

PEC 3 Maze

Maze3D es una versión muchísimo más simple, pero que parece el juego doom original.

Desarrollado por Andrés López Torres

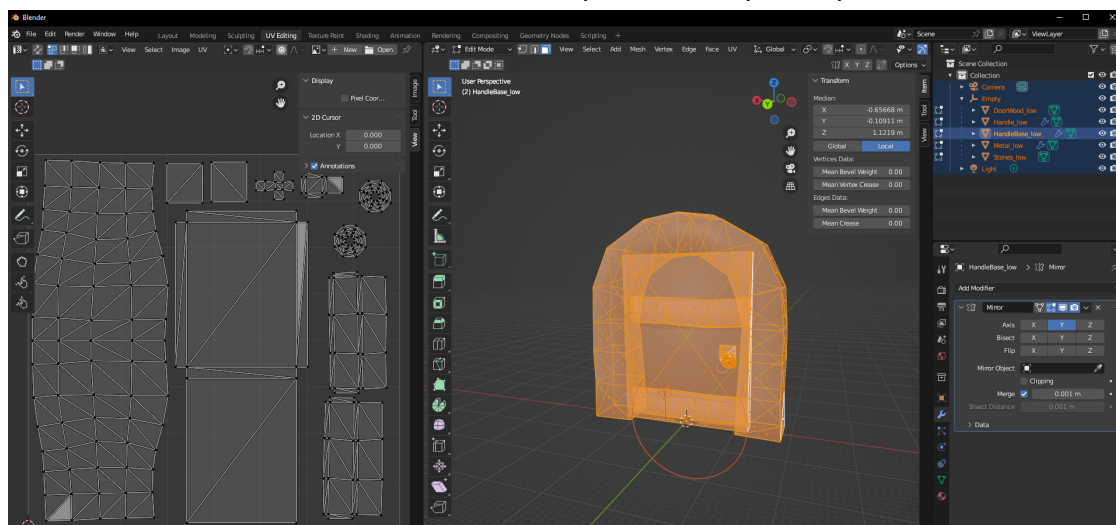
Enero del 2023

Realización:

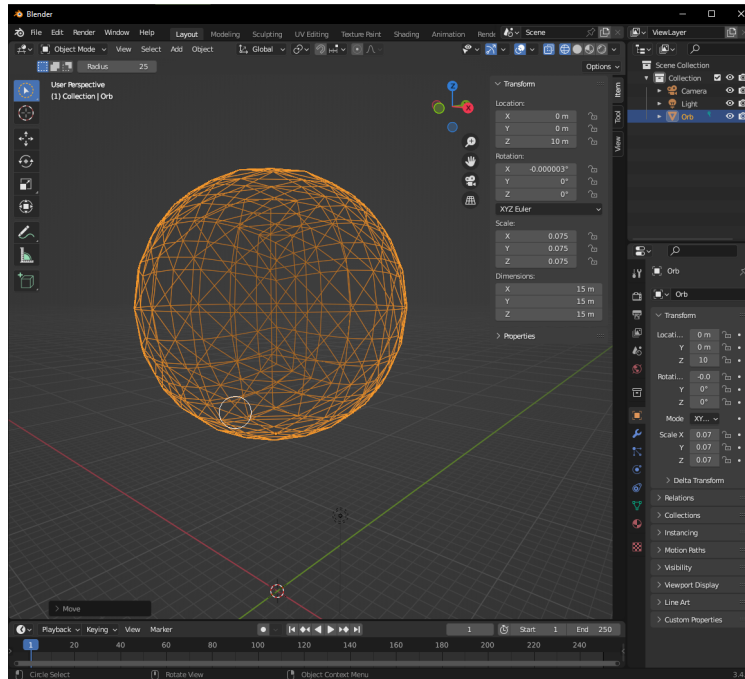
Se procede a optimizar el método GenMeshCubicmap ahora llamado GenMeshCubicmapV2, que permite realizar lo que se solicita en la guía accediendo a dos vectores globales creados que guardan la lista de props que estarán presente en cada nivel y la lista de objetos que indica el tipo de prop y la posición para ser añadidos en el mapa, basados en los colores que se definan en el rango verde, donde 0 es sin prop de ahí en adelante se ocuparan los props necesarios.

Para la puerta, quise buscar qué tan complejo era incluir un objeto 3d descargado de internet, así que tomé el modelo Low Poly Stylized Door del usuario [Murat Can ÜNAL](https://sketchfab.com/3d-models/low-poly-stylized-door-705f87c0501a41e487d4480d092120da), de sketchfab (<https://sketchfab.com/3d-models/low-poly-stylized-door-705f87c0501a41e487d4480d092120da>) y le hice unas cuantas modificaciones para poderlo llevar al proyecto, (modificaciones tales como escala, y la posibilidad de usar la puerta por detrás o por el frente para que sea más fácil de acomodar en el mapa y no tener modelos diferentes de puertas.

Adjunto pantallazo, donde he verificado que el modelo está triangulado y con sus uvs correctas, así como añadir los modificadores de mirror para usar la puerta por ambos lados



Se añade un coleccionable en el juego (un orbe mágico) haciendo lo mismo que en el paso anterior:



También tomado de sketchfab:

<https://sketchfab.com/3d-models/magic-orb-348d3a01dd31484eb339a397b94e6d2b>

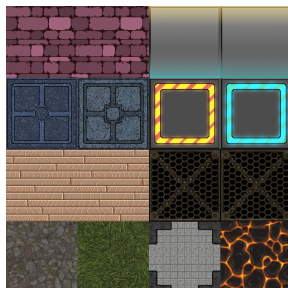
Por el usuario [epma](#).

Siguiendo con la modificación del método anterior, se implementó que cuando rojo fuera 0, era un bloque transitable, si era 128 es para poner props y si era 255 es una pared.

Así mismo para poder colocar un prop, se organizó de tal forma que el canal verde siendo 0, indica que no hay prop y del 1 al 5 están los props correspondientes al proyecto, siendo así:

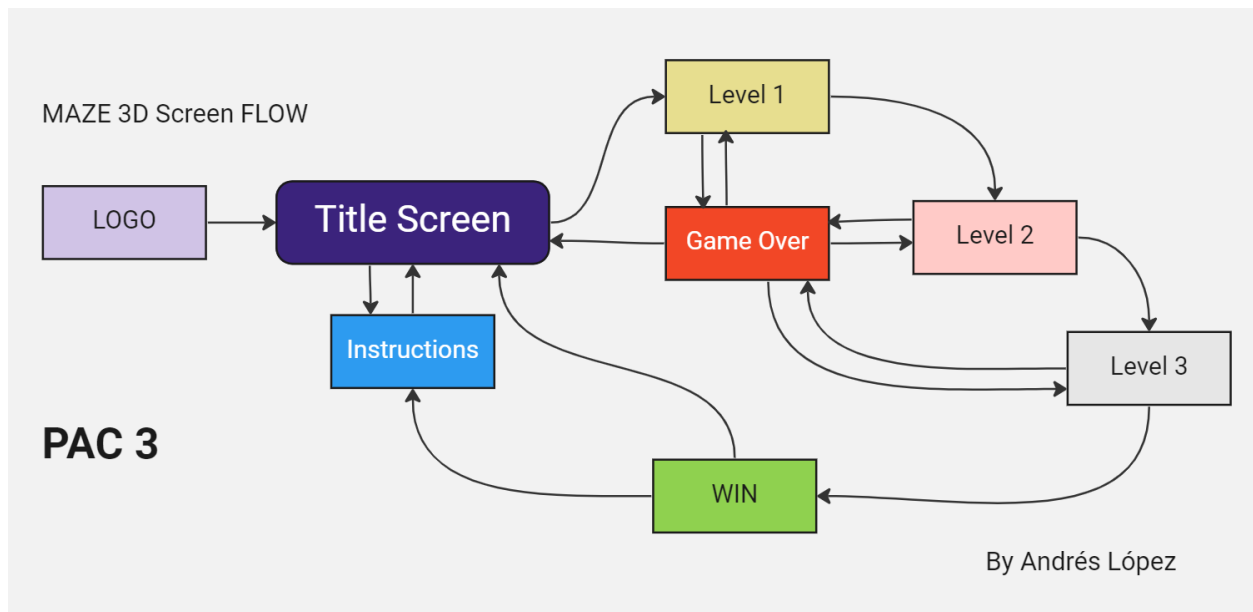
- 1 - Castillo
- 2 - Torre
- 3 - Avion
- 4 - Puerta de Salida
- 5 - Coleccionable

Para finalizar la modificación del método, se garantizo que cuando el canal azul tuviera los valores de 0 a 3, tendría las texturas diferentes correspondientes en el atlas consolidado:



Siendo 0 cuadrante superior izquierdo, 1 superior derecho, 2 inferior derecho y 3 inferior izquierdo.

Diagrama de Flujo de pantallas



Como se aprecia en el gráfico anterior tenemos las siguientes pantallas:

Logo: Se muestra el logo y después de un tiempo pasamos a la pantalla del título (no es posible volver a regresar a esta pantalla sin cerrar la aplicación).

Title Screen: Tenemos la pantalla del título donde podremos empezar a jugar (Desde el nivel 1) o ir a instrucciones.

Level 1-3: Antes de empezar a jugar en un nivel se nos presentará una pantalla que no está en el diagrama, pero que se usa para Cargar la información de cada nivel (dicha pantalla aparecerá por 3 segundos), luego en el nivel ya cargado, solo podremos o avanzar (llegar a la puerta para cambiar de nivel si hemos coleccionado todos los orbes o para ganar el juego si estamos en el nivel 3) o perder al caer en la lava. Desde aquí, al ir a la pantalla de game over tendremos la opción de continuar en el nivel actual.

Game Over: esta pantalla solo permite continuar en el nivel actual o regresar a la pantalla del título.

Win: La pantalla de Ganar solo se puede acceder al finalizar el level 3 y solo permitirá ir a la pantalla del título o a la pantalla de instrucciones.

Instructions: a esta pantalla solo podemos llegar desde la pantalla de ganar o desde la pantalla del título, esta pantalla solo redirigirá a la pantalla del título y no permite ir a ninguna otra.

Posteriormente se crearon los 3 mapas requeridos para el juego, de acuerdo a las indicaciones dadas en la guía.

Se utiliza el software ASEPRITE para facilitar la creación de los mismos.

Se añade la pantalla de carga entre cada nivel.

Interacción

Se implementa la mecánica de recoger orbes como requerimiento para pasar cada nivel (de esta forma se promueve la exploración del escenario :))

Sonido

Se agregan sonidos para cuando se cambia de nivel (abrir una puerta).

Se agrega sonido al caer a la lava.

Se agregan músicas para el título, nivel1, nivel2, nivel3 y pantalla de game over.

Controles

W, A, S, D: para mover al personaje

Ratón: mueve la cámara del personaje

ESC: Sale de la aplicación.

Recursos externos

<https://opengameart.org/content/jrpg-pack-1-exploration> Pack de Musicas incluidas en el juego

<https://opengameart.org/content/ui-sfx-pack-2-sample> Efecto del coleccionable

<https://opengameart.org/content/5-hit-sounds-dying> Sonido de muerte del personaje

<https://opengameart.org/content/5-chiptunes-action> Pista Ending para la pantalla de Game Over

/ Win

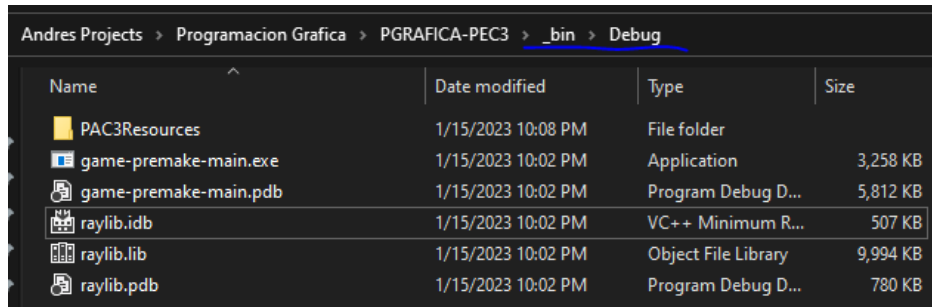
<https://sketchfab.com/3d-models/low-poly-stylized-door-705f87c0501a41e487d4480d092120da>

Modelo de la puerta

<https://sketchfab.com/3d-models/magic-orb-348d3a01dd31484eb339a397b94e6d2b> Modelo del orbe.

Version Compilada

Se deja en el directorio `_bin` en la carpeta Debug con los recursos correspondientes.



Name	Date modified	Type	Size
PAC3Resources	1/15/2023 10:08 PM	File folder	
game-premake-main.exe	1/15/2023 10:02 PM	Application	3,258 KB
game-premake-main.pdb	1/15/2023 10:02 PM	Program Debug D...	5,812 KB
raylib.idb	1/15/2023 10:02 PM	VC++ Minimum R...	507 KB
raylib.lib	1/15/2023 10:02 PM	Object File Library	9,994 KB
raylib.pdb	1/15/2023 10:02 PM	Program Debug D...	780 KB