

Informe de Programación Declarativa.

Azul

Liset Alfaro lisetalfarogonzalez@gmail.com

Facultad de Matemática y Computación (MATCOM),
Universidad de la Habana (UH), Cuba.

1. Idea general

Azul es un juego de mesa, como los juegos de este tipo el azar es una característica muy importante además del uso de estrategias por cada uno de los jugadores. Para la reproducción del juego es fundamental una primera etapa de análisis con el objetivo de representar correctamente cada una de sus partes, esto fue posible mediante la creación de objetos dinámicos. Al analizar las especificaciones y reglas del juego se tuvo en cuenta que los jugadores actúan como agentes sin la intervención de un usuario y tienen conocimiento del estado del juego (ambiente) en todo momento, además de poder modificar su propio estado y el de su ambiente. A cada jugador se le asigna un Id y está constituido por un tablero personal conformado por la zona de preparación (stairPlayer), el muro (board), la fila de penalización (playerFloor), además una puntuación parcial (points). Mientras que el estado del ambiente está dado por los respectivos estados de las factorías, el centro de la mesa, la tapa de la caja y el turno del siguiente jugador.

2. Diseño

Para representar y organizar mejor cada proceso se dividió el proyecto en varios módulos:

azul.py

En este módulo encontramos lo principal en lo que refiere a la simulación del juego. Las principales fases del juego se encuentran implementadas aquí. Inicialmente se genera el juego, dígame la bolsa, la tapa, las factorías, el centro, etc; posterior a ello se generan los jugadores y entonces se desarrolla el juego.

game.pl

Acá tenemos todas las funciones relacionadas con el juego: Los generadores, los trabajos con las factorías y las tapa así como el centro.

player.pl

En esta se tiene a todas las funciones que le interesan solamente a los jugadores como los generadores, los movimientos, ubicar en las escaleras(stairs) correspondientes, actualizar el piso, entre muchas otras.

printing.pl

En este módulo se tienen , como su nombre sugiere, las funciones para imprimir en la consola.

punctuation.py

Donde se tiene las funciones que calculan los puntos de un jugador durante y al final de la partida.

utils.py

En este módulo se tiene todo tipo de funciones, principalmente con listas, que facilitan la simulación

3. Estrategia

La estrategia a seguir no es la más eficiente pero sí permitió simular muchísimas situaciones que no son muy usuales para un jugador aventajado. Para elegir, el jugador lista sus posibles acciones: Tomar X azulejos de color C de la factoría F o del Centro; escoge random la acción que va a realizar y entonces la ubica en una de las escaleras(stairs) según los siguientes criterios(para X azulejos de color C):

- 1- Si la fila con capacidad X está libre y no tiene en la fila del muro la ficha de color Color correspondiente, entonces se colocan los azulejos en esa posición.
- 2- De todas las escaleras que tienen el color C y no están completas, quedémonos con las que no hagan perder más de 4 puntos(Esto porque si mandamos al piso 3 fichas tendremos una pérdida de 4 puntos).Si alguna puede completarse completamente X azulejos, entonces se escoge esa escalera, si ninguna se completa entonces se escoge la más vacía(garantizando que se pierdan la menor cantidad de puntos).
- 3- De todas las escaleras vacías donde se pueda poner este color(recordemos que si ya existe en esa fila ese color en el muro entonces no se puede poner) seleccionemos nuevamente la que menos puntos nos haga perder.
- 4- De todas las escaleras que tienen el color C y no están completas, veamos aquellas que hacen perder más de 4 puntos. Entonces se escoge la más vacía(garantizando que se pierdan la menor cantidad de puntos).
- 5- Se mandan los X azulejos al piso(o la tapa en su defecto).

4. Simulación

La simulación se realizó para la mayor cantidad de jugadores posibles (4), pero resulta bastante sencillo cambiar la cantidad de jugadores que juegan pues la mayoría de las funciones donde intervienen todos los jugadores reciben la cantidad máxima de jugadores.

Una partida de Azul consta de un número indeterminado de rondas hasta que se cumpla alguna de las condiciones de finalización, estas pueden ser que al menos un jugador complete una línea en su muro o que se acaben los azulejos y ya no sea posible preparar una nueva ronda. Cada ronda consta a su vez de tres fases:

- Preparación de ronda: En esta se llenan las factorías tanto de la bolsa como de la caja cuando la primera se quede sin azulejos.
- Oferta de factoría: En esta fase los jugadores toman los azulejos de las factorías o del centro hasta que a estos no les quede azulejos que ofrecer y los ubican en sus escaleras
- Alicatado a la pared: En esta fase llevamos a los muros de cada jugador el color de los azulejos que logramos completar en nuestra escalera.

Posterior a estas acciones se calculan los daños finales infringidos por las fichas que pasaron al suelo de cada jugador; empieza entonces el ciclo nuevamente hasta que se llegue a una condición de fin de juego. Un juego finaliza si al menos un jugador completa una fila de azulejos en su muro o si no hay más azulejos que poner en las factorías. Cuando el juego está terminado se dan los jugadores.

5. Ejecución de la simulación

Para correr el programa basta con ejecutar en SWI-Prolog o cualquier interprete azul(). Este ejecutará la simulación e irá imprimiendo todo su desarrollo.