Diagrama de Clases

Aquí vamos a mostrar los diagramas de clase, primero separados en los tres subsistemas principales: Controlador, modelo y Vista; y luego mostrando la relación entre las principales clases de cada uno de estos.

**Subsistema controlador de juego** - *(Logical diagram)*



*Figure: 4*

**creacion y unirse** - *(Sequence diagram)*



*Figure: 5*

Aquí podemos ver como es la interaccion necesaria para la creación y para unirse a partidas.

**Subsistema interfaz de juego** - *(Logical diagram)*



*Figure: 12*

**Subsistema interfaz de juego** - *(Sequence diagram)*



*Figure: 13*

**interface2** - *(Sequence diagram)*



*Figure: 14*

**modelo** - *(Logical diagram)*



*Figure: 15*



Como podemos observer, utilizamos el patron observer en este diagrama, entre la clase ControladorCambios (del subsistema controlador) y la clase Observer (de interfaz de juego), con el fin de actualizar el mapa(con todos sus elementos) para que pueda ser mostrado en pantalla por todos los usuarios que estén jugando en esta etapa.

Utilizamos este patrón ya que veíamos que en nuestro sistema existía la posibilidad de que el servidor tuviera que actualizar el mapa de la ciudad a varios clientes, y el patrón Observer hace mucho mas fácil esta tarea: simplemente se suscribe al usuario correspondiente a la lista que tiene Controlador Cambios. Y el método notify de este se encarga de ejecutar el método update correspondiente a cada objeto de la clase Observer(correspondiente a cada jugador). En caso que cierto jugador haya terminado la partida, se lo retira de la lista y deja de recibir actualizaciones del mapa.