Integrantes

Amanda Marrero Santos - C411

Manuel Santiago Fernández Arias – C411

AZUL

❖ Modelación

Los elementos básicos en la modelación de un juego de azul son:

- Jugadores
- Bolsa de azulejos
- Fábricas de azulejos
- Suelo
- Bolsa Auxiliar de azulejos

En el juego de azul participan de dos a cuatro jugadores. Cada jugador está identificado con los siguientes elementos:

- Tablero(T): es una matriz cuadrada de 5 por 5 con una distribución de colores predefinida, lo representamos como una lista de listas
- Escalera(PatterLine)(PL): es una matriz triangular superior, representada con una lista de listas
- Lista de Negativos (LN): representada como una lista
- Puntuación(P): el valor de su puntuación total
- Identificador (I): identificador secuencial para cada uno de los jugadores
- Un jugador J = [T, PL , LN , P , I]

La bolsa de azulejos en un primer momento está formada por 20 azulejos del cada color (azul, amarillo, rojo, negro, azul claro). En cada ronda se abastecen las fábricas de 4 azulejos y si en algún momento ,al empezar una ronda del juego, no hay azulejos en la bolsa ,entonces tomamos los azulejos de la bolsa auxiliar (aquellos que se van desechando en cada ronda).

La cantidad de fábricas en un juego depende de la cantidad de jugadores:

- 2 jugadores -> 5 fabricas
- 3 jugadores -> 7 fabricas
- 4 jugadores -> 9 fabricas

El suelo lo vamos a considerar como una fábrica más, solo que al inicio de cada ronda tendrá un azulejo negativo que se le asignará a la primera persona que coja azulejos del mismo. EL suelo se va llenando con los azulejos que desecha cada jugador en su turno. El jugador que al finalizar la ronda tenga en su poder el azulejo negativo será el indicado para empezar la ronda siguiente.

❖ Funciones Predefinidas

Dentro de las funciones predefinidas en PROLOG las que más utilizamos fueron:

- length (L, R), que triunfa si el length de L es igual a R
- nth0(I, L, R), que triunfa si el elemento i-esimo (en base 0) de L es R
- append (X, L, RL), que triunfa si RL es el resultado de añadir el elemento X a la lista L
- selectchk (Z, L, RL), que triunfa si Z es el primer elemento de L y RL el resto
- **findall (F, O, L):** crea la lista L con todas las unificaciones de las variables que aparecen en F correspondientes a todas las respuestas, obtenidas por retroceso cronológico, para el objetivo O.

❖ Estrategia

Definimos 4 estrategias de juego, todas basadas en la selección de un movimiento viable.

Un movimiento es toda aquella posibilidad de escoger un conjunto de azulejos de un mismo color de una fábrica o del suelo, va estar conformado por los siguientes elementos:

- P -> valor booleano que representa si ese movimiento fue el primer movimiento que selecciono el suelo en esa ronda
- F/S -> valor booleano que representa si se seleccionó una fábrica o el suelo
- I -> índice en la lista de fábricas en caso de que se haya elegido una
- C:X -> color C con una cantidad X
- R –> el resto de los azulejos que estaban en el lugar seleccionado

Un movimiento viable es aquel que cumple con alguna de las siguientes condiciones:

- Que exista alguna línea en el PatterLine vacía y que su color en el tablero en dicha fila no este ocupado
- Que exista alguna línea en el PatterLine, que contenga su mismo color y que tenga al menos un espacio libre

Estrategias Implementadas:

- Siempre Random
  En cada turno generamos la lista de los movimientos viables y seleccionamos uno random. Si es el caso de que en un turno determinado no existen movimientos viables tomamos el primer movimiento.
- Diagonal Superior
  Esta consiste en tratar de completar primero los elementos del tablero que se encuentran de la diagonal superior.
  En cada turno generamos la lista de todos los movimientos viables. Dicha lista la dividimos en dos listas, una que contiene todos los movimientos viables de la diagonal superior(L1) y el resto en la otra lista(L2). Si existe algún elemento en L1 entonces tomamos un elemento random de la misma; en otro caso si existe algún elemento en L2 tomamos un elemento aleatorio de la misma y si no existen movimientos viables tomamos el primer movimiento.
- Rápido y Furioso
  La siguiente estrategia consiste en tratar de completar las dos primeras filas del tablero a toda costa. Para ello dividimos la lista de los movimientos viables en dos, una que contenga los movimientos de las dos primeras filas del tablero y otra que contenga los restantes. La elección será random, priorizando siempre la primera lista. En caso de que no existan movimientos viables se tomará el primer movimiento.
- Siempre a salvo
  Esta estrategia consiste en minimizar en cada movimiento la cantidad de azulejos que añadimos a su lista de negativos. Para ello se computan todos los movimientos viables y se toma el que menos fichas añada a la lista de negativos, de esta forma puede que no se maximicen los puntos alcanzados, pero se mantienen a toda costa los puntos acumulados anteriormente.

❖ Simulación

Dentro de la simulación de la partida, el predicado más importante sin lugar a dudas es play con los siguientes argumentos:

- LP -> lista de jugadores
- CF -> cantidad de fabricas
- B -> Bolsa de azulejos
- G -> piso
- NB -> bolsa auxiliar
- R -> donde se almacena el resultado

El predicado play consiste en realizar una ronda de jugadas, luego con el resultado obtenido de la ronda actualiza el estado de la partida, o sea, los jugadores, el suelo, la bolsa y la bolsa auxiliar. Tras dicha actualización se verifican las condiciones de parada del juego:

- Que exista algún jugador que haya completado una fila
- Que no queden más azulejos disponibles en la bolsa, ni en la bolsa auxiliar

Si nos encontramos ante una situación de parada se llama al predicado endGame que determina el ganador.

El resto de los predicados que juegan un papel determinante en la simulación:

- Makeround (LP, F, G, R) -> triunfa si R es la lista de los nuevos jugadores tras una ronda de juego

  Predicados auxiliares de makeRound:

  - makeMoveI (P, F, G, PMR) -> triunfa si PMR es el resultado del movimiento del jugador P con las fabricas F y el piso G. Este es el predicado asociado a las estrategias (makeMove0: Siempre Random, makeMove1: Diagonal Superior, makeMove2: Rápido y Furioso, makeMove3: Siempre a salvo)
  - move (P, MV, R) -> triunfa si R es el resultado de que el jugador P realice el movimiento MV.

- SumOfPoints (B, PL, NL, R) -> triunfa si R es la cantidad de puntos obtenidos con ese tablero(B), patterLine(PL), y lista de negativos(NL); al finalizar una ronda

  Este predicado se apoya en predicados más sencillos como:

  - SumPointsOfMove (B, X, Y, R) -> triunfa si R es la cantidad de puntos obtenidos al poner un azulejo en la posición X, Y del Tablero(B).
  - completeLines (B, R) -> triunfa si en el tablero B existen R filas completas
  - completeColumn (B, R) -> triunfa si en el tablero B existen R columnas completas.
  - completeColor (B, R) -> triunfa si en el tablero B existen R colores completos.
- UpdatePlayers (LP, NPL) -> triunfa si NLP es el resultado de actualizar los jugadores de LP
- IsEnd (LP, R) -> triunfa si existe jugador en LP que tenga al menos una fila completada en su tablero

❖ Ejecución

Para la ejecución de la simulación, basta con llamar al predicado game (N, R), que triunfa si en una simulación de una partida de azul gana el jugador R.