

Universidad Nacional Autónoma
de México
Facultad de Ingeniería



COMPUTACIÓN GRÁFICA AVANZADA

Documentación Proyecto Final

Integrantes

Arroyo Camacho María Fernanda

Fuentes Xartuni Yerardi

Grupo

1

Semestre 2025-1

■ Introducción.

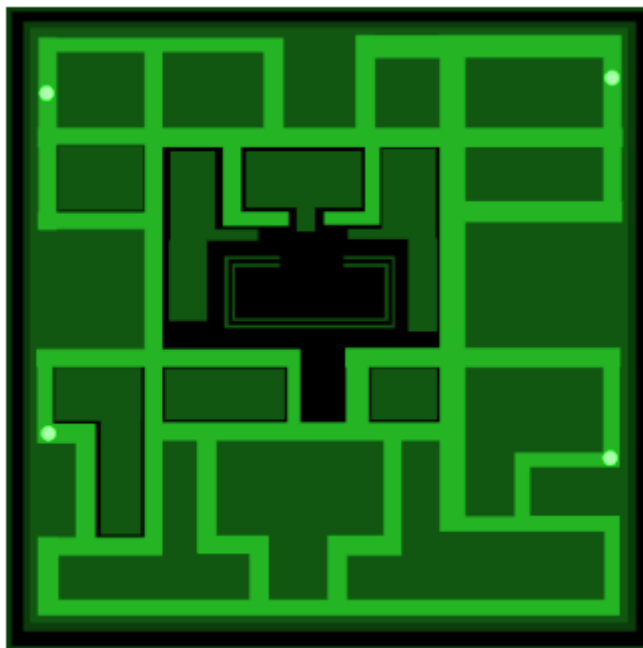
Este proyecto de computación gráfica avanzada tiene como objetivo la creación de un videojuego de simulación tipo *farm*, en el que los jugadores puedan simular el trabajo de granja, como recolectores de vegetales. Es importante destacar que se aplicarán conceptos clave para su desarrollo, como renderizado de escenas en 3D, animaciones de personajes, texturizado, iluminación y manejo de cámaras en diferentes perspectivas. Cuidando proveer una experiencia fluida y atractiva para los usuarios, que pueden ser desde niños hasta adultos. Este videojuego nos retará a aplicar los conocimientos teóricos vistos en clase, así como el desarrollo de nuestra creatividad al dar alguna propuesta de juego original en el terreno de los videojuegos modernos.

■ Objetivo del juego (Gameplay).

El objetivo principal de este videojuego es recolectar vegetales, que van sumando puntos. El mapa de juego es similar a un laberinto del cual debes salir, obteniendo la mayor cantidad posible de recolecciones. Se cuenta con 3 vidas.

■ Diseño.

1. Blend map



2. Experiencia de usuario

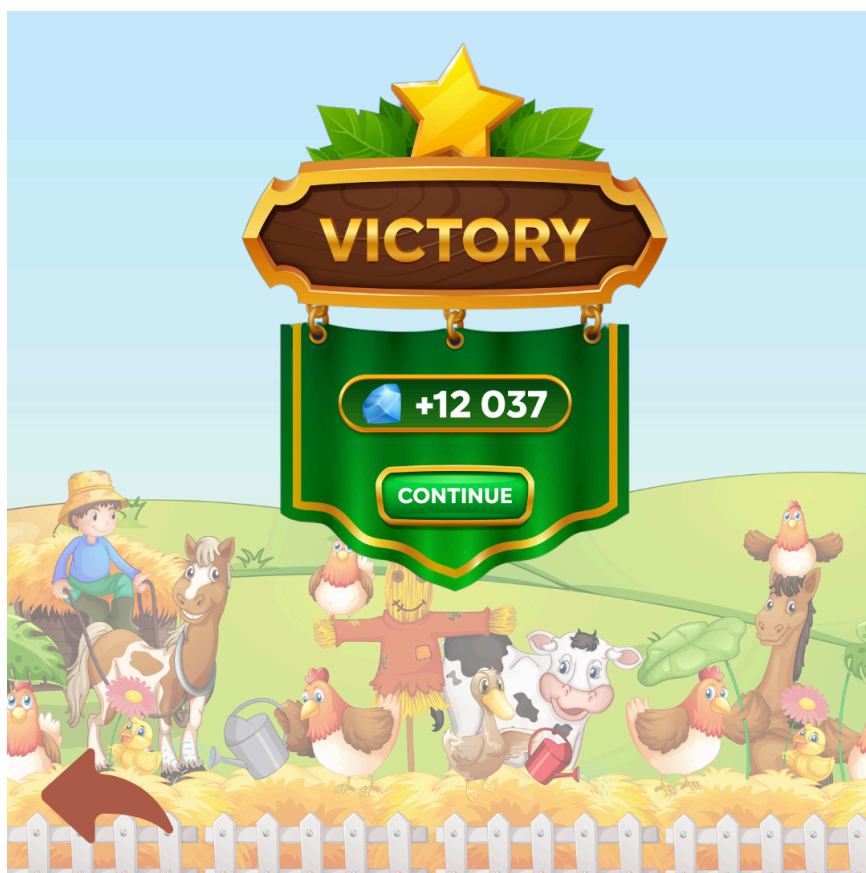
Pantalla de inicio



Game Over



Win



Personajes



Puntos



■ Plan de trabajo.

El proyecto se desarrolló siguiendo este cronograma:

- Semana 1 de noviembre: Se llevó a cabo la fase de creación e idea del proyecto y se inició la búsqueda de información y modelos necesarios así como los métodos para implementación.
- Semana 2 de noviembre: Continuó la búsqueda de información y modelos, y se comenzó la implementación inicial utilizando OpenGL. También se añadieron algunas ideas del mapa e imágenes básicas.
- Semana 3 de noviembre: Se trabajó en el desarrollo más avanzado con OpenGL y en la creación del entorno general del proyecto en donde se implementó de forma definitiva la textura, animaciones, personaje, puntos. Así mismo, se realizaron distintas pruebas generales.
- Semana 4 de noviembre: Se finalizó el desarrollo del ambiente, el sonido y las imágenes de pantalla de inicio, win y game over.
- Semana 1 de diciembre: Se realizó la documentación del proyecto.

■ Licenciamiento de modelos, modelos, sonido.

Se utilizan modelos 3D provenientes de dos fuentes principales: algunos descargados de plataformas en línea, asegurando las licencias de dichas páginas, y otros creados de manera personalizada para el proyecto. Esto garantiza que todos los modelos empleados cuentan con los permisos necesarios para su uso comercial o no comercial. De igual forma, los elementos de sonido utilizados, como música o efectos, se seleccionan bajo licencias que aseguran su utilización adecuada y conforme a los derechos de autor correspondientes.

■ Precio estimado (En cuanto venderían su proyecto).

El precio de nuestro proyecto, un innovador videojuego que combina gráficos de buena calidad, una jugabilidad envolvente y una narrativa única, es de \$1000 MXN por copia. Este costo refleja el esfuerzo del equipo, la dedicación invertida en cada detalle y la experiencia excepcional que ofrece a los jugadores. Al adquirirlo, también apoyarás la creación de productos originales que enriquecen la industria del desarrollo de videojuegos.

■ Desarrollo.

Se mencionará de manera superficial algunos puntos del desarrollo

Al igual que lo vimos en la práctica de OpenAL, se procede a generar buffers donde estarán almacenados los archivos con extensión .wav. Posteriormente, se setea la lógica para escucharlo respectivamente desde la cámara de tercera persona y más información requerida del espacio virtual para su correcto funcionamiento.

```
source1Pos[0] = modelMatrixMayow[3].x;
source1Pos[1] = modelMatrixMayow[3].y;
source1Pos[2] = modelMatrixMayow[3].z;
alSourcefv(source[1], AL_POSITION, source1Pos);

// Listener for the Thris person camera
listenerPos[0] = modelMatrixMayow[3].x;
listenerPos[1] = modelMatrixMayow[3].y;
listenerPos[2] = modelMatrixMayow[3].z;
alListenerfv(AL_POSITION, listenerPos);

glm::vec3 upModel = glm::normalize(modelMatrixMayow[1]);
glm::vec3 frontModel = glm::normalize(modelMatrixMayow[2]);

listenerOri[0] = frontModel.x;
listenerOri[1] = frontModel.y;
listenerOri[2] = frontModel.z;
listenerOri[3] = upModel.x;
listenerOri[4] = upModel.y;
listenerOri[5] = upModel.z;
```

En cuanto a los puntos:


```

glm::vec3(-14.2, 0, 9.6),
glm::vec3(-19.2, 0, 9.6),
glm::vec3(-24.2, 0, 9.6),
glm::vec3(-29.2, 0, 9.6),
glm::vec3(-34.2, 0, 9.6),
glm::vec3(-39.2, 0, 9.6),
glm::vec3(-44.2, 0, 9.6),
glm::vec3(-49.2, 0, 9.6),
glm::vec3(-54.2, 0, 9.6),
glm::vec3(-59.2, 0, 9.6),
glm::vec3(-64.2, 0, 9.6),
glm::vec3(-69.2, 0, 9.6),
glm::vec3(-74.2, 0, 9.6),
glm::vec3(-79.2, 0, 9.6),
glm::vec3(-84.2, 0, 9.6),

```

se ajustan en los distintos ejes.

La lógica de los puntos se basa en que las agarre el personaje principal, cuentan con un collider para detectar la colisión sobre los puntos y así sumarlos.

```

if(spheres_to_win == 0 && !gameFinished)
{
    modelMatrixMayow = glm::mat4(1.0);
    modelMatrixMayow = glm::translate(modelMatrixMayow, glm::vec3(0.0f, 0.05f, 5.0f));
    modelMatrixMayow = glm::rotate(modelMatrixMayow, glm::radians(-90.0f), glm::vec3(0, 1, 0));
    textureActivaID = textureVictoryID;
    alSourcePause(source[1]);
    alSourcePlay(source[4]);
    gameFinished = true;
}

```

La lógica de puntuación radica en un texto dentro del entorno OpenGL que suma puntos para el score final del usuario, puede ser mayor o igual a la cantidad de puntos que te otorgan todas las esferas del mapa.

```

modelText = new FontTypeRendering::FontTypeRendering(screenWidth, screenHeight);
modelText->Initialize();

```

El código procesa las colisiones basadas en etiquetas (it->first) con la variable REGEX, identificando el número de esfera que colisionó, con el formato de posición. El modelo del personaje principal actualiza puntuaciones y colisiones, se eliminan elementos de collidersSBB y se ajusta el número de esferas restantes para la pantalla de You Win. Finalmente, se actualiza el estado de detección de colisiones.

■ Resultados y trabajo a futuro.

Los resultados obtenidos para este proyecto fueron favorables, sin embargo notamos algunas áreas de mejora. Por ejemplo, la eficiencia del código y la

utilización de recursos para asegurar un rendimiento óptimo en diferentes dispositivos.

Respecto a la experiencia de usuario, la creación de niveles y otros mapas de juego pueden brindar una experiencia más divertida al aumentar la dificultad para los jugadores. Inclusive se podrían implementar diferentes modos de juego, más personajes o diferentes skins para el personaje principal.

También, desarrollar un menú de opciones más completo que permita a los usuarios personalizar la experiencia de juego como los ajustes de sonido, gráficos, controles, entre otros.

En general, las principales mejoras que notamos serían retroalimentadas por los usuarios, la conectividad entre ellos y la dificultad de los niveles.

■ Conclusiones.

El desarrollo de este proyecto nos ofreció una gran experiencia con respecto a la creación de videojuegos mediante la implementación de los conceptos aprendidos en clase como lo son animaciones, iluminación, texturizado, colusiones, generación de audio, entre otros. Sin embargo, nos presentó un enorme reto su realización. Desde la correcta instalación del software hasta las especificaciones de diseño y retos creativos, así como las restricciones y posibles mejoras para que la experiencia de usuario fuera entretenida, el cual es el principal propósito de cualquier videojuego.

■ Referencias.

Martell Ávila, R. (06 de 12 de 2024). Youtube. Obtenido de Reynaldo Martell Avila: <https://www.youtube.com/@reynaldomartellavila9051>

Mixamo, Adobe. (06 de 12 de 2024). Adobe. Obtenido de Mixamo: <https://www.mixamo.com/>

OpenGL. (06 de 12 de 2024). OpenGL. Obtenido de The Industry's Foundation for High Performance Graphics: <https://www.opengl.org/>