

## 1. Arquitectura (lenguaje de programación, versión de CPCtelera...)

El lenguaje de programación que hemos utilizado para el desarrollo del videojuego es ensamblador z80.

La versión de CPCtelera que hemos utilizado ha sido la de la rama development con el último commit: **e2c6822**

Se puede obtener esta herramienta en el siguiente enlace:

- <https://github.com/Ironaldo/cpctelera/tree/development>

## 2. Problemas encontrados (mascaras, doble buffer, falta memoria...)

- Máscara incorrecta que impedía generar el archivo cdt
- Error al implementar el doble buffer y dibujar los obstáculos cada vez que se actualizaba el protagonista
- Problemas de memoria, que nos obligó a rechazar ciertos cambios y borrar niveles
- Las imágenes del menú y manual junto con el tilemap ocupaban demasiado comprimidas, por lo que tuvimos que prescindir de algunas pantallas y el tilemap lo sustituimos por un color de fondo gris.

## 3. Guiño al juego Chicago's 30

Utilizamos dos sprites de los personajes principales del juego Chicago's 30, uno es el del enemigo y otro el del protagonista:

- **Enemigo:** El enemigo de Chicago's 30 aparece como un enemigo especial a partir del nivel 9.



- **Protagonista:** El protagonista aparece como enemigo final en el último nivel, es decir, en el nivel 20.  
[INSERTAR CAPTURA AQUÍ TAMBIÉN]

## 4. Horas de trabajo dedicadas

Al principio, dedicamos 2 semanas para aprender lo básico del lenguaje y el entorno de trabajo, esto nos ocupó 8 horas semanales por cada miembro del equipo. Después de esas dos semanas, tuvimos otras 5 semanas en las que hemos seguido dos procesos paralelos: aprendizaje de la arquitectura y desarrollo del videojuego.

A continuación, se describe con más detalle las tareas empleadas en el desarrollo del videojuego:

1. En la primera semana decidimos las mecánicas del personaje principal, el movimiento, el teletransporte, la creación de los obstáculos de los mapas y las colisiones con ellos.
2. En la segunda y tercera semana implementamos en las mecánicas de los personajes, los disparos, agacharse y correr. También los enemigos, objetos, el cambio de vista de un mismo nivel y el paso al siguiente nivel.
3. Las dos últimas las dedicamos a la corrección de errores, como el borrado de la bala al colisionar con un enemigo o el doble buffer, y a la estética del juego, que al principio utilizamos tiles, pero como nos faltaba memoria tuvimos que rechazar la idea y utilizar pequeños sprites que se repetían sobre los obstáculos.