Trabajo Práctico
Organizador de Futbol 5

Entrega Nº 4

Materia: Diseño de Sistemas
Profesor: Nicolas Passerini
Ayudante: Gisela Decuzzi
Alumnos:

- **Acosta Naiara**
- **Brandoni Agustin**
- **Coiro Tomas**
- **Leder Brian**
- **Luis Ostiglia**

2014

Punto 1

Para el desarrollo de ordenamiento de equipo se decidió implementar un Strategy Stateless. El partido no conoce el Strategy (estrategia), la misma se recibe por parámetro al llamar al método partidoOrdenaJugadores(Ordenamiento criterio), por este motivo es Stateless. La decisión que sea Stateless se debe a que si fuese Statefull como el Administrador(Actor) puede modificar varias veces el ordenamiento hasta estar conforme, cada vez que el Administrador deseara ordenar el partido, se tendría que modificar previamente el atributo de ordenamiento del partido, enviandolo por parámetro evitamos esta situación que en este caso notamos innecesaria que se produzca reiteradas veces.
Siendo un Strategy se logró la posibilidad de aplicar diferentes criterios, utilizando el polimorfismo como herramienta fundamental para la implementación del strategy. Cada criterio de ordenamiento (handicap, promedio de calificaciones del último partido, promedio de cierta cantidad de calificaciones, mix de ordenamientos, etc), entienden el mismo mensaje, ordenar(Jugador jugador), en cambio si no se hubieran manejado estos criterios de modo polimórfico la solución se tornaba menos extensible para nuevos criterios de ordenamiento en comparación con la solución elegida. El Strategy staless, en comparación a una solución sin utilizar un strategy y en definitiva sin utilizar mensajes polimorficos, aumenta la cohesión ya que la estrategia sólo determina el valor del criterio luego de aplicarlo al jugador, y el partido se encarga de ordenar su propia lista.

Para el requerimiento de división de equipos se prefirió utilizar un command, ya que se tenían ciertas acciones (modos de división de equipos) que producen un efecto en el partido, es decir modifican el mismo.
Se prefirió un command por sobre el strategy ya que si se hubiese implementado un strategy cada estrategia hubiese tenido que solicitarle al partido que modifique su lista, pasando a ser un pasa manos de metodos ya que al partido se le iba a solicitar que se divida, al modo de división se le iba a solicitar que divida, y este le iba a solicitar al partido que se divida según sea el modo. Por lo que con la solución implementada, cada comando o modo de división de equipos tiene efecto directamente sobre el partido. En este caso el partido conoce los command que debe ejecutar, ya que si observamos lo planteado en el libro “Patrones de Diseño” de Gamma sobre la estructura tentativa de Command el partido sería el invoker y el command (modalidad de division de equipo) conoce al partido ya que debe saber en donde aplicar la acción. En decir la relación Partido-Command se debe a que el partido invoca al comando, y la relación comando-Partido se debe a que el comando conoce a donde debe impactar.

Por último cabe mencionar que se decidió que el partido entienda los métodos partidoOrdenaJugadores(Ordenamiento criterio) y partidoDividiEquipos() y que no sea el administrador el que entienda estos mensajes, como da a entender el enunciado, ya que se entiende al administrador del enunciado como el Actor y no como la clase. Si se hubiese implementado que el administrador entienda los mensajes mencionados, la clase Administrador pasaría a ser lo denomina “Clase Gerente” donde lo único que realiza es solicitarle a otras clases procesar un método