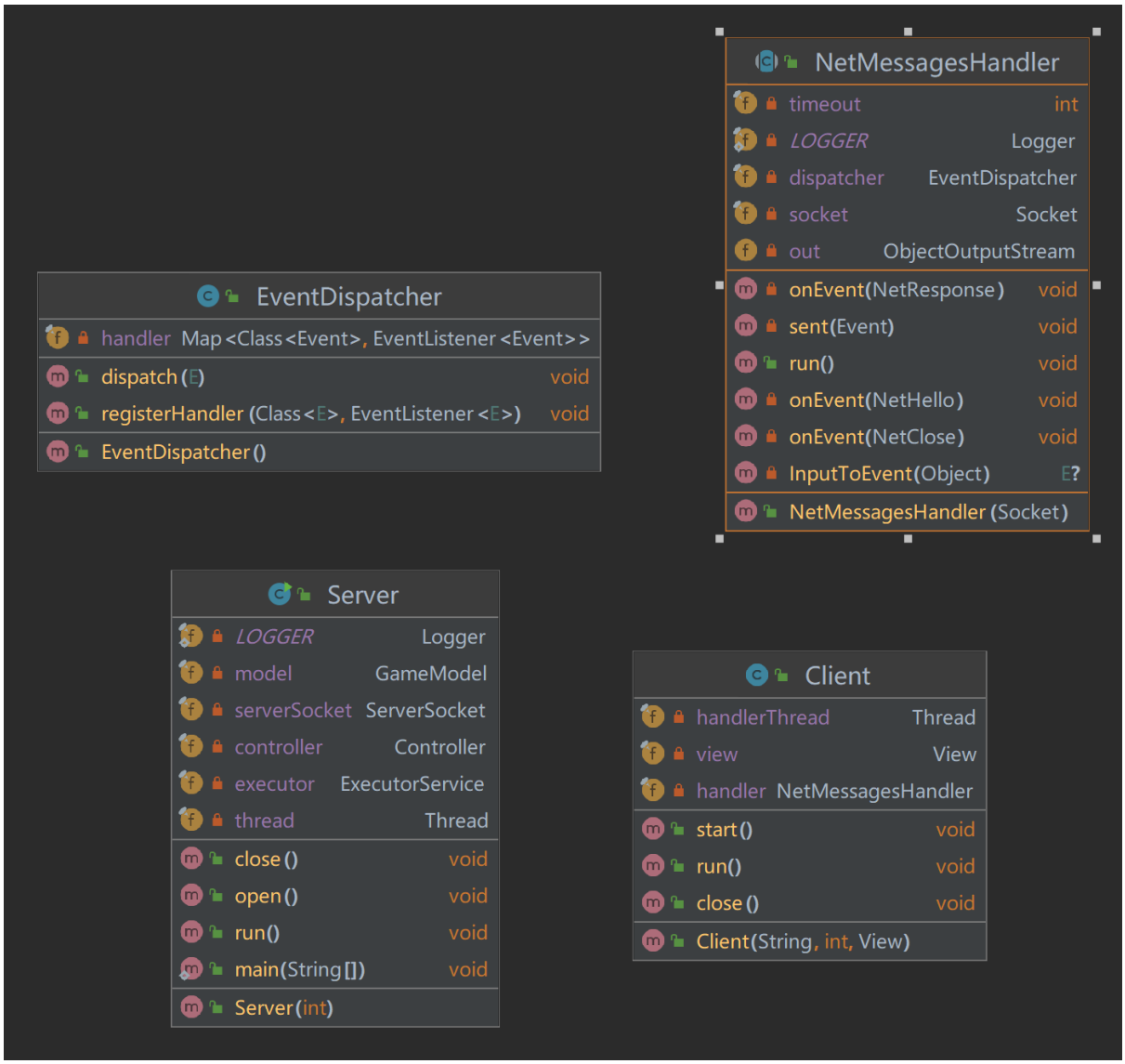
# To be reviewed




Descrizone:

SuperClass EventDispatcher: entita' progettato per associare event a suo Handler(un functiuonal interface) . con funzione dipatch chiama la funzione Handler ad esso associato.

NB : tutti le classe che richede invio di dati sul reti sono subClass di EventDispatcher in quanto associa ad ogni event che richede invio di dati associa la funzione sent di MessageHandler.

netMessageHandler: classe gestore dei messagi in rete durante la sua runtime si mette in ascorto e contiene a se tutti i Handler di messaggi in arrivo(Handler definiti

in controller oppure in model)

NB:Maggior parte dei evnti che vengono inviati sono associati al Handler che coincide con il sent() del netMessaggeHandler. e handler dei messaggi ricevuti invece sul controller.

Server : la classe server che hosta il model e controller si occuppa di assocciare i metodi definiti in controller al messageHandeler (non e' definito in UML in quanto istanziato nel Runtime)

NB:La richesta di modifica al View avviene attaverso invio di un istanza di Subclasse di Evento con messageHandler.e ricezione di un evento viene fatto dispatch e invocato il metodo assiociato a quel evento(metodo definito in controller).

Generalmente questi metodi modifica o richede dati del model.

Client:la classe che hosta il view del giocatore e ha funzione simile al Server con invio e ricezione istanze di SubClasse di eventi.

NB : la richesta di modifca del model avviene simiralmente al sever attraverso invio di un istanza di Subclasse di Evento con messageHandler.

PS. nel UML mancano delle classe e colegamenti .Anche se UML non e' fatta da me la parte di codice ho implemtato io e se possibile vi mandero' un UML piu' chiaro e completo.Scusate per il ritardo portato e grazie per la vosta complensione.