



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico I

Curry Game

Ingeniería del Software 2

Integrante	LU	Correo electrónico
Aleman, Damián Eliel		damianealeman@gmail.com
Arjovsky, Martín		martinarjovsky@gmail.com
Fixman, Martín		martinfixman@gmail.com
Harari, Ignacio		nachotee@hotmail.com

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

User Stories

Las User Stories mostradas en **negrita** son las serán completadas en el primer Sprint.

Descripción	Esfuerzo	Value
Como participante quiero poder registrar una cuenta para acceder al juego.	2	8
Como participante quiero buscar desafíos para poder aceptarlos.	5	5
Como participante quiero armar equipos para poder aceptar o postular desafíos con ese equipo.	3	5
Como administrador quiero administrar jugadores disponibles para poder organizar el juego.	3	2
Como administrador quiero consultar las cuentas que están en el sistema para poder analizar el uso del juego por los participantes.	5	2
Como participante quiero autenticar mi cuenta (log in) para poder entrar en el sistema de juego.	2	8
Como participante quiero poder ver mi cap y mis fichas para decidir mis apuestas.	2	3
Como participante quiero ver tabla de posiciones de todos los participantes en base a sus desafíos ganados/perdidos para poder conocer el ranking.	3	2
Como participante quiero postular desafío para desafiar a otros participantes a que jueguen contra mi equipo.	2	5
Como participante quiero aceptar el desafío postulado por otro participante para poder jugar contra otros equipos.	2	5
Como administrador quiero poder crear jugadas ofensivas y defensivas.	5	3
Como administrador quiero poder simular una acción ofensiva y defensiva.	5	8
Como administrador quiero editar fórmulas y números de resolución de acciones para poder cambiar el juego en caso de que se necesite.	5	3
Como participante quiero ver log del partido e información pertinente para poder visualizar las simulaciones anteriores.	3	8
Como participante quiero nombrar mi equipo, elegir mi jugador estrella y mi tecnico.	2	5

1. Explicación de los diagramas

1.1. Estrategias Jugadas y Acciones

La representación específica para las jugadas ofensivas y defensivas son ligeramente distintas entre sí, si bien a grandes rasgos parecerían ser similares.

En el caso de la Ofensa, tenemos tres niveles definidos: *EstrategiaOfensiva*, *Jugada Ofensiva* (existe como concepto pero no es una clase concreta en la aplicación), y *AccionOfensiva*. Comenzando por lo más atómico, una *AccionOfensiva* es simplemente la acción ofensiva más atómica que puede realizarse, en nuestro dominio sería un Pase o un Tiro. Debido a que sabemos que las Jugadas Ofensivas siempre deben (lógicamente) terminar en un tiro, decidimos modelar a las Acciones Ofensivas utilizando el partón de *Decorator*, de modo que podemos decorar un Tiro (ya sea de 2 o 3 puntos) con la cantidad de Pases que querramos (esto también debería permitirnos en el futuro agregar otro tipo de acciones ofensivas para decorar una jugada que finalmente siempre culminará en un tiro) éste tiro decorado representa lo que nosotros llamamos una **Jugada Ofensiva**. Finalmente, la *EstrategiaOfensiva* es la encargada de crear las acciones ofensivas, armar el decorator y devolver la Jugada Ofensiva, dependiendo de las características específicas que tenga la estrategia que estemos queriendo modelar, por ejemplo si quisiéramos modelar una Colectiva Externa de 3 puntos luego de 2 Pases, sería ésta clase la encargada de generar una Jugada Ofensiva que conste de dos Pases y finalmente un Tiro3Puntos.

En el caso de la Defensa, también tenemos una clase *AccionDefensiva*, que son las acciones defensivas atómicas (Intercepción, Bloqueo, etc), pero interpretamos a la *EstrategiaDefensiva* de una manera distinta. Una *EstrategiaDefensiva*, tendrá definido un método para cada *AccionOfensiva* que exista en el sistema, y al intentar realizarse una *Accion Ofensiva*, utilizando *Double Dispatch* se encargará de devolver una lista *Acciones Defensivas* correspondientes a la Ofensiva, que representarán a los jugadores encargados de defender contra la *Accion Ofensiva* solicitada, según cómo esté definido el algoritmo de la *EstrategiaDefensiva*, por ejemplo, un *HombreAHombre* responderá un Pase del Base con una Intercepcion del Base contrario.

1.2. Logger y Twitter

Ambas funcionalidades se manejan de manera casi idéntica para la mayor parte de las acciones, por lo cual para ayudar a la claridad de los diagramas de secuencia, fueron incluidas sólo en algunos de ellos, y omitidas en el resto, es fácil extrapolar desde los ejemplos en los cuales están incluidas cómo funcionarían en el resto de las jugadas.

En el caso del logger, cada accion al calcular su resultado (es decir, cuando se calcula si triunfa o no) se encarga de crear un logger y llamar a la funcion del logger correspondiente a la acción particular que se esté realizando, por ejemplo, un pase llamaría a la función del logger `loguearPase`, dentro de la cual, el logger sabrá interpretar la accion pasada para escribir un String descriptivo del resultado de la simulación de esa acción.

Para las estadísticas de twitter, tenemos un objeto llamado `ScraperEstadísticasTwitter` que dado un jugador, se encarga de conseguir las estadísticas de sus tweets e interpretarlas, para devolver entonces el modificador de la acción en el momento que se requiere calcular el triunfo.

1.3. Pase

Tanto Pase como ambos Tiros heredan de una misma clase `AcciónOfensiva` siguiendo el partón de Decorator, esto nos permite simular el pase actual uno a la vez, e ir desarmando el decorator hasta llegar al último elemento que, en una jugada defensiva (representada aquí como el decorator que contiene varios pases y un tiro) siempre será un tiro.

Una vez obtenidas la jugada ofensiva y la jugada defensiva del turno de los técnicos, se utiliza un Double Dispatch para obtener la Acción Defensiva apropiada en relación a la acción ofensiva que se esté simulando, por ejemplo, si yo quisiera hacer un Pase, el simulador en sí no sabe qué tipo de acción ofensiva es la que comienza la jugada devuelta por el técnico, por lo cual llamaría al método de Pase `obtenerReacciónDefensiva` pasando por parámetro a la `EstrategiaDefensiva` que haya elegido el técnico defensor; luego, el Pase sí puede pedirle a la estrategia defensiva la respuesta correcta, y lo hace llamando a `responderPaseDe`, y devolviendo una lista de Acciones Defensivas (que representan a la lista de jugadores que están “marcando” al que intentó hacer la acción ofensiva), que en el caso del Pase serán objetos del tipo Intercepción, luego el simulador podrá con cada intercepción ver si fue exitosa o no y así decidir el resultado del turno.

Tanto las Acciones Ofensivas como las Defensivas tienen una función `simularTriunfo`, que es llamada por el simulador en caso de que la acción haya sido exitosa, el simulador se pasa a sí mismo a éste método en una suerte de Double Dispatch, para que luego la acción que haya sido exitosa pueda decirle al Simulador cual es la acción siguiente correcta, por ejemplo, en caso de tener una Intercepción exitosa, el método de `simularTriunfo` le indicará al simulador que debe continuar el turno, pero con los roles de defensa y ofensa invertidos.

1.4. Tiro

Es muy similar al pase para la mayor parte de la simulación, solo que se comportará distinto al llamar `obtenerReaccionOfensiva`, pidiéndole a la `Estrategia Defensiva` esta vez `responderTiroDe`, ya que la acción ofensiva es en este caso un tiro. Similarmente, si fuera triunfante el `simularTriunfo` se encargará de que se sumen los puntos correspondientes dependiendo del tipo de tiro

1.5. Contraataque

Para adaptar la estrategia a la interpretación que nosotros realizamos del dominio, si bien el enunciado explica al contraataque como una estrategia ofensiva que se realiza in-

mediatamente después de un robo de pelota, nosotros cambiamos la definición de la misma (manteniendo la habilidad de representar el dominio) debido a que el contra-ataque no comparte muchas de las características que sí tienen las otras Estrategias Ofensivas, como por ejemplo, no es posible elegirla como Estrategia Ofensiva al principio de un turno, puesto que solo puede ser reacción de haber robado un pase— en cambio, nos paramos un paso detrás, y decidimos que la Estrategia de Contraataque es en sí defensiva, e implica primero intentar robar el balón del equipo contrario, y en caso de ser exitoso, ahí sí contraatacar.

La parte específica de la simulación del contra-ataque (por ejemplo, el hecho que una vez empezado el tiro es inbloqueable por el equipo defensor) está incluida en el simulador, que tiene un método específico para `simularContraataque`. Llamar a `simularContraataque` (y también decidir qué tiro se utilizará) será la responsabilidad de la acción ofensiva de **intercepcionContraofensiva**, que es la acción defensiva principal de una Estrategia Contraataque, todo esto será parte de su método `simularTriunfo`.

1.6. MVP

La jugada de MVP no tiene ninguna característica demasiado especial, es una EstrategiaOfensiva como el resto, solo que al armar la Jugada Ofensiva que devolverá al llamar al método de `darAccion`, armará un Decorator que represente el pase necesario para llegar al MvP y el tiro del MvP, puede hacer ésto con facilidad pues recibe por parametro al equipo del cual requiere dar una Jugada, con lo que chequeará en el equipo para saber el MvP y armar las acciones correspondientemente.

1.7. Pelota Dividida

Para simular la pelota dividida se va creando objetos de la clase `Rebote`, que representan un intento de rebote de un jugador. El simulador de turnos mediante el llamado `SimularPelotaDividida` llama primero al jugador defensivo con mayor posición y manda un mensaje `triunfaCon` para ver si efectivamente ese jugador agarra la pelota, en caso afirmativo el `SimuladorDeTurnos` se envía un mensaje para seguir con la simulación de la jugada, dado que no cambio el turno. Si la pelota no es agarrada por el jugador defensivo, se crea otro objeto de la clase `rebote` del jugador ofensivo y se analiza nuevamente si triunfa el rebote. Cada jugador tendrá chance de atraparla de forma intercalada entre el equipo defensivo y el ofensivo en orden decreciente de su posición mediante las llamadas de `triunfaCon`. Si ninguno de los 10 jugadores se hace con la pelota, la misma sale de la cancha y ahí sí se termina el turno.