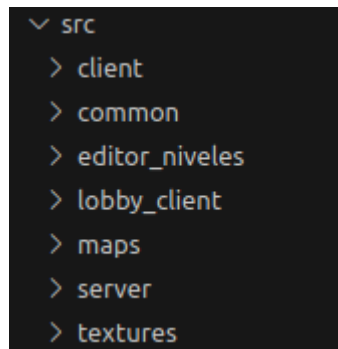


Documentación técnica

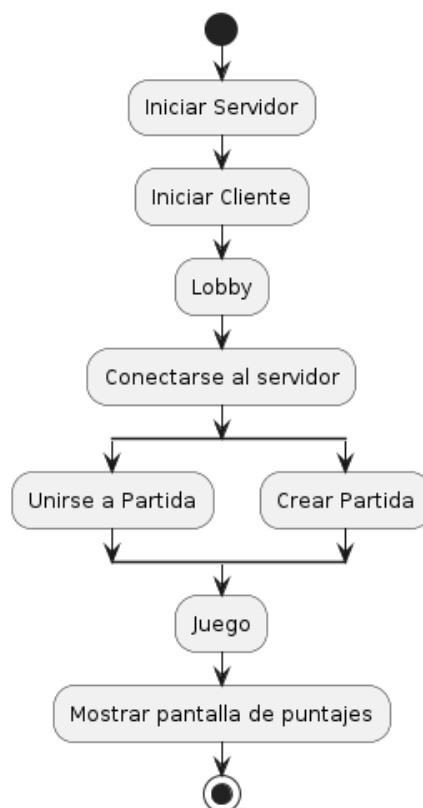
Estructura del proyecto

En la carpeta común del proyecto se encuentran los archivos: config.yaml, Readme.md, CMakeLists.txt, .gitignore y la carpeta src:



Dentro de esta carpeta se encuentra la lógica del juego y sus distintos componentes.

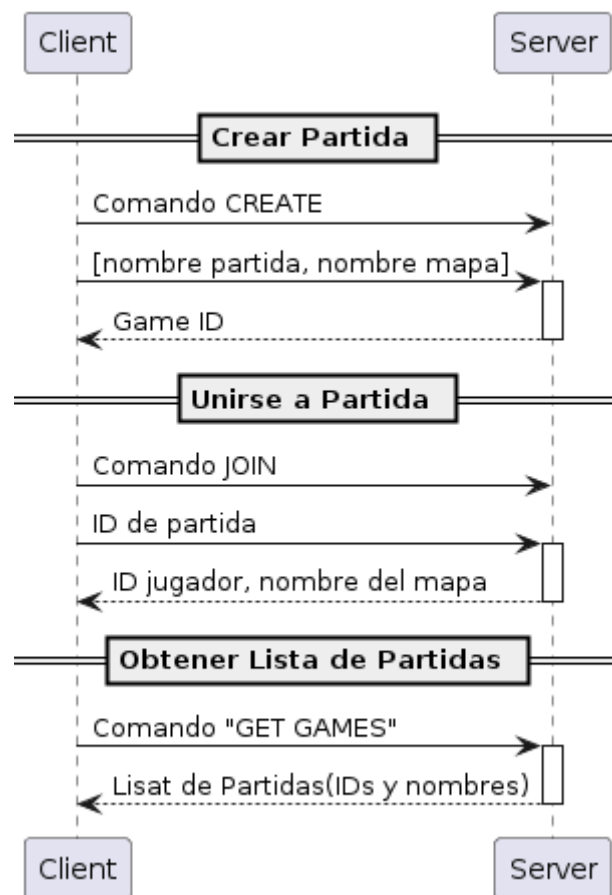
El flujo general del juego resulta ser algo así:



También, antes de ejecutar al cliente se puede acceder al editor de niveles, para jugar con un mapa customizado.

Protocolos

Lobby

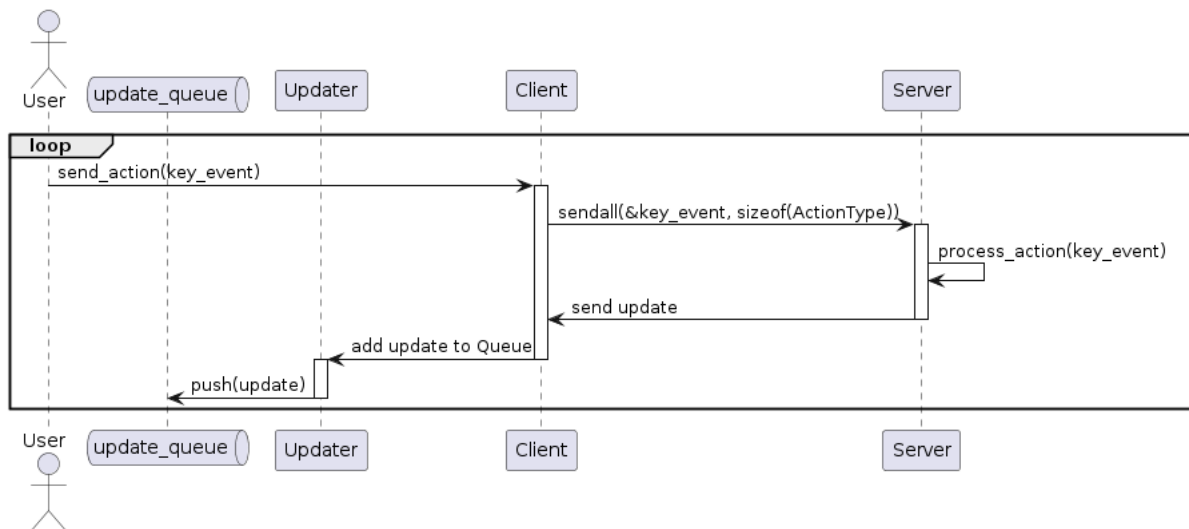


Los comandos que envía el cliente: CREATE, JOIN, GET GAMES, son de tipo uint8_t. Cuando el cliente envía CREATE, el servidor lo procesa y espera el nombre de la partida y el nombre del mapa a utilizar. Estos datos los utiliza el lobby manager para la creación de la partida. Y retorna el id de la partida creada de tipo uint8_t, y se lo manda devuelta al cliente.

Cuando el cliente envía JOIN, el servidor espera el id de la partida a la que se quiere unir, se agrega el jugador a dicha partida en el lobby manager. Retorna y envía al cliente el id del jugador y el nombre del mapa.

Cuando el cliente envía GET GAMES, el servidor le responde con una lista de partidas disponibles.

Updates



En breves palabras, el usuario manda una acción, en este caso

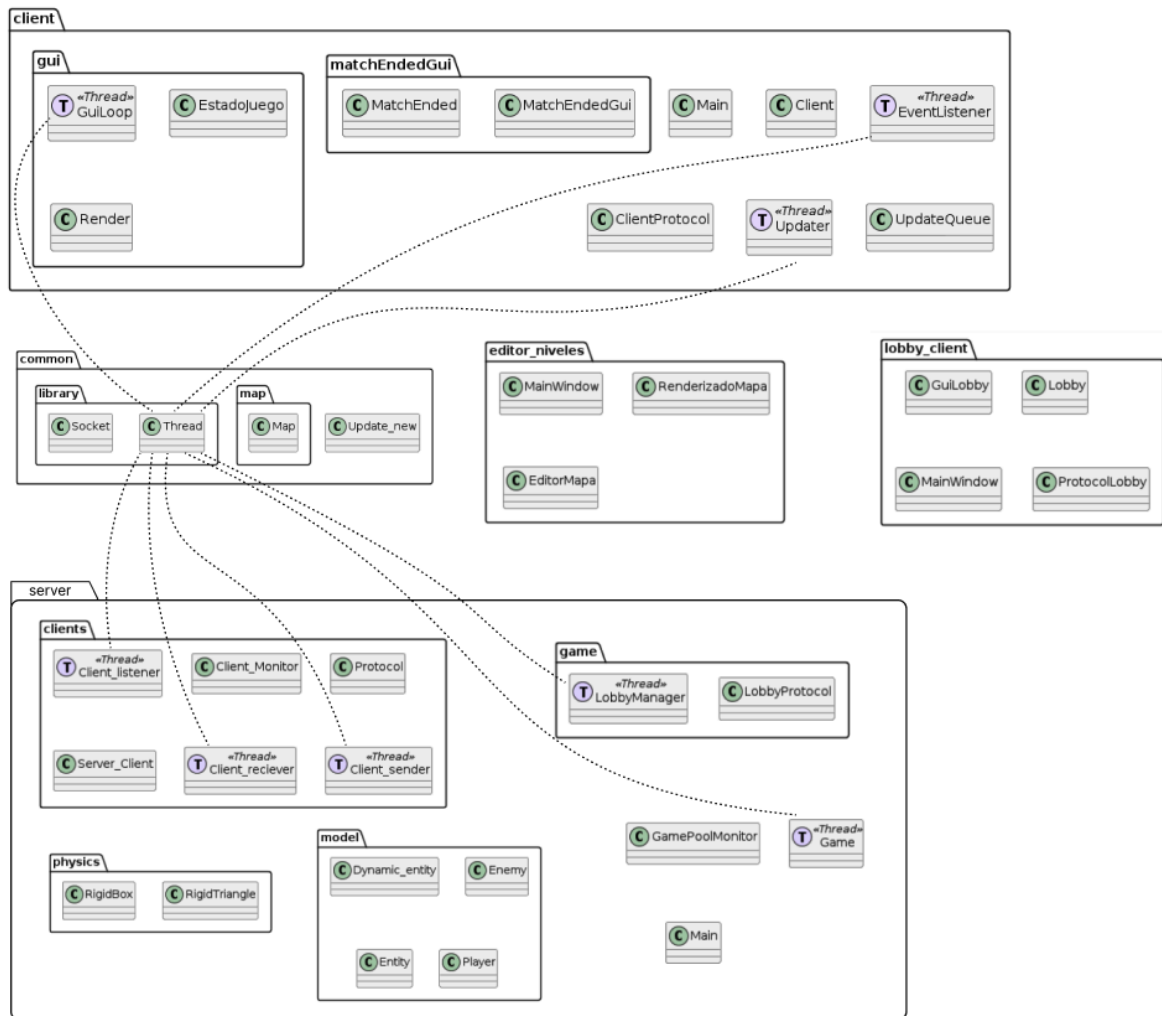
Las Updates están modeladas de la siguiente manera:

Existen cuatro tipos de updates, en todo los casos se recibe/envía una modificación en el estado de algún elemento del juego.

- Create entity: se crea la entidad.
- Delete entity: se elimina una entidad.
- Position: se actualiza la posición de algún elemento.
- Value: Se actualiza algún valor de un elemento en el juego.

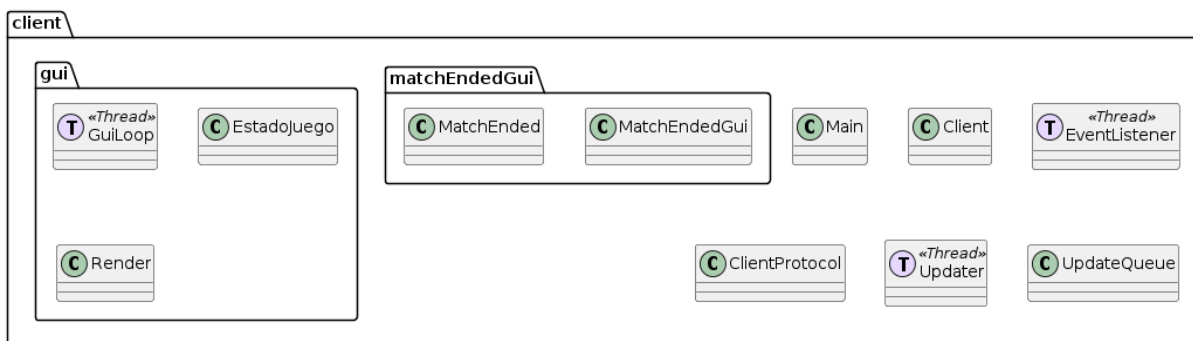
Clases

Diagrama general de clases:



Client

Estas son las clases presentes en el cliente.



Las clases que dicen thread, heredan de la clase Thread presente en library.

GUI

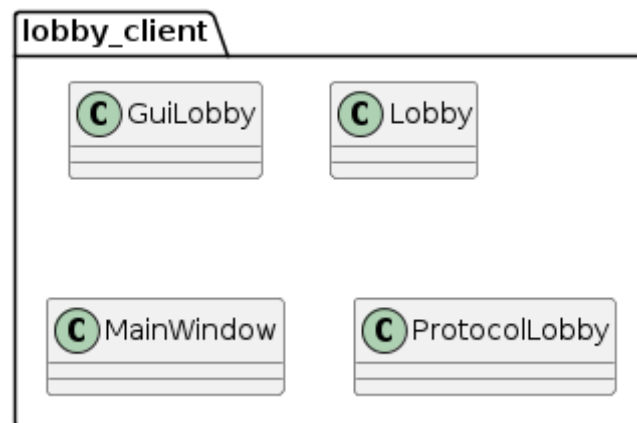
En esta carpeta se encuentran los archivos vinculados al módulo de la interfaz gráfica.

Match Ended Gui

En esta carpeta se encuentran los archivos del módulo de fin de partida, donde se muestran los puntajes de los jugadores. Este módulo fue realizado con QT.

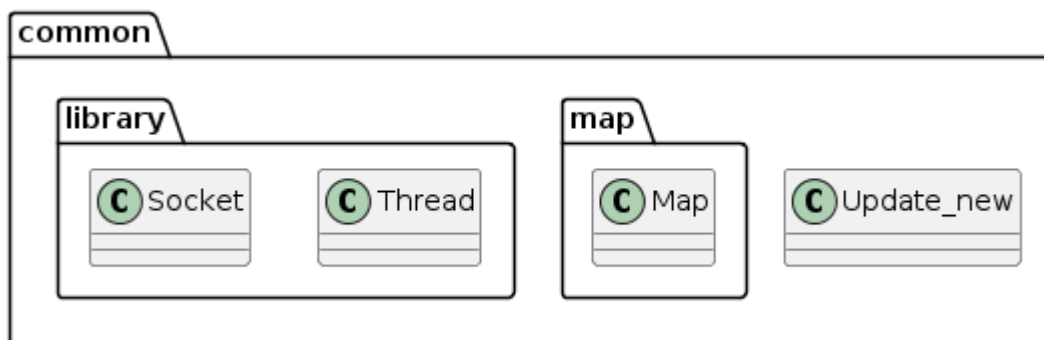
Lobby-client

El módulo del lobby desde el lado del cliente. Está compuesto por estas cuatro clases, que involucran la lógica para mantener la comunicación con el server. También fue realizado en QT



Common

Dentro de esta carpeta se encuentran los archivos utilizados por el servidor y el cliente.



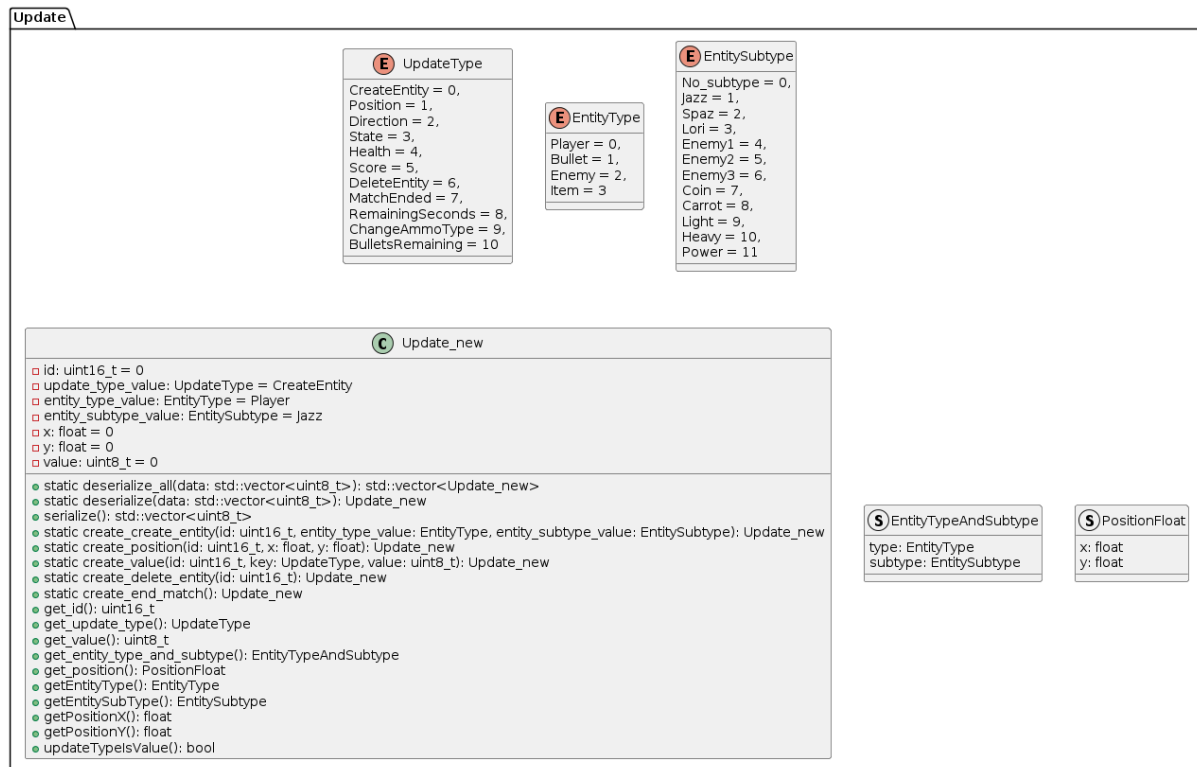
Library

Esta carpeta contiene las librerías de socket y thread utilizadas a lo largo del proyecto.

Map

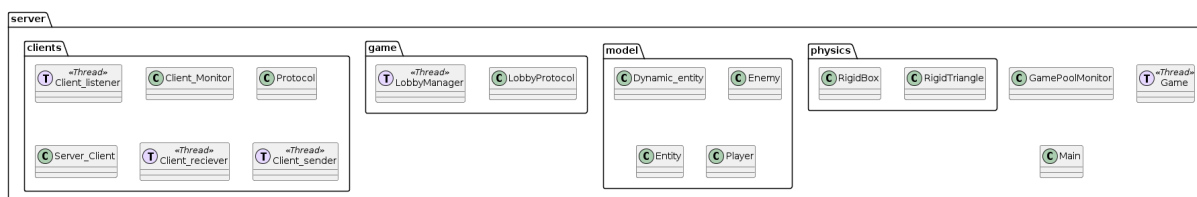
Módulo de mapas, su función es de leer los archivos yaml de los distintos mapas disponibles en el juego. Al mismo tiempo, el editor de niveles convierte la información que se carga en el visualizador, y la traduce a un archivo yaml.

Update: La clase update es moldeable según el tipo de update que se quiera mandar.

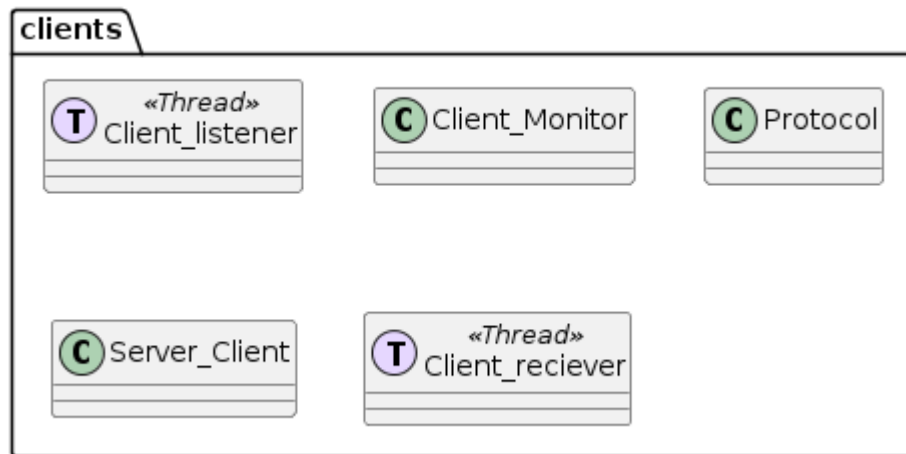


Server

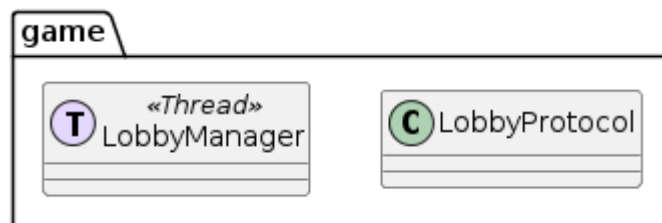
Diagrama del server



Clients



Game



En esta carpeta se encuentran los archivos relacionados a lobby, se encuentra el lobby manager, el protocolo que recibe y envía información a través del socket de la siguiente manera.

Y el lobby manager se encarga de orquestar las partidas, según el cliente pida.

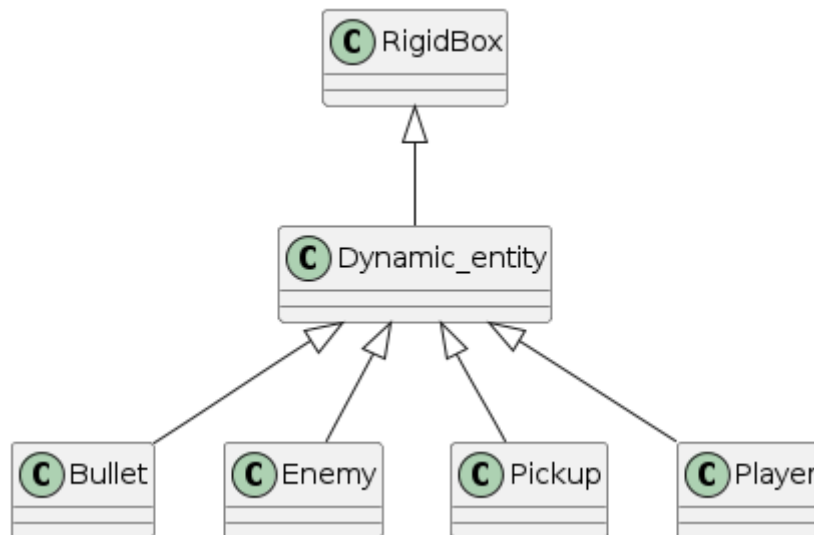
Loader

En esta carpeta se encuentra el módulo para la configuración, está la clase Config, que se encarga de la lectura del archivo **config.yaml** para la lógica del juego.

Model

Archivos con las estructuras y funcionamiento de las entidades principales del juego.

Dentro de esta carpeta hay una subcarpeta que contiene constantes de los distintos tipos de ítems, estados de juego y tipos de municiones.



Physics

Protocolo de física. En esta carpeta se encuentra la implementación de Rigid Box.

Editor de niveles

El módulo del editor de niveles consta de estas tres clases. Está realizado con la librería QT.



En la carpeta **maps**, se encuentran los yaml con los mapas ya implementados. Y en la carpeta **textures**, se encuentran las imágenes de las distintas texturas que son utilizadas para los mapas y los sonidos del juego.

Otros archivos importantes:

config.yaml: este es un archivo yaml que guarda la información de configuración del juego, modificable a gusto.