

PROYECTO DE PROLOG DE PROGRAMACIÓN DECLARATIVA

IMPLEMENTACIÓN DE PROLOG.

Alejandro Díaz Roque C412; Rafael A. Orrasch Santiago C412

20. 12. 2020

Resumen

Este es el reporte de desarrollo del proyecto de Prolog de Programación Declarativa. Es la implementación del juego Azul. Un juego de hasta 4 jugadores cada uno con un tablero.

1. Introducción

Azul consiste en un juego de hasta 4 jugadores, donde cada jugador debe llenar su pared o tablero con azulejos de distintos colores. Gana la partida el jugador que más puntos tenga, al final de la misma. Cada jugador toma los azulejos de los expositores de factoría. Si son 2 jugadores, son 5 expositores, si son 3 jugadores son 7 expositores, si son 4, 9. Cada expositor se llena con 4 azulejos, tomados de la bolsa de los azulejos. La partida acaba cuando un jugador llena una línea horizontal de azulejos de su pared.

Este es un resumen, obviando detalles, acerca del juego. Nosotros hicimos una implementación del mismo en Prolog, de la que daremos una explicación.

2. Definición de las componentes del juego

Para definir los azulejos usamos átomos, que son el respectivo color de cada azulejo. De esta forma, podemos generar una bolsa (`Bag`), nueva, usando la función `create_list`.

Los expositores de factoría se guardan todo el tiempo en una lista de factorías, llamada `FactoryList`. Esta lista de factorías se crea de la siguiente con la función `build_factories`.

Cada jugador tendrá un tablero personal. Este tablero personal se define en la función `compose_board`, con su línea de patrones (`PatternLines`), muro a rellenar (`Wall`), su seguimiento de la puntuación (`ScoreTrack`), y su línea de suelo (`FloorLine`).

3. Simulación del juego

La implementación requiere de interacción con el usuario. Usando la función, en cada paso se especifica en detalle el tipo de información requerida a escribir. Esta debe ser un átomo, que termine en `..`. Esto significa, que la decisión de cada jugador, la toma un usuario escribiendo datos en la entrada estándar, usando la consola de prolog. Si una instrucción es mal escrita, el juego termina automáticamente.

El juego comienza con la función `play_azul`. La función `init_db` va a rellenar la información necesaria en dependencia de la cantidad de jugadores, y luego `game` arranca las fases del juego.

La función `fill_factories` toma los tokens de la bolsa `Bag`, los pone en cada factoría vacía. Luego, comienza la ronda de oferta de factoría. En esta ronda, cada jugador debe ser controlado por un usuario externo para tomar la decisión de escoger un color de token, y un lugar de donde tomarlo. Mientras queden factorías con tokens, los jugadores siguen rotándose en esta ronda.

Luego, en la ronda de alicatado de pared. Cada jugador, pasa sus tokens desde la `PatternLine` correspondiente a la `LineWall`. El predicado de la forma `game([Players, [], []], Bag, (Round, building_wall), TokensInTopBox))`, describe que se hace en esta ronda. Aquí el jugador moverá, tal como indican las reglas, el primer azulejo de cada `PatternLine` que haya sido completado en la fase anterior, para la `LineWall` correspondiente, y el resto de los azulejos los devuelve a la `Bag`.

Luego está la fase final, donde se cuentan los puntos de cada jugador, y se escoge un ganador.

4. Como correr el software

Con una consola interactiva de prolog, como puede ser `swipl`, ejecutar el predicado `[game_definitions].`, y leer las instrucciones para introducir los átomos pertinentes.