Lenguaje de Programación 2

Entrega de TP

Arboles de juego: 12 Hombres

Alumno: Andrés Matitas Aguilera Acosta

Consideraciones:

1. El desarrollo del juego transcurre en una matriz char normal, donde las reglas se verifican en funciones de una manera modularizada.
2. El árbol creado es de 5 niveles en las fases iniciales, es decir, en las fases donde el tablero está casi vacío, con pocas fichas, de manera que la computadora pueda procesar y almacenar todos los nodos del árbol que queremos. Y luego, varían los niveles según la fase de juego.
3. El árbol posee n hijos, cada nodo creado es insertado al árbol por addnode, es decir, cada hijo apunta a su hermano, se utiliza listas simples enlazadas.
4. Se utiliza el corte alpha beta, que llega a las hojas o nodos terminales, donde solamente ahí son evaluados, y luego con el algoritmo sube hasta la raíz el valor del mejor nodo evaluado.
5. En todo momento, el corte alpha beta hace su correcto funcionamiento, que es el principal desafío de este trabajo practico.
6. Se evalúa únicamente mediante la función alpha beta.
7. Una vez obtenido el nodo, se utiliza una función para elegir el hijo de la raíz que posee el valor obtenido del mejor nodo, es decir, el mejor nodo evaluado realmente esa en esa rama.
8. Para sacar una ficha, se evalúa un solo nivel, sacando todas las fichas posibles del contrario y luego evaluando el tablero para escoger cual es la mejor jugada.
9. Cuando se logra un nodo ganador, la evaluación es crítica y se detiene la generación del árbol.
10. Para crear los árboles, se utilizan las técnicas necesarias. Primeramente para colocar, se analiza las jugadas posibles, es decir, donde se pueden colocar las fichas. Para el movimiento, se selecciona todas las fichas y se calcula todos los movimientos válidos, para la fase voladora, se verifica nada más que el espacio donde se puede mover este libre, es decir, genera correctamente en las tres fases, los arboles correspondientes.
11. Los nodos de los arboles están claros en el código fuente.
12. Se utiliza funciones de evaluaciones inspiradas en un trabajo universitario, en donde, se investigó cuáles son las mejores jugadas, en 12 hombres
13. Una función de evaluación inspirada personalmente, trata de colocar una jugada critica a evaluar primeramente. Es una jugada donde el contrario puede hacer mill y la inteligencia no lo puede hacer, entonces trata de evitar un mill contrario.
14. La declaración de las funciones de evaluación, ya escritas en el código fuente:

The evaluation functions were motivated by

Simona-Alexandra PETCU, Stefan HOLBAN

Computer Science and Engineering, Politehnica University of Timisoara

1. La función principal de evaluación divide el juego en 3 fases, y evalúa de forma distinta en cada fase de la siguiente manera:

//Evaluation function for Phase 1 e = 18 \* (1) + 26 \* (2) + 1 \* (3) + 9 \* (4) + 10 \* (5) + 7 \* (6);

//Evaluation function for Phase 2 = 14 \* (1) + 43 \* (2) + 10 \* (3) + 11 \* (4) + 8 \* (7) + 1086 \* (8);

//Evaluation function for Phase 3 = 16 \* (1) + 10 \* (5) + 1 \* (6) + 1190 \* (8);

Cada número corresponde a una función de evaluación, es decir a una forma de evaluar cada fase.

1. En el juego es posible cambiar la forma de evaluación, con la opción 5, es posible cambiar el valor de retorno de la función principal de evaluación, es decir, la formula cambia de números.
2. Es posible elegir la cantidad de fichas en #DEFINE CANTIDAD DE FICHAS 12.
3. Se puede probar el juego en modo HUM vs HUM.
4. El error encontrado podría decir que en momentos, encontré fallas al intentar sacar una ficha por parte de la inteligencia, en raras ocasiones.
5. Durante la competencia me percate que la regla en donde si un jugador que no empezó la partida saca una ficha en su última jugada, no puede mover, en mi código no se llegó a activar esa opción.
6. Y podría decir que siempre se pueden mejorar las funciones de evaluación.