Generación de mapas y raycasting en 3D

Imagen Digital – Rubén Vázquez Angamarca

Índice

Explicación del proyecto

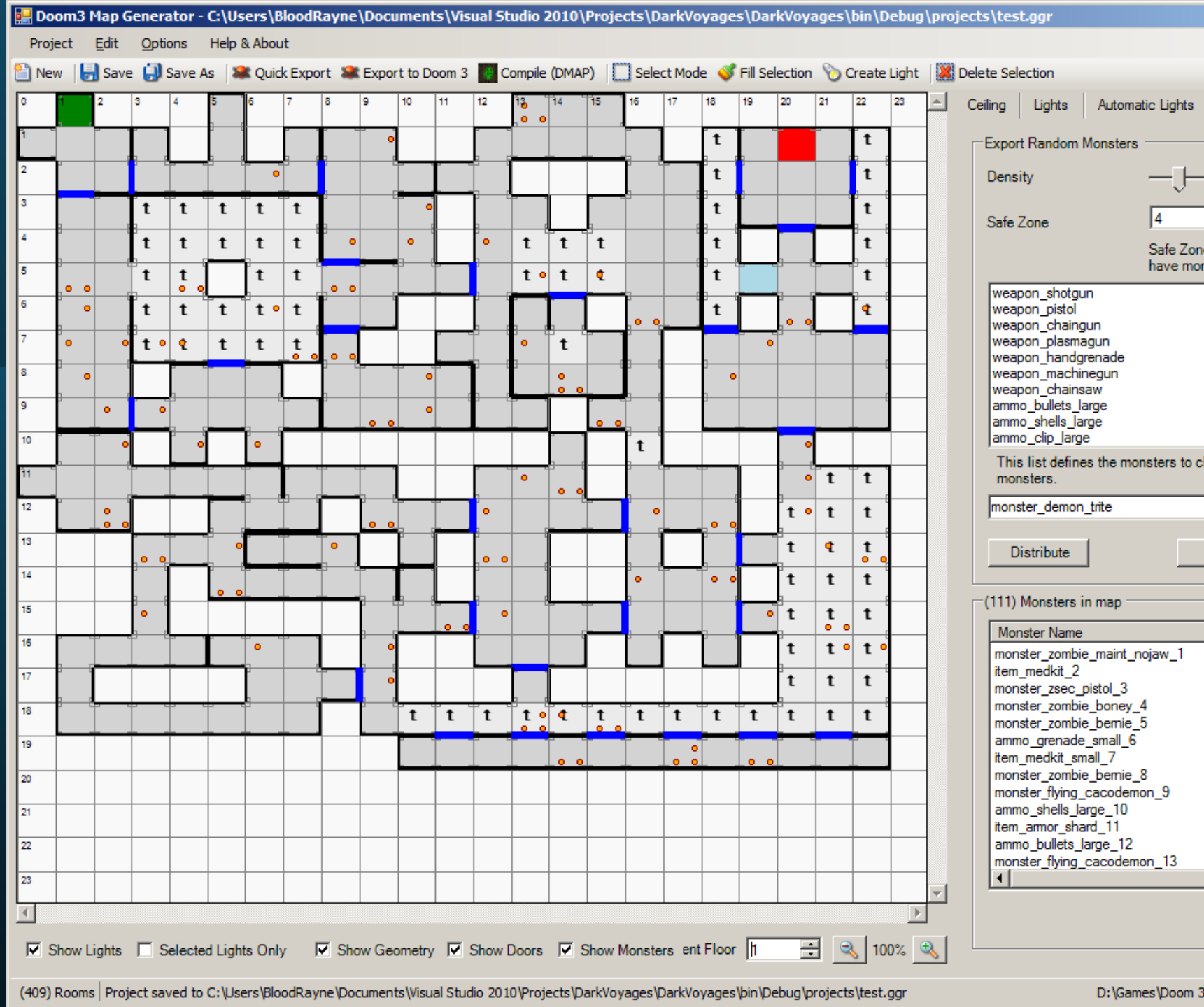
Demostración del proyecto

Explicación de la generación de mapas

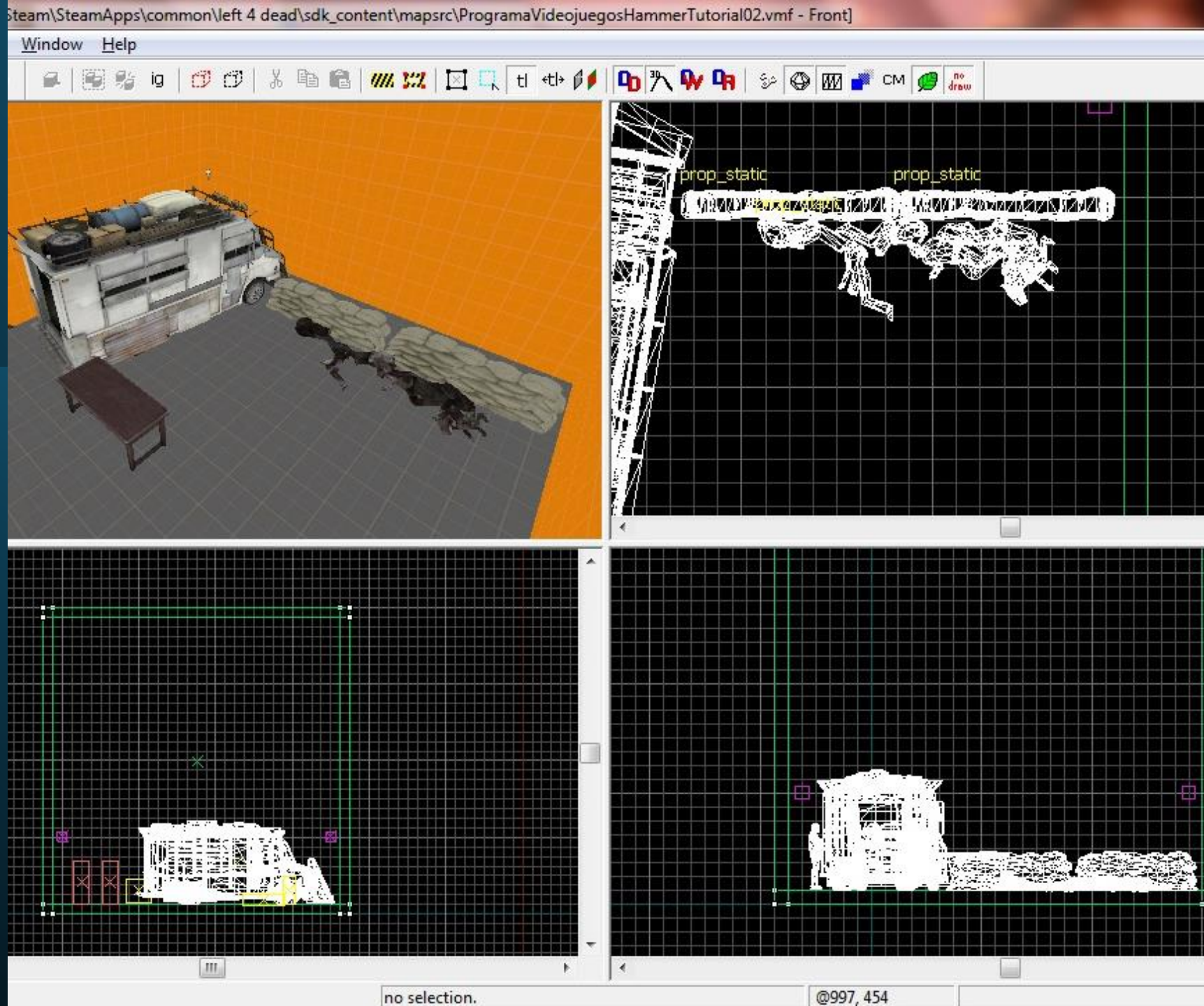
Explicación del raycasting

Análisis de posibles mejoras

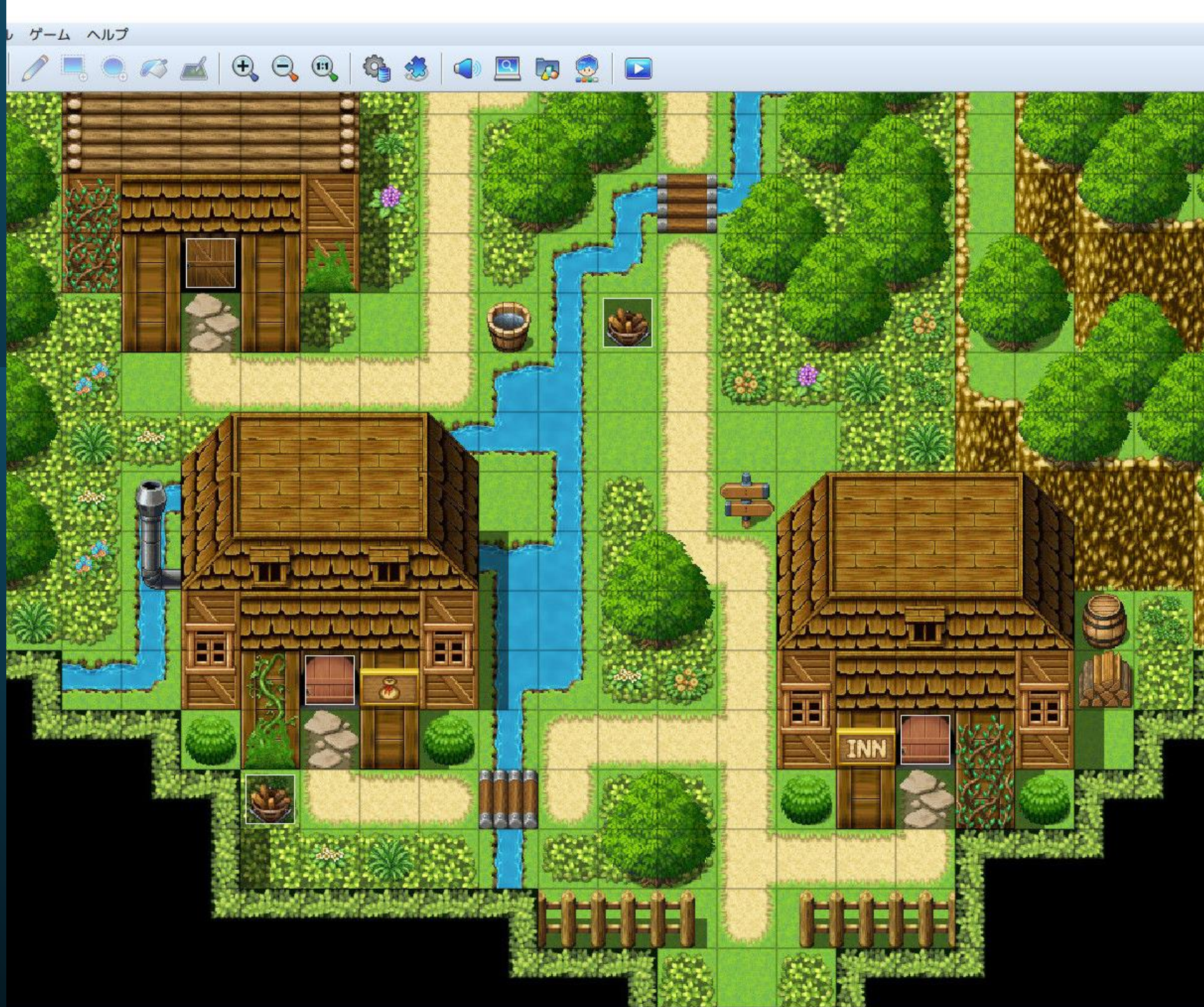
Explicación del proyecto



Explicación del proyecto

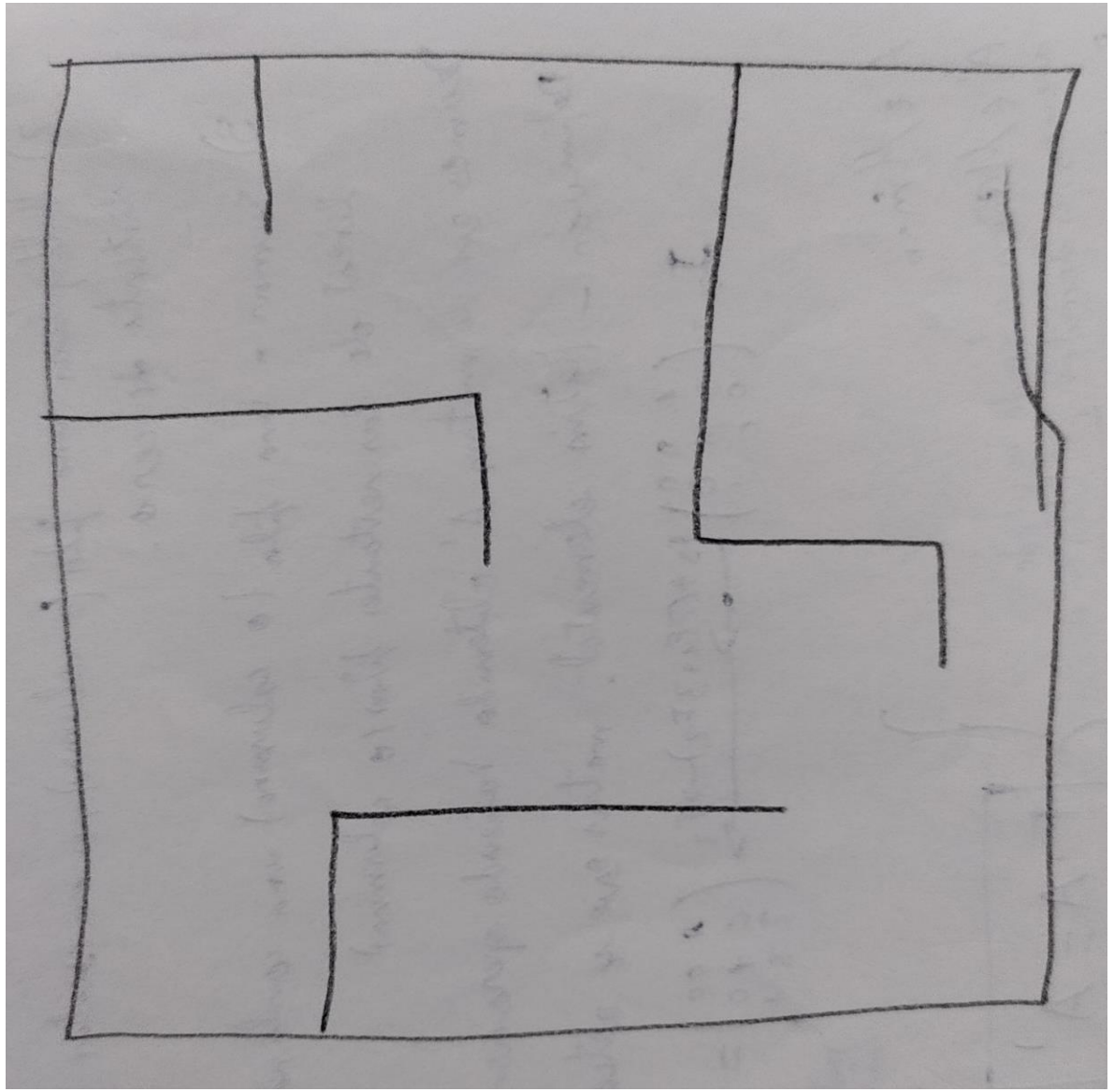


Explicación del proyecto

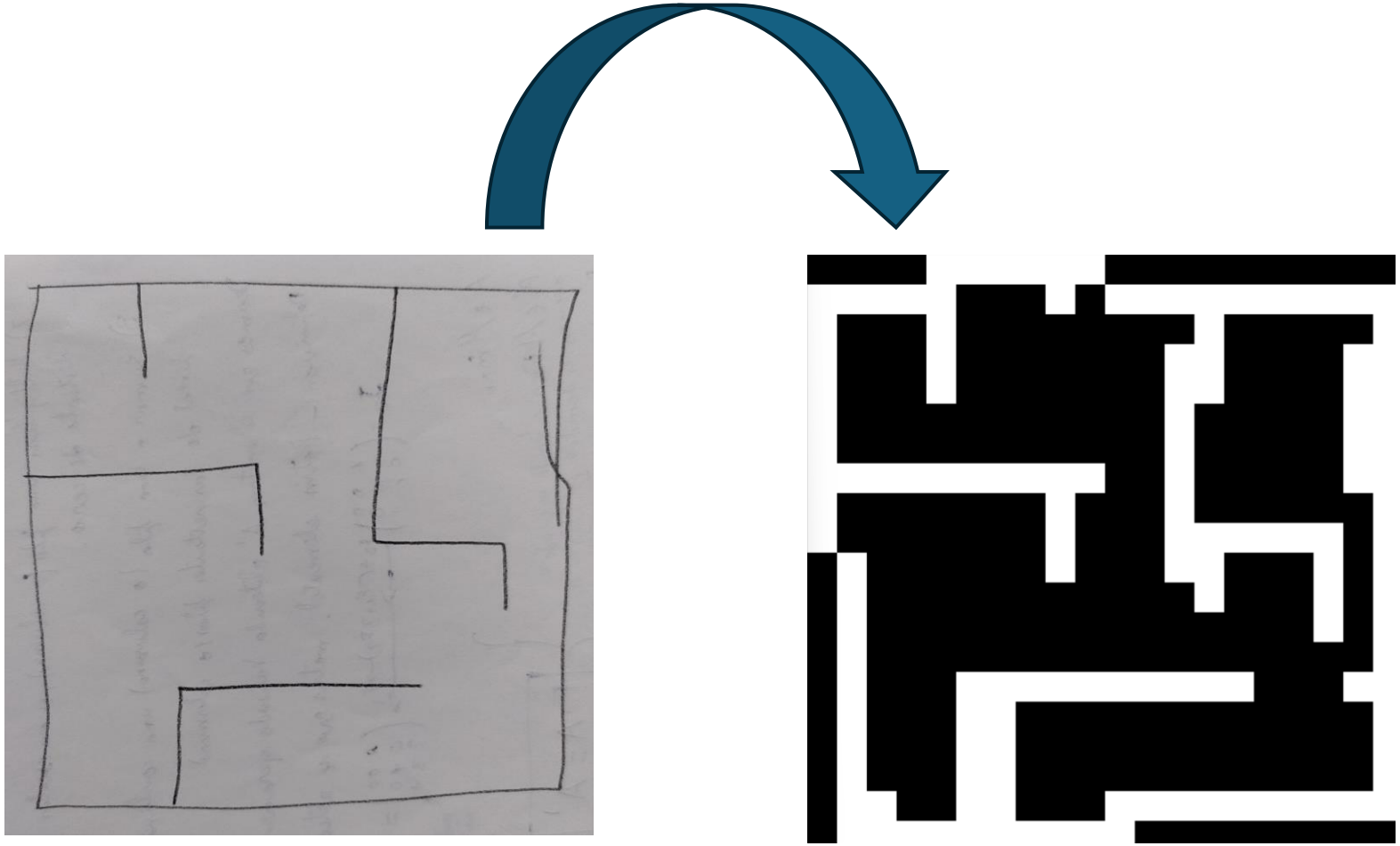


Demostración del proyecto

Generación de un mapa a partir de una imagen.



Explicación de la generación de mapas



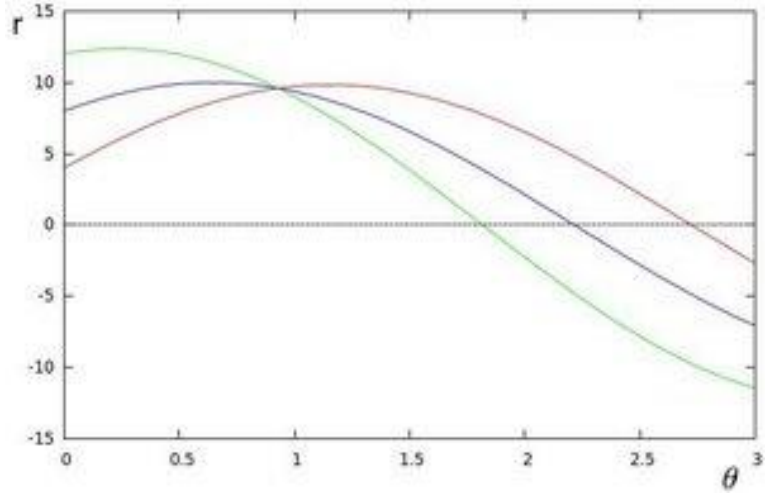
Explicación de la generación de mapas

1. Escalado de imagen
2. Creación de imagen umbral
3. Detección de líneas
4. Creación del mapa en una matriz
5. Escalado de la imagen de la matriz

Explicación de la generación de mapas

3. Detección de líneas

- Usando cv2.HoughLinesP()
- Basado en la transformada de Hough
- Devuelve las líneas detectadas.



Explicación de la generación de mapas

4. Creación del mapa en una matriz

- Creamos una matriz según el tamaño indicado
- Escalamos los puntos obtenidos de las líneas a la matriz
- Dibujamos líneas con cv2.line()

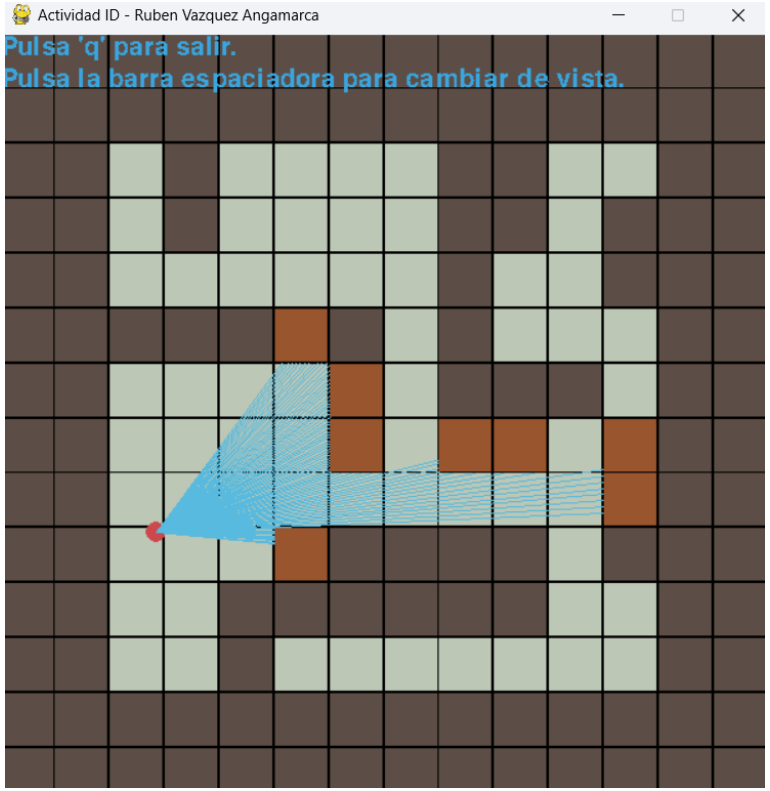
Explicación del raycasting

1. Ponemos al jugador en el mapa.
2. Lanzamos “rayos” desde el jugador hacia donde mira.
3. Dibujamos el entorno según la distancia de los rayos.

Explicación del raycasting

2. Lanzamos “rayos” desde el jugador hacia donde mira.

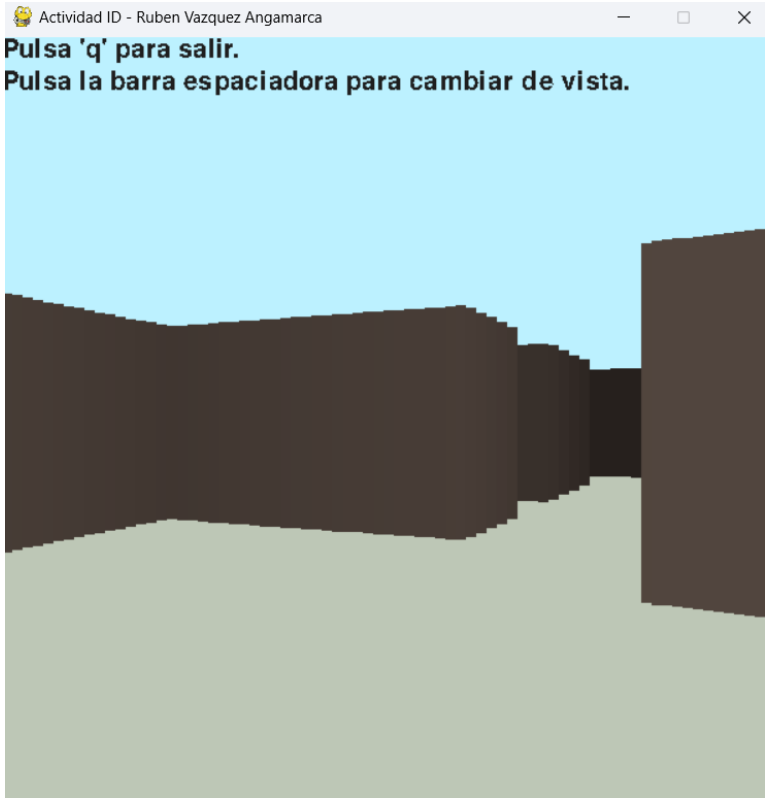
- El jugador mira en una dirección.
- Se lanzan rayos desde el jugador hasta que golpeen una pared.
- Se obtiene la distancia del jugador a la pared.



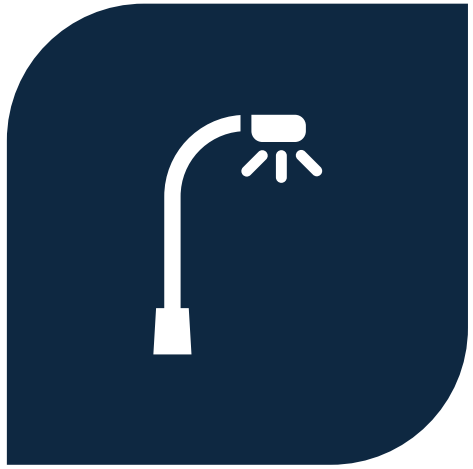
Explicación del raycasting

3. Dibujamos el entorno según la distancia de los rayos.

- Pintamos una pared donde haya golpeado el rayo.
- La pared será más o menos alta dependiendo de la distancia.



Análisis de posibles mejoras



DETECCIÓN DE LÍNEAS EN IMÁGENES CON LUCES Y SOMBRAS.

OPTIMIZACIÓN DEL ALGORITMO DE RAYCASTING.

FIN