**Universidad Nacional de Ingeniería UNI**

**Facultad de Electrotecnia Computación**

**Recinto Universitario Simón Bolívar R.U.S.B.**

**Proyecto Final de Programación Grafica**

**El Laberinto del Tesoro**

**Elaborado por:**

* Sánchez Espinosa Ernesto Gabriel.
* Vivas Mena Ángel Lenin.
* Espinoza Donayre Erick Joao.
* Pavón Solís Russell Ali.

**Carné:**

* 2022-0351U.
* 2020-0326U.
* 2020-0333U.
* 2022-0534U.

**Grupo:** 3T1-CO.

**Docente:** Danny Oswaldo Chávez .

Fecha: 03/04/2024

**Resumen**

En la última década, la industria de los videojuegos ha visto un crecimiento significativo y diversificación, con los juegos de terror ganando popularidad por su capacidad de ofrecer experiencias inmersivas. Este documento detalla el desarrollo de un videojuego de terror basado en un laberinto, creado con C++ y OpenGL en Visual Studio. El juego presenta mecánicas únicas como la interacción con estatuas guía y el uso estratégico de una radio con distorsión variable para localizar un cofre del tesoro. A través de pruebas rigurosas y optimización, se logró equilibrar la inmersión y el desafío, proporcionando a los jugadores una experiencia gratificante y desafiante. Este proyecto destaca por combinar habilidades técnicas en programación y diseño de juegos para crear un producto sólido y atractivo desde el punto de vista técnico y creativo.

**Introducción**

En la última década, la industria de los videojuegos ha experimentado un notable crecimiento y diversificación. Los videojuegos de terror han capturado la imaginación de millones, ofreciendo experiencias intensas y envolventes. Este documento describe el proceso de desarrollo de un videojuego de terror basado en un laberinto, utilizando *C++* y *OpenGL* en el entorno de desarrollo Visual Studio. El juego presenta mecánicas únicas, como la interacción con estatuas guía y la gestión del riesgo que representa un monstruo merodeador. Además, se integra una radio que facilita la búsqueda del objetivo principal: un cofre del tesoro, cuya proximidad se refleja en la distorsión de la señal de la radio. Este proyecto busca no solo desarrollar habilidades técnicas en programación y diseño de videojuegos, sino también ofrecer una experiencia de juego inmersiva que combine el miedo y la estrategia.

**Objetivos**

* **Desarrollo Técnico**: Implementar un videojuego en *C++* y *OpenGL* utilizando Visual Studio, que incluya mecánicas de juego básicas.
* **Experiencia del Usuario**: Crear una atmósfera de terror envolvente que combine elementos visuales y sonoros para mejorar la experiencia del jugador.
* **Interacción del Jugador**: Diseñar y programar mecánicas de juego que involucren el uso de estatuas como guías, y una radio para localizar el tesoro, aumentando la tensión mediante la amenaza constante de un monstruo capaz de escuchar los ruidos de la radio.

**Desarrollo del Proyecto**

En el inicio del proyecto se llevó a cabo un proceso de lluvia de ideas que generó un amplio repertorio de propuestas, centrado principalmente en el ámbito de los videojuegos. Tras varios debates sobre qué podría resultar llamativo y accesible con los conocimientos adquiridos, surgió la idea de un videojuego de terror, utilizando los recursos computacionales para jugar con la mente de los usuarios.

A pesar de que la idea inicial era un juego de terror, también se propuso la creación de un videojuego del género shooter. Se mantuvieron los avances del proyecto original mientras desarrollábamos esta segunda idea. Después de investigar en varios repositorios relacionados con este género, se determinó que la complejidad matemática requerida para llevar a cabo este nuevo proyecto era excesiva para los conocimientos del grupo. Por lo tanto, se decidió retomar el proyecto inicial.

Los avances hasta ese momento incluían la implementación de una linterna, un minimapa funcional y un mapa descargado como prueba para la programación del sistema de colisiones. Este mapa resultó cómodo inicialmente para los integrantes del grupo, pero no era fácil de editar ni decorar con lo que se optó por investigar por otros estilos de mapeado; en este proceso se definieron las mecánicas de juego y el argumento de este.

Como inspiración se tomaron juegos como Slenderman, que se desarrollan en un laberinto oscuro donde hay estatuas colocadas estratégicamente como guías hacia la meta. Además, un monstruo acecha al jugador. El objetivo del personaje es encontrar un cofre oculto en la oscuridad del laberinto, lo que llevó a descartar la función del minimapa. Para evitar al monstruo, el jugador debe estar atento a la estática y apagar la linterna cuando esté cerca del monstruo. También se implementó una radio que ayuda a encontrar la ubicación del cofre (siendo este el final del juego), con la desventaja de que usarla por mucho tiempo atrae al monstruo debido al ruido que produce.

Con respecto a la creación del mapa, se utilizó una técnica de mapeado 2D en el cual se ingresan símbolos representativos de elementos de un mapa en un archivo .txt, el cual contenía una disposición de símbolos que marcaban los muros, la posición inicial del jugador, la posición inicial del enemigo, el posicionamiento de las estatuas guía y por último la posición del cofre meta.

En cuanto a la programación, a pesar de la optimización realizada, se encontró que faltaban muchos elementos estéticos y funcionales para el género elegido. Se recurrió a diversos repositorios, tutoriales de *YouTube* y consultas con *ChatGPT* para construir paso a paso los archivos con los cálculos geométricos necesarios.

Inicialmente se planeó utilizar *Python* como lenguaje de programación, pero el tiempo necesario para aprenderlo y familiarizarse con él no cumplía con el plazo de entrega establecido. Por lo tanto, se optó por *C++*, dado que los conocimientos existentes eran suficientes para trabajar con este lenguaje y facilitar la integración con *OpenGL*. Se utilizó Visual Studio como entorno de desarrollo y herramientas como *CMake*.

Una vez desarrollado el prototipo del videojuego, se realizaron pruebas exhaustivas para detectar errores o mal funcionamiento. Las principales dificultades encontradas fueron la programación del sistema de colisiones, la skybox (inicialmente se había planeado que el laberinto fuera un espacio cerrado) y el comportamiento del monstruo.

Después de realizar los cambios necesarios para refinar el código, se logró plasmar las ideas desarrolladas por el equipo en un videojuego. Gracias a las habilidades individuales de cada integrante, se logró convertir lo que parecía un desafío imposible en una realidad cumplida dentro del tiempo y las expectativas establecidas.

**Conclusión**

El desarrollo de este videojuego de terror ha permitido explorar y aplicar diversas habilidades en programación, diseño de juegos y optimización de rendimiento. La implementación de mecánicas de juego innovadoras, como el uso de estatuas guía y una radio con distorsión variable, ha proporcionado una experiencia de juego única y envolvente. A través de pruebas rigurosas y optimización, se ha logrado un equilibrio entre la inmersión y el desafío, ofreciendo una experiencia de terror gratificante y desafiante para los jugadores. Este proyecto sirve como un ejemplo de cómo combinar técnicas de programación avanzada con creatividad en el diseño de juegos para crear productos atractivos y técnicamente sólidos.

**Referencias Bibliográficas**

* American Psychological Association. (2020). *Publication Manual of the American Psychological Association (7th ed.).* Washington, DC: American Psychological Association.
* Dörner, R., Göbel, S., Effelsberg, W., & Wiemeyer, J. (2016). *Serious Games: Foundations, Concepts and Practice*. Springer.
* Prensky, M. (2001). *Digital Game-Based Learning*. McGraw-Hill.
* Sheraadams. (s. f.). *GitHub - sheraadams/OpenGL-Skyboxes-and-Photoshop-tutorial: Creative commons skyboxes for use in OpenGL C++ and photoshop tutorial for how to create custom high resolution cubemap skyboxes*. GitHub. <https://github.com/sheraadams/OpenGL-Skyboxes-and-Photoshop-tutorial>
* *Tutorial 25 - SkyBox*. (s. f.). <https://ogldev.org/www/tutorial25/tutorial25.html>
* Amin. (2019, 28 julio). *Creating a Skybox Using C++, Qt and OpenGL*. Amin. <https://amin-ahmadi.com/2019/07/28/creating-a-skybox-using-c-qt-and-opengl/>
* Assimp. (s. f.). *GitHub - assimp/assimp: The official Open-Asset-Importer-Library Repository. Loads 40+ 3D-file-formats into one unified and clean data structure.* GitHub. <https://github.com/assimp/assimp>
* The Cherno. (2017, 17 septiembre). *Welcome to OpenGL* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=W3gAzLwfIP0>
* *OpenGL básico*. (s. f.). <https://www.opengl-tutorial.org/es/beginners-tutorials/>
* *LearnOpenGL - OpenGL*. (s. f.). <https://learnopengl.com/Getting-started/OpenGL>
* *Estatua v1 Modelo 3D gratis - .obj .stl - Free3D*. (s. f.). <https://free3d.com/es/modelo-3d/statue-v1--561600.html>
* *Calavera v3 Modelo 3D gratis - .obj - Free3D*. (s. f.). <https://free3d.com/es/modelo-3d/skull-v3--785914.html>
* Sketchfab. (s. f.). *Forest Skull - Download Free 3D model by FletchTech*. <https://sketchfab.com/3d-models/forest-skull-ce4d33f9b9a3440f97aeec7ed9293b79>
* Sketchfab. (s. f.-a). *Forest Ground 10 - Download Free 3D model by Publicdomaintextures*. <https://sketchfab.com/3d-models/forest-ground-10-5d5abe1813f942b5ab3bc6c41447d1ca>
* Sketchfab. (s. f.-c). *Rustic Wood Material 1 - Download Free 3D model by chrisg4919*. <https://sketchfab.com/3d-models/rustic-wood-material-1-7e9205e31cd841948f1682fed50901d8>
* Sketchfab. (s. f.-c). *Macabre Gargoyle Statue - Download Free 3D model by UMBRAVISIO*. https://sketchfab.com/3d-models/macabre-gargoyle-statue-a3120d7783a44fd2b3d52392eda37ad4
* *Freesound*. (s. f.). https://freesound.org/