

Jogo Rubynho

Descripción de la Arquitectura

Versión [1.0]

Historia de revisiones

Fecha	Versión	Descripción	Autor
14/09/2014	1.0	Documentació Inicial	Mauricio Calcagno

1. Introducción

En este documento se hace una introducción a la arquitectura del sistema, debido al proceso a seguir (SCRUM) la misma no es aún definitiva sino que es solamente un bosquejo

1.1 Propósito

Este documento proporciona una apreciación global y comprensible de la arquitectura del sistema usando diferentes puntos de vista para mostrar distintos aspectos del sistema. Intenta capturar y llegar a las decisiones de arquitectura que se han hecho en el sistema

1.2 Alcance

El objetivo de este documento es brindar un primer acercamiento a la arquitectura del sistema

2 Vista de Historias de Usuario

2.1 Historias de Usuario más Relevantes para la Arquitectura

2.1.1 Se crea una estrategia nueva y se la guarda como precargada

En esta historia el usuario crea una estrategia nueva en código ruby y la misma se guarda como precargada

2.1.2 Mostrar en pantalla el resultado de la simulación

Luego de realizada la simulación o mientras se realiza la misma se podrá ver en pantalla una animación que representa la simulación hecha del partido

2.1.3 Crear una liga nueva

2.1.4 Crear una copa nueva

2.1.5 Crear un partido entre dos equipos

3 Vista del Modelo de Diseño

3.1 Diseño de Clases



Arquitectura v2.pdf

Se adjunta el Diagrama de Diseño de Clases

3.1.1 Tournament

Esta clase es el modelo del Campeonato (Copa o Liga), serían las

entidades campeonato en sí mismas

3.1.2 Tournament_Controller

Esta clase se encarga de implementar la lógica del campeonato, calcular cruces, posiciones, estadísticas (en caso de que luego se agreguen) y es el encargado de cuando un partido tenga que desarrollarse de invocarlo, así como también la inclusión o no de un Equipo en el campeonato.

3.1.3 Match

Es el partido, guarda información del estado del partido mientras este se ejecuta, referencia a los Equipos y las entidades de Estadísticas, Resultados y los datos a consumir por el cliente para la visualización del mismo

3.1.4 Match_Controller

En esta **clase** se encapsula la lógica del partido, es el encargado de invocar al simulador y pasarle los Equipos y sus respectivas estrategias.

3.1.5 Simulator

Esta clase encapsula la lógica de la simulación propiamente dicha, mientras ejecuta la simulación va creando la estructura de visualización, estadísticas y resultado final.

3.1.6 Result

Es la entidad de resultado

3.1.7 Statistik

Son las estadísticas del partido

3.1.8 Play

Es la estructura que guarda el transcurso del partido en un formato específico que luego es interpretado por el cliente para la visualización animada de la simulación

3.1.9 Team

Es la entidad de Equipo

3.1.10 Player

Es la entidad Jugador

3.1.11 Team_Controller

Esta clase encapsula la lógica de Equipos y se encarga de modificar las relaciones Equipo – Jugador, recae en ella las

responsabilidades de administrar las operaciones de ABM de jugadores y en cambiarlos de equipo como cualquier otra historia relativa a los mismo que se implemente en el futuro.

3.1.12 User

Es la entidad que se encarga de guardar el perfil del usuario, la cual referencia las Apuestas y los Mensajes del usuario

3.1.13 Bet

Es la entidad de Apuesta

3.1.14 Message

Es la entidad de Mensaje

3.1.15 Observer

Es una interfaz que referencian tanto Tournament como Match para implementar el Patrón Observer, esta interfaz se utiliza para notificar de cambios en los Campeonatos y Partidos

3.1.16 Message_Controller

Encapsula la lógica de mensajes (envío y recepción) y como implementa Observer se encarga de enviar notificaciones de cambios en los campeonatos en los que el usuario esté inscripto

3.1.17 Bet_Controller

Encapsula la lógica de apuestas (ABM y notificaciones) y como implementa Observer se encarga de enviar notificaciones de cambios en los campeonatos o partidos en los que el usuario haya apostado