

Listado de correcciones resueltas para la entrega final

Hacer sí o sí:

- **Precisión en el disparo de la pistola:** Ahora para infligir daño al enemigo se debe apuntar al mismo, además, el daño cambia en función de la distancia con respecto al enemigo.
- **Recolectar armas + balas:** El problema estaba en que un jugador podía conectarse antes que el otro y agarrar items, entonces, al momento de conectar el segundo jugador se seguían viendo esos items, pero en realidad no deberían mostrarse. La solución fue crear una pantalla de espera y una vez que se conecten todos los usuarios para iniciar el juego(según el mapa la cant. de jugadores es distinta), estos podrán interactuar con el mismo.
- **Recolectar tesoros:** Sucedió lo mismo que con las armas y balas.
- **Pantalla de victoria/derrota(top 5) :** Al finalizar el juego(ya sea porque murieron los demás jugadores o se desconectaron), se muestran las estadísticas finales del juego del top 5 de mejores jugadores.
- **Manual de usuario:**
Manual del Proyecto:
Documentación técnica: Se encuentran en la carpeta "Docs" del repositorio.

- **Uso de CPU(100% Client):** Primero se observó que el tiempo de update del modelo en el peor de los casos era de 0.05ms, mientras que el renderizado tardaba como mínimo 80ms. Es decir, casi el 100% del tiempo se empleaba en el dibujado, por lo que se buscó ver en qué parte del renderizado se empleaba tanta cantidad de tiempo.

Se encontró que en la clase Panel_window, dentro de la iteración de un for, correspondiente a la creación de clases Ray_Panel(para el renderizado de las paredes), se estaba pasando como parámetro el mapa entero por cada iteración;esto causaba que el $\approx 30\%$ del tiempo del renderizado se empleará en dicha función. Al ver el contenido de dicha clase, se vio que no se usaba el mapa para nada, por lo que se eliminó dicho parámetro. Luego de su eliminación el tiempo empleado en dicho for fue alrededor de un 5%.

Igualmente, el tiempo de renderizado sigue siendo alto, y esto se debe exclusivamente al renderizado de todos los objetos del mapa al desencolarlos en la función update de la clase Panel_Window. Se consultó al tercer

compañero del grupo preguntando si se podían hacer un par de modificaciones para mejorar la performance, pero nos comentó que hacer modificaciones en dicha clase implicaría un cambio drástico en la lógica del ray casting y que esto conllevaría a romper por completo el modelo actual.

Otro ajuste que se hizo para reducir el tiempo de CPU, fue el agregado del Constant Rate Loop, usando como parámetro los 100 ms se redujo el % de uso de CPU de un 25% a un 15%. Subir el "constant_rate_loop_render_ms" a valores mayores de 250ms, ya generan que el juego ande lento para el ojo humano, pero subir los ms con respecto a los 100 ms reducen notoriamente el uso de CPU.

También, una solución alternativa para las computadoras de bajos recursos es bajarle la resolución al cliente(520x320 o 320x200 por ejemplo). Estos dos parámetros previamente mencionados se pueden cambiar en el yaml que se encuentra en `yaml_configs/client_settings`.

Al probar el juego con resolución de 520x320 y 220ms se logró alcanzar un uso de cpu del 6%, en lugar del 25% constante que se tuvo en la primera entrega en nuestras computadoras.

Hacer si se puede:

Drop por muerte: Al morir, se dropea un cartucho de balas y el arma especial equipada, si es que se tiene una en uso.

Finalización por tiempo: Después de que transcurran 10 minutos de juego, los jugadores empezaran a sufrir un daño de 20 de vida cada 10 segundos. Esto causa una especie de muerte súbita que hará que el juego finalice transcurrido cierto tiempo. Ambos tiempos pueden ser modificados en el archivo de configuración del servidor.

Click and drag en editor: Se puede hacer seleccionando el ítem con click derecho y luego manteniendo el click derecho y moviendo el mouse por donde se quiere ubicar el objeto

Comentarios:

- **Leaks en el servidor al jugar una partida:** Ya se liberó dicha memoria y no hay más leaks
- **Socket operador= :** Se cambió su implementación

- **Caída de ambos clientes en algunos casos en que un cliente se retirara:** Se corrigieron esos casos.
- **Tampoco hay animación al moverse:** Se agregó una animación básica de movimiento para los 3 personajes distintos
- **Error por el linkeo de librería dinámica:** Se vio que su uso era innecesario, se dejó de usar.
- **Prints de debug:** Se borraron dichos prints
- **Cambio de ubicación de controles:** Se cambiaron por los recomendados en la clase anterior a la entrega final

Opcional:

- **Precisión en ametralladora:** Ahora se necesita apuntar al objetivo para poder dañarlo
- **Precisión en cañón de cadena:** Ahora se necesita apuntar al objetivo para poder dañarlo
- **Instalador:** Logramos hacer un instalador pero por falta de tiempo no logramos hacer que las carpetas se instalen correctamente en el path establecido. Por este motivo decidimos que se utilice el instalador pero que para correr el servidor/cliente/editor sea necesario estar en la carpeta build y correrlo dentro de la misma.
- **Música:** Se corrigió un error por el cual no se tenía en cuenta el volumen dado por el archivo de configuración del cliente, por defecto la música se escuchara, el volumen puede ser cambiado por el archivo de configuración