

Universidad De La Habana

MATCOM

Proyecto de Programación
Declarativa

HIVE GAME

Autores:

Olivia González Peña C411

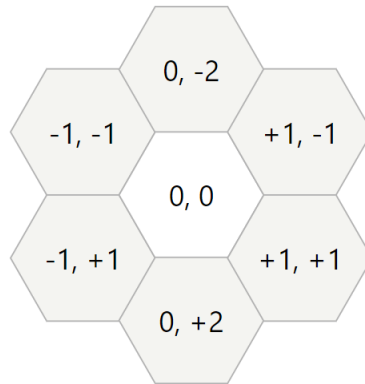
Juan Carlos Casteleiro Wong C411

Introducción

La implementación brindada presenta tres modalidades para el desarrollo del juego: Jugador Vs PC, PC Vs PC y Jugador Vs Jugador. Al iniciar la ejecución del proyecto deberá escogerse la modalidad deseada. Una vez iniciada la partida se asignarán de forma aleatoria el color de las fichas de cada jugador, comenzando siempre por el jugador con fichas blancas.

Interpretamos la acción de colocar una ficha por un jugador, como crear un hexágono con las siguientes propiedades:

- Row: fila correspondiente en el tablero
- Column : columna correspondiente en el tablero
- Bug : tipo de bicho
- Color : color de la ficha
- Level : indicador para conocer la cantidad de fichas en la misma coordenada



Para la representación del tablero compuesto por hexágonos utilizamos el diseño horizontal de doble altura de coordenadas duplicadas, donde las filas contiguas tienen una diferencia de dos unidades [1].

Los bichos son representados del 1 al 8:

- 1 - Abeja Reina
- 2 - Hormiga Soldado
- 3 - Saltamontes
- 4 - Escarabajo
- 5 - Araña
- 6 - Mosquito
- 7 - Mariquita
- 8 - Bicho Bola

El color se representa con 0 para las fichas blancas y 1 para las negras.

El Level de una nueva casilla será 0, al mover una ficha encima de otra el Level será actualizado a Level + 1, de modo que de todas las fichas que compartan las mismas coordenadas tendrá mayor Level la que se encuentra en la cima de la torre.

Desarrollo de la Partida

En cada turno el jugador puede tener dos opciones para jugar, colocar una nueva ficha en el tablero o mover una existente; nuestra solución garantiza mostrar solo aquellas posibilidades que sean válidas para cada turno, de forma tal que, por ejemplo, si la abeja reina del jugador actual no se encuentra en el tablero aun, no será mostrada la opción de mover ficha; asimismo, en caso de colocar todas las fichas de la mano, se mostrarán solo las opciones para mover estas.

Al ser elegida la opción de colocar una nueva ficha, se muestran los bichos que aun dispone el jugador en la mano, y al seleccionar el deseado, se muestran las casillas válidas donde colocarlo. En la primera jugada se colocará siempre la ficha seleccionada en las coordenadas (0,0).

De ser escogida la opción de mover una ficha, se mostrarán las coordenadas de aquellas que al menos puedan realizar un movimiento válido; una vez seleccionadas estas, se deberá elegir las coordenadas destino.

De no existir ninguna de las opciones anteriores para jugar, se pasa el turno de forma automática.

La ejecución termina en el turno en que un jugador logre rodear la Abeja Reina del adversario o ambas; en caso de llegar al turno 100 se declara empate en la partida.

El tablero es representado como una lista de hex(Row, Column, Bug, Color, Level), de forma tal que la lista [hex(0,0,1,0,0), hex(1,1,3,1,0), hex(0,2,5,1,0) hex(-1,-1,4,0,1)] corresponde a un tablero con una Abeja Reina Blanca en la posición (0,0), un Saltamontes Negro en la posición (1,1), una Araña Blanca en la posición (0,2) y un Escarabajo Blanco encima de otra ficha en la posición(-1,-1). Las fichas que se encuentra cubiertas por otras son representadas como uselessHex, en el ejemplo anterior existiría una uselessHex con coordenadas (-1,-1).

En cada turno se muestra el estado actual del tablero, así como las fichas cubiertas por otras.

Movimiento de los Bichos

Para implementar la lógica del movimiento de los bichos, definimos predicados para computar los vecinos vacíos y ocupados de las casillas, los destinos a los que se puede llegar con deslizamiento, las casillas vacías adyacentes a la colmena, así como para revisar la condición de que se mantenga una única colmena con todos las jugadas que se realicen.

Abeja Reina

Para mover la Abeja Reina se hallan las casillas a las que se pudiera deslizar, y para cada una se simula esta jugada, resultando solo como posibles aquellas que mantengan conectada a la colmena.

Escarabajo

Para mover al Escarabajo se repite el proceso anterior, y se valoran además las casillas llenas a su alrededor a las que se pudiera encaramar, filtrando estas para posibilitar el movimiento solo a aquellas que, igualmente, mantengan conectada la colmena.

Saltamontes

Para mover al Saltamontes, se parte de los vecinos ocupados que tenga este, y desde ellos se avanza en la misma dirección en la que se encuentran con respecto al saltamontes, hasta la próxima casilla vacía. Se simula que el saltamontes se encuentra ahora en estas posiciones y serán parte de las posibilidades a valorar, solo aquellas que mantengan unida la colmena.

Araña

Para mover la Araña, se avanza siempre por casillas a las que se pueda llegar por desplazamiento, teniendo en cuenta la restricción de que sean los tres pasos a casillas adyacentes a la colmena y; valorando como opciones finales aquellas casillas a las que si la araña se desplazara, se sigue manteniendo la colmena unida.

Hormiga

Para mover la Hormiga, se computan las casillas a las que se puede desplazar siempre permaneciendo adyacente a la colmena, hasta llegar nuevamente a su lugar de origen; siendo solución todas las casillas de

este camino que cumplan que mantienen unida a la colmena de ser esta el destino escogido.

Mosquito

Para mover al Mosquito, se busca, primeramente, el nivel en el que se encuentra, si es cero, se analizan los bichos vecinos del mismo; para cada uno se simula que el mosquito es de este tipo de bicho hallando sus posibles movimientos, resultando como solución la unión de las soluciones obtenidas en cada caso. Para el caso en el que el mosquito este encaramado, se comportará directamente como si fuera un escarabajo.

Mariquita

Para mover la Mariquita se recorren los vecinos ocupados, y de estos últimos a su vez, los vecinos ocupados que tengan vecinos vacíos donde pudiera aterrizar en el tercer paso; se simula este movimiento obteniendo como solución aquellos que mantengan unida la colmena.

Estrategia de la PC

En caso de que el jugador sea la PC, el comportamiento del flujo explicado es el mismo, cambiando esta vez la forma enfrentar cada decisión.

De tener las opciones de colocar una ficha nueva en el tablero y mover una previamente colocada, la PC escogerá una de estas de forma aleatoria.

De haber seleccionado la opción de colocar una nueva, escogerá entre los bichos de su mano de forma aleatoria cual colocar. Para escoger el destino de esta ficha, analizará si la reina del rival está ya en el tablero e intentará acercarse a ella lo más posible con el fin de rodearla; escogiendo así, entre las casillas posibles para el destino, la que menor distancia tenga con respecto a la reina del rival.

En caso de haber decidido mover una de sus fichas en el tablero, escogerá entre las posibles de forma aleatoria e, igualmente, luego de seleccionada, el destino de esta dependerá de la distancia con respecto a la reina del rival; siguiendo la estrategia de intentar acercarse a ella hasta acorralarla y así, ganar el juego.

Especificaciones

Una ficha que tiene dos fichas adyacentes en común con una casilla, no podrá llegar a esta realizando un deslizamiento.

Tanto la mariquita como el escarabajo se pueden bajar en una casilla vacía adyacente si están en un nivel mayor que 0, sea alcanzable por deslizamiento o no.

En ningún caso, los bichos que se mueven más de una casilla por turno pueden retroceder en el camino.

A pesar de conocer las desventajas de la estrategia de la PC, y de estar lejos de ser "inteligente"; decidimos presentarla ya que no logramos integrar al resto del proyecto el algoritmo Minimax sugerido. Lo mismo ocurrió con los movimientos del Bicho Bola.

Igualmente sería muy útil integrar una plataforma visual que permita visualizar el tablero y escoger a través de este las jugadas a realizar.

Referencias

[1] - <https://www.redblobgames.com/grids/hexagons/>