Linee Guida per l'implemetazione di View

View

- nel package view (d'ora in poi omesso) in asset/game ci sono da implementare metodi e classi per una gestione minimale del Client Questo significa :
    - nessuna classe ha metodi per svolgere funzioni complesse come calcoli(a meno di strette esigenze)
    - ogni classe in generale implenta metodi per aggiornare in maniera totale la classe stessa e per disegnare la classe stessa
      In generale quindi c'è un metodo update che aggiorna tutti i valori della classe stessa(pulendo quelli vecchi) e più metodi draw che ritornano gli attributi della classe

Importante tener fede ai messaggi che si inviano dal server -> se una cosa non viene inviata dal server ma va in qualche modo aggiornata allora va aggiunto qualcosa nel server

- nel package asset/exception vanno inserite le eccezzioni che eventialmente servono in caso di errore nel micro-Model che ogni giocatore(lato client quindi) ha.
- package cli e gui da non toccare per ora

ClientController

nel package controller/client (d'ora in poi omesso) si trova il controller del client\

- nel package client/game sono presenti in maniera analoga al controller le fasi di gioco (sono ancora da implementare)
- nel package client/networkHandler e presente un'interfaccia network e dovranno essere sviluppati i messaggi in maniera pseudo-speculare al servercontroller

RoadMap

Tutte le scadenze sono fissate a Martedì 17

Luca	massima priorità a finire le classi in viev/asset ->controllo di flusso -> portarsi a buon punto con le fasi di gioco controller/game
Sara	revisione flusso totale client-controller(con documento che vi propongo domani, **solo a livello di messaggi*, ovvero controllare se ci sono tutti e basta) -> cercare di finire tutte le classi in client/networkHandler (specularmente come in controller/server/virtualview)
Davide	finire test in controller/networking -> sistemazione lato model -> inizio cli