

Proyecto Forrest

Liga de Simulación 2D

RoboCup

Descripción de Versiones

Versión 7.0

Histórico de Revisiones

Versión	Fecha	Resumen de cambios	Autor
1.0	09-03-2006	Descripción de la versión 1.0	Grupo <i>Forrest</i>
2.0	22-03-2006	Descripción de la versión 2.0	Grupo <i>Forrest</i>
3.0	15/04/2006	Descripción de la versión 3.0	Grupo <i>Forrest</i>
4.0	30/04/2006	Descripción de la versión 3.1	Grupo <i>Forrest</i>
5.0	10/05/2006	Descripción de la versión 4.0	Grupo <i>Forrest</i>
6.0	20/05/2006	Descripción de la versión 5.0	Grupo <i>Forrest</i>
7.0	30/05/2006	Descripción de la versión 6.0	Grupo <i>Forrest</i>

Índice

1. Introducción	3
2. Registro de versiones	4
2.1. Versión 1.0	4
2.2. Versión 2.0	5
2.3. Versión 3.0	6
2.4. Versión 3.1	7
2.5. Versión 4.0	8
2.6. Versión 5.0	9
2.7. Versión 6.0	10

1. Introducción

Este documento tiene propósito llevar un registro de todas versiones que fueron generadas del sistema *Forrest* a lo largo de este proyecto. En la Fig. 1 se muestra gráficamente como fueron evolucionando estas versiones, en donde cada arista del árbol representa una relación entre dos versiones.

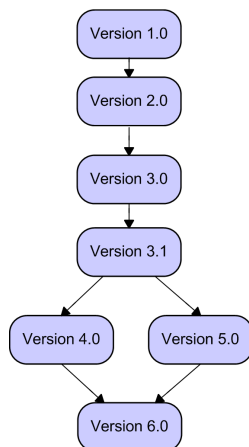


Figura 1: Evolución de las versiones del sistema *Forrest*.

En la siguiente sección, se presenta la descripción de cada una de estas versiones con la siguiente información: fecha de creación, autores, descripción de sus características, principales cambios realizados y un reporte de los errores y limitaciones encontradas en la misma.

2. Registro de versiones

2.1. Versión 1.0

Fecha de la versión

09-03-2006

Autores

Raúl Canales, Serrana Casella, Pablo Rodríguez

Descripción

En esta versión se integra por primera vez la estrategia procedural definida para el equipo *Forrest* con el framework UvA Trilearn 2003 [CANa]. La misma se divide en dos partes según el estado del juego: *estrategia con pelota* cuando el equipo *Forrest* tiene posesión de la pelota y *estrategia sin Pelota* cuando ocurre lo contrario. En esta versión también se incluyen conceptos importantes como: *función de utilidad*, *posicionamiento estratégico* y *mirada estratégica*.

Componentes de la versión

- Archivos fuentes: Todo el proyecto
- Binarios: forrestJugador y forrestCoach
- Archivos de configuración: player.conf, formations.conf
- Script de arranque: start.sh

Cambios Asociados

Se extienden varias funcionalidades del framework UvA Trilearn 2003 asociadas al Modelado de la Realidad.

Errores conocidos y Limitaciones

- Cuando hay una jugada de pelota parada a favor del equipo contrario, el jugador más cercano a la pelota del equipo *Forrest* intenta interceptarla. El simulador se lo impide pero este comportamiento le ocasiona un gasto de energía importante al jugador.
- En reiteradas oportunidades, cuando hay una jugada de pelota parada a favor del equipo *Forrest* ningún jugador ejecuta la falta. Esto ocasiona que el juez cobre pelota dividida por lo que el equipo pierde la posesión de la pelota.
- Cuando un jugador ejecuta una jugada de pelota parada y no tiene opción de pase intenta salir jugando con la acción Dribble. Esto ocasiona que en varias oportunidades el juez cobre falta a favor del equipo contrario.
- Pobre poder ofensivo y baja eficacia en los tiros al arco.

2.2. Versión 2.0

Fecha de la Versión

22-03-2006

Autores

Raúl Canales, Serrana Casella, Pablo Rodríguez

Descripción

En esta versión se rediseñan varios de los componentes desarrollados en la versión 1.0 con el objetivo de desacoplar el diseño del equipo del framework UvA Trilearn. Funcionalmente el equipo no se modificó en esta versión.

Componentes de la versión

- Archivos fuentes: Todo el proyecto
- Binarios: forrestJugador y forrestCoach
- Archivos de configuración: player.conf, formations.conf
- Script de arranque: start.sh

Cambios Asociados

- Se crea la clase Forrest como subclase de Player. Todo lo que se definió del equipo *Forrest* en la clase Player fue borrado y redefinido en la nueva clase.
- Se redefinieron e implementaron los metodos: goalieMainLoop, defenderMainLoop, midfielderMainLoop, attackerMainLoop. Estos metodos pertenecen a la clase Player y son necesarios para implementar la estrategia del equipo (son llamados desde el main del framework).
- La implementación de las nuevas funcionalidades del modelado de la realidad definidas por el equipo *Forrest* se pasaron al archivo ForrestWorldModel.cpp.
- Se creo el archivo ForrestType.h en donde se encuentran definidas todas las constantes y enumerados utilizados por el equipo Forrest.
- Se restablecieron las clases del framework modificadas para la implementación del equipo *Forrest* en la Versión 1.0.
- Se agregaron y unificaron comentarios en el código.

Errores conocidos y Limitaciones

Persisten los mismos errores de la Versión 1.0

2.3. Versión 3.0

Fecha de la Versión

15/04/2006

Autores

Raúl Canales, Serrana Casella, Pablo Rodríguez

Descripción

En esta versión se rediseña completamente la función de utilidad y se modifica el proceso de selección de acciones. Con el fin de simplificar la verificación de la nueva versión se deja de utilizar la *estrategia sin pelota* implementada en la versión 1.0.

Componentes de la versión

- Archivos fuentes: Todo el proyecto
- Binarios: forrestJugador y forrestCoach
- Archivos de configuración: player.conf, formations.conf
- Script de arranque: start.sh

Cambios Asociados

- Se rediseña completamente todos los métodos relacionados con la función de utilidad.
- Se modifica la clase Forrest para incorporar el nuevo proceso de selección de una acción.

Errores conocidos y Limitaciones

- Cuando un jugador ejecuta una jugada de pelota parada y no tiene opción de pase intenta salir jugando con la acción Dribble. Esto ocasiona que en varias oportunidades el juez cobre falta a favor del equipo contrario.

2.4. Versión 3.1

Fecha de la Versión

30/04/2006

Autores

Raúl Canales, Serrana Casella, Pablo Rodríguez

Descripción

En esta versión se incorporan nuevos factores de evaluación a la función de utilidad (densidad de oponentes, riesgo de perder la pelota).

Componentes de la versión

- Archivos fuentes: Todo el proyecto
- Binarios: forrestJugador y forrestCoach
- Archivos de configuración: player.conf, formations.conf
- Script de arranque: start.sh

Cambios Asociados

- Se rediseñan algunos metodos de la funcion de utilidad.
- Se modifica la clase Forrest para incorporar el nuevo proceso de selección de una acción.

Errores conocidos y Limitaciones

- Persisten los errores de la versión 3.0

2.5. Versión 4.0

Fecha de la Versión

10/05/2006

Autores

Raúl Canales, Serrana Casella, Pablo Rodríguez

Descripción

En esta versión la estrategia del equipo es diseñada e implementada para la utilización del método *grafos de coordinación*. Este método permite que cada jugador tome sus acciones de forma coordinada con el resto de sus compañeros, con el fin de maximizar la utilidad global del equipo en cada ciclo. Para la evaluación de las distintas acciones que puede tomar un jugador se utiliza la función de utilidad presentada en la versión 3.1.

Componentes de la versión

- Archivos fuentes: Todo el proyecto
- Binarios: forrestJugador y forrestCoach
- Archivos de configuración: player.conf, formations.conf
- Script de arranque: start.sh

Cambios Asociados

- Se incorporan nuevas clases para implementación del diseño de la técnica de *grafos de coordinación*.
- Se realizan varias modificaciones y ajustes a la clase Utilidad.
- Se agregan nuevas funcionalidades al modelado de la realidad y a las habilidades básicas de los jugadores ofrecidos por el framework.

Errores conocidos y Limitaciones

- Cuando el arquero del equipo Forrest tiene en su poder la pelota, uno o varios defensas del propio equipo intentan interceptarle la pelota. Esto ocasiona que el juez cobre *pase atrás* a favor del equipo contrario al borde del área grande.

2.6. Versión 5.0

Fecha de la Versión

20/05/2006

Autores

Raúl Canales, Serrana Casella, Pablo Rodríguez

Descripción

En esta versión la estrategia del equipo es diseñada e implementada para la utilización del método *planificación en línea de dos niveles*. A diferencia del método utilizado en la versión 3.1, esta técnica, no tan solo tiene en cuenta el estado actual del juego para la evaluación de una acción, sino que también considera los estados futuros. En particular la cantidad y calidad de las acciones que se puedan tomar a partir de la acción elegida en el ciclo actual.

Componentes de la versión

- Archivos fuentes: Todo el proyecto
- Binarios: forrestJugador y forrestCoach
- Archivos de configuración: player.conf, formations.conf
- Script de arranque: start.sh

Cambios Asociados

- Se define la clase PlanLinea2N la cual implementa el método de *planificación en línea de dos niveles*.

Errores conocidos y Limitaciones

No se detectaron.

2.7. Versión 6.0

Fecha de la Versión

30/05/2006

Autores

Raúl Canales, Serrana Casella, Pablo Rodríguez

Descripción

En esta versión la estrategia del equipo se basa en el método *de grafos de coordinación*. Para resolver la evaluación de una determinada acción, se utiliza el método de *planificación en línea de dos niveles* presentado en la versión 5.0.

Componentes de la versión

- Archivos fuentes: Todo el proyecto
- Binarios: forrestJugador y forrestCoach
- Archivos de configuración: player.conf, formations.conf
- Script de arranque: start.sh

Cambios Asociados

- Se modificaron las clases Regla1, Regla2,...,Regla10 para que utilicen el método *getEvaluacion* de la clase PlanLinea2N.

Errores conocidos y Limitaciones

No se detectaron.

Referencias

- [CANa] Canales, R.; Casella, S.; Rodríguez, P. “*RQRESPROT-Prototipo2_UvaTbasico - Análisis del equipo UvA Trilearn - Distribución 2003*”. Tejera, G. (tutor). Trabajo de Grado. Facultad de Ingeniería. Instituto de Computación - Montevideo, Diciembre 2005.