW_KEY_LEFT) == GLFW_RELEASE && glfwGetKey(window, GLFW_KEY_RIGHT) == GLFW_RELEASE
697     && glfwGetKey(window, GLFW_KEY_UP) == GLFW_RELEASE && glfwGetKey(window, GLFW_KEY_DOWN) == GLFW_RELEASE)
698     banderaPacman = 0;
699 else if (glfwGetKey(window, GLFW_KEY_LEFT) == GLFW_PRESS) {
700     modelMatrixPacmanCorriendo = glm::rotate(modelMatrixPacmanCorriendo, 0.02f, glm::vec3(0, 1, 0));
701     modelMatrixPacmanDescanso = glm::rotate(modelMatrixPacmanDescanso, 0.02f, glm::vec3(0, 1, 0));
702     banderaPacman = 1;
703 }
704 else if (glfwGetKey(window, GLFW_KEY_RIGHT) == GLFW_PRESS) {
705     modelMatrixPacmanCorriendo = glm::rotate(modelMatrixPacmanCorriendo, -0.02f, glm::vec3(0, 1, 0));
706     modelMatrixPacmanDescanso = glm::rotate(modelMatrixPacmanDescanso, -0.02f, glm::vec3(0, 1, 0));
707     banderaPacman = 1;
708 }
709 else if (glfwGetKey(window, GLFW_KEY_UP) == GLFW_PRESS) {
710     modelMatrixPacmanCorriendo = glm::translate(modelMatrixPacmanCorriendo, glm::vec3(0.0, 0.0, -0.02));
711     modelMatrixPacmanDescanso = glm::translate(modelMatrixPacmanDescanso, glm::vec3(0.0, 0.0, -0.02));
712     banderaPacman = 1;
713 }
714 else if (glfwGetKey(window, GLFW_KEY_DOWN) == GLFW_PRESS) {
715     modelMatrixPacmanCorriendo = glm::translate(modelMatrixPacmanCorriendo, glm::vec3(0.0, 0.0, 0.02));
716     modelMatrixPacmanDescanso = glm::translate(modelMatrixPacmanDescanso, glm::vec3(0.0, 0.0, 0.02));
717     banderaPacman = 1;
718 }

```

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:

```

925 //----->renderizando de Pacman según el estado de banderaPacman
926
927 if (banderaPacman == 1) {
928     modelMatrixPacmanCorriendo[3][1] = terrain.getHeightTerrain(modelMatrixPacmanCorriendo[3][0], modelMatrixPacmanCorriendo[3][2]);
929     glm::vec3 aux = glm::vec3(terrain.getNormalTerrain(modelMatrixPacmanCorriendo[3][0], modelMatrixPacmanCorriendo[3][2]));
930     modelMatrixPacmanCorriendo[1][0] = aux[0];
931     modelMatrixPacmanCorriendo[1][1] = aux[1];
932     modelMatrixPacmanCorriendo[1][2] = aux[2];
933     glm::mat4 modelMatrixPacmanCorriendoBody = glm::mat4(modelMatrixPacmanCorriendo);
934     modelMatrixPacmanCorriendoBody = glm::scale(modelMatrixPacmanCorriendoBody, glm::vec3(0.005, 0.005, 0.005));
935     modelPacmanCorriendo.render(modelMatrixPacmanCorriendoBody);
936 }
937 else {
938     modelMatrixPacmanDescanso[3][1] = terrain.getHeightTerrain(modelMatrixPacmanDescanso[3][0], modelMatrixPacmanDescanso[3][2]);
939     glm::vec3 aux = glm::vec3(terrain.getNormalTerrain(modelMatrixPacmanDescanso[3][0], modelMatrixPacmanDescanso[3][2]));
940     modelMatrixPacmanDescanso[1][0] = aux[0];
941     modelMatrixPacmanDescanso[1][1] = aux[1];
942     modelMatrixPacmanDescanso[1][2] = aux[2];
943     glm::mat4 modelMatrixPacmanDescansoBody = glm::mat4(modelMatrixPacmanDescanso);
944     modelMatrixPacmanDescansoBody = glm::scale(modelMatrixPacmanDescansoBody, glm::vec3(0.005, 0.005, 0.005));
945     modelPacmanDescanso.render(modelMatrixPacmanDescansoBody);
946 }

```

El resultado final es el siguiente:

Pacman en modo descanso.

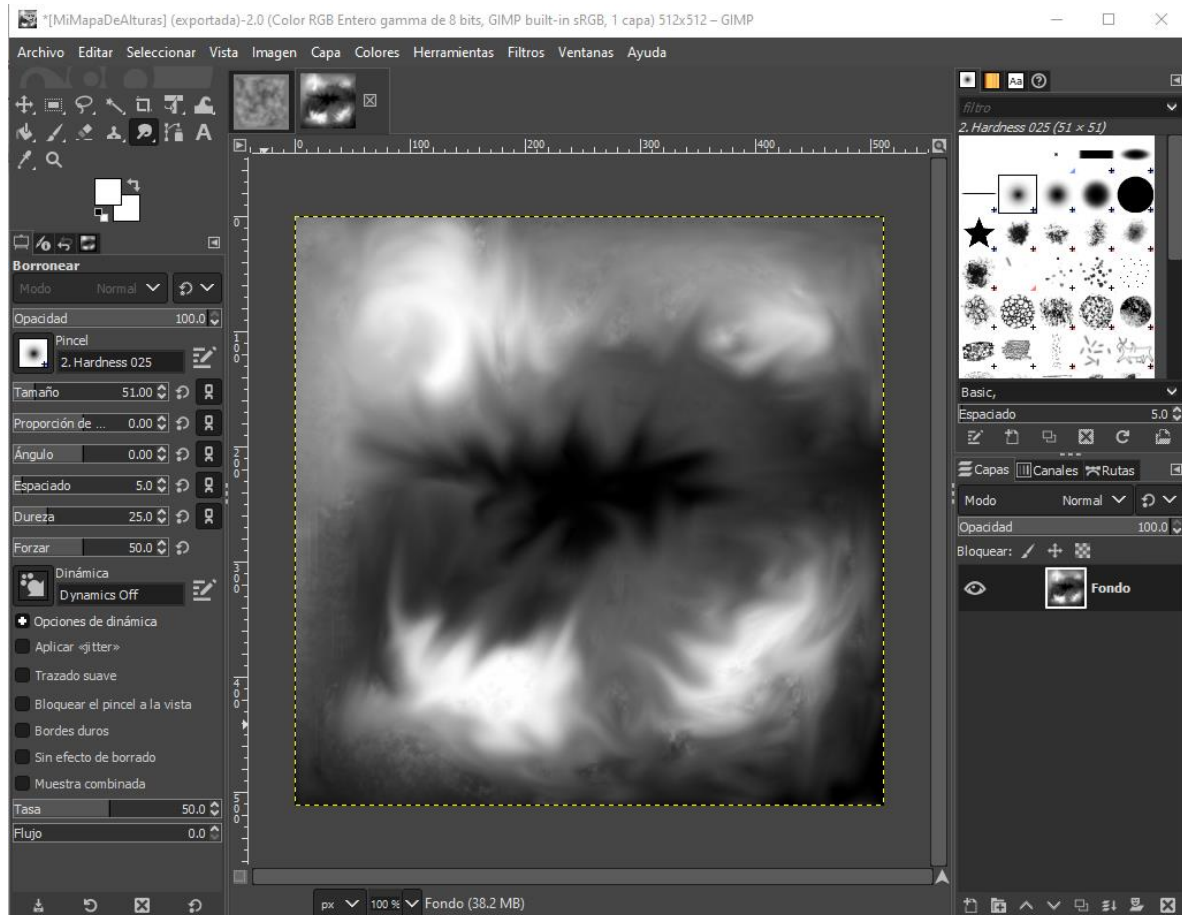


Pacman Corriendo.



2. Realizar con gimp de un mapa de alturas diferente al que se realizó y el cual servirá como base para las siguientes prácticas **(25 Puntos)**.

Se creo el mapa de altura de 512x512px y se guardó en la carpeta de texturas del proyecto.



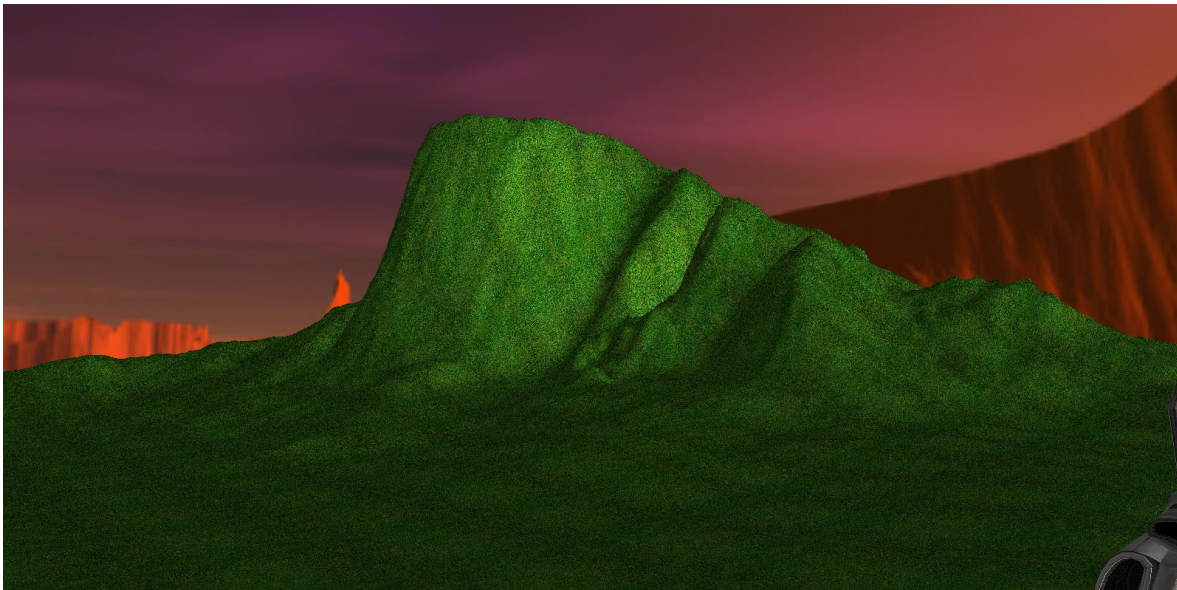
3. Cargar el mapa de alturas personalizado en OpenGL y ajustarlo (Tamaño, y altura máxima) acorde a su diseño **(25 Puntos)**.

Se cargo el mapa de alturas definiendo una rejilla de 200 y una altura máxima de 30. Tambien se situó en la parte central del escenario:

```
91 //-----> Se carga el mapa de alturas del terreno
92 Terrain terrain(-1, -1, 200, 30, "../Textures/MiMapaDeAlturas.png");

814 //----->Posiciona el mapa de alturas en el origen(a la mitad de la cantidad de rejillas)
815 terrain.setPosition(glm::vec3(100, 0, 100));
816 terrain.render();
```

El resultado final es el siguiente:



- Utilizando el método **getNormalTerrain** de la clase **Terrain** ajustar el modelo a las pendientes del mapa de alturas (**hint**: recuerden que el sistema de referencia de la matriz del modelo define los ejes de ese sistema de referencia, basta con hacer el eje **y** del modelo a la normal, y hace ortogonales los demás ejes) **(30 Puntos)**.

La función `getNormalTerrain`, permite que el modelo en movimiento pueda ajustarse a las pendientes en cada punto del terreno, para poder hacer esto se declara una nueva variable auxiliar en donde se almacenará de forma provisional el vector normal, acto seguido se asignan estos valores a la primera columna de la matriz de transformaciones del modelo.

```
925 //----->renderizando de Pacman según el estado de banderaPacman
926
927 if (banderaPacman == 1) {
928     modelMatrixPacmanCorriendo[3][1] = terrain.getHeightTerrain(modelMatrixPacmanCorriendo[3][0], modelMatrixPacmanCorriendo[3][2]);
929     glm::vec3 aux = glm::vec3(terrain.getNormalTerrain(modelMatrixPacmanCorriendo[3][0], modelMatrixPacmanCorriendo[3][2]));
930     modelMatrixPacmanCorriendo[1][0] = aux[0];
931     modelMatrixPacmanCorriendo[1][1] = aux[1];
932     modelMatrixPacmanCorriendo[1][2] = aux[2];
933     glm::mat4 modelMatrixPacmanCorriendoBody = glm::mat4(modelMatrixPacmanCorriendo);
934     modelMatrixPacmanCorriendoBody = glm::scale(modelMatrixPacmanCorriendoBody, glm::vec3(0.005, 0.005, 0.005));
935     modelPacmanCorriendo.render(modelMatrixPacmanCorriendoBody);
936 }
937 else {
938     modelMatrixPacmanDescanso[3][1] = terrain.getHeightTerrain(modelMatrixPacmanDescanso[3][0], modelMatrixPacmanDescanso[3][2]);
939     glm::vec3 aux = glm::vec3(terrain.getNormalTerrain(modelMatrixPacmanDescanso[3][0], modelMatrixPacmanDescanso[3][2]));
940     modelMatrixPacmanDescanso[1][0] = aux[0];
941     modelMatrixPacmanDescanso[1][1] = aux[1];
942     modelMatrixPacmanDescanso[1][2] = aux[2];
943     glm::mat4 modelMatrixPacmanDescansoBody = glm::mat4(modelMatrixPacmanDescanso);
944     modelMatrixPacmanDescansoBody = glm::scale(modelMatrixPacmanDescansoBody, glm::vec3(0.005, 0.005, 0.005));
945     modelPacmanDescanso.render(modelMatrixPacmanDescansoBody);
946 }
```

El resultado final es este:



Conclusión.

Para poder añadir un terreno en un escenario en 3D es necesario tomar en cuenta varias cosas, entre ellas la posición de los modelos respecto al terreno, estos factores pueden determinar la calidad de animación de los proyectos y también pueden lograr darles un toque más realista a las simulaciones. Para el caso de esta práctica se tomaron en cuenta la posición, altura y pendiente del terreno para poder mover los modelos o personajes sobre el escenario.

Enlace de demostración en video

<https://youtu.be/ryT6Z2OEheY>

Enlace a repositorio

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