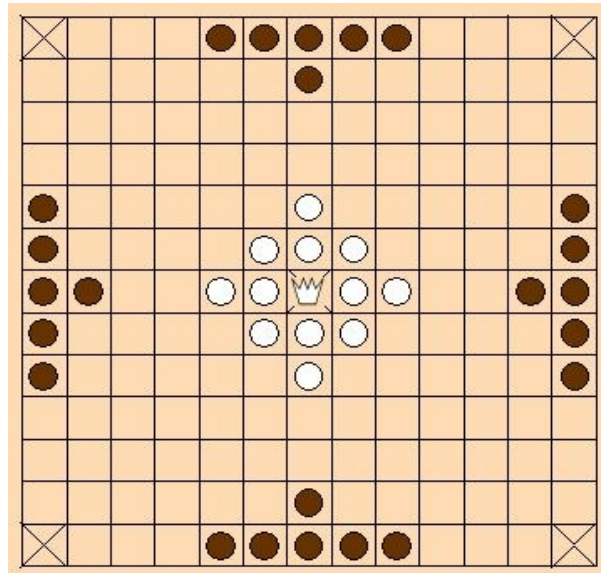


ESTUDIO DEL EQUILIBRIO EN EL JUEGO DEL HNEFATAFL

INTELIGENCIA ARTIFICIAL I. CURSO 2011-2012

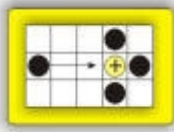
El *Hnefatafl* es un antiguo juego de mesa germánico. La palabra *tafl* significa *mesa* o *tablero* en nórdico antiguo, pero actualmente es la palabra para *ajedrez* en islandés moderno.



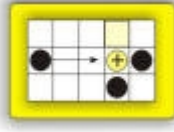
El juego se juega en un tablero cuadrículado (ver figura de arriba), donde hay fichas negras y blancas. Las blancas tienen rey, pero las negras no. Hay distintas variantes según la región donde se jugaba. Se distinguen en los tableros iniciales, pero en todas, las reglas son iguales. A continuación se resumen las **reglas** (ver referencias 1 y 4):

1. Se configura la disposición inicial, en donde el rey va en el centro rodeado de sus peones blancos y los peones negros se encuentran en los lados del tablero. La casilla del centro se denomina *trono* y las casillas de los extremos se denominan *escapes*.
2. Las negras mueven primero.
3. Todas las fichas se mueven en línea recta cualquier número de casillas (como la torre del ajedrez). No se pueden saltar otras fichas.
4. Un peón es capturado y sacado del tablero cuando el oponente ocupa ambas casillas adyacentes en una fila o columna. También se puede capturar un peón si en un lado tiene el trono (vacío) o uno de los escapes y en el lado opuesto se sitúa una ficha enemiga. Sin embargo, una ficha puede moverse de forma segura sobre una casilla vacía entre dos fichas enemigas (no existe el suicidio).
5. El rey es capturado si es rodeado por cuatro fichas enemigas (norte, sur, este y oeste). También si es rodeado por tres lados con fichas enemigas, y la cuarta es el trono (vacío), uno de los escapes, o el límite del tablero. Además, si el rey tiene en alguno de sus lados algún

peón blanco, ambos pueden ser capturados si son inmovilizados, es decir, que el rey y el peón estén totalmente rodeados por fichas enemigas (o por el trono (vacío), casilla de escape o el límite del tablero). Si el rey es capturado, las negras ganan la partida.



Rey capturado por 4 peones negros



Rey capturado por 3 peones negros



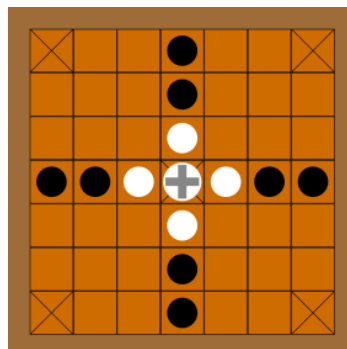
Rey y peón capturados

6. Sólo el rey puede colocarse en la casilla central y en las casillas de las esquinas. Aunque cualquier peón puede atravesar el trono.
7. Las blancas ganan la partida si el rey consigue llegar a uno de los cuatro escapes.

Este trabajo se centra en estudiar el **equilibrio** del juego, es decir, analizar si tanto las blancas como las negras tienen las mismas probabilidades de ganar (siempre y cuando los jugadores no cometan errores graves).

Se pide:

1. Implementar en Lisp, siguiendo el entorno de juegos visto en clase (tema 7, y referencias 2 y 3), una representación de la variante de *Brandub*. En esta versión se tienen las mismas reglas, pero el tablero es de 7x7 y el estado inicial es como el de la siguiente figura:



- Diseñar una estructura de datos para el tablero, los movimientos y los distintos estados del juego. Téngase presente que el juego es *asimétrico*, es decir, existen distintos movimientos y piezas para ambos jugadores.
- Diseñar una función de evaluación estática que trabaje en modo *min* y *max* con cada tipo de jugador (blancas o negras). Implementar y comparar mínimo 3 funciones heurísticas distintas para estados no finales, e identificar la de mejor comportamiento durante el juego.

- Implementar el algoritmo *minimax* con poda *alfa-beta* de manera que la profundidad aumente durante el juego, identificando una cota óptima. Incluir una función aleatoria para elegir en caso de varios nodos de juego con igual valor de función de evaluación estática, con el fin de poder simular partidas distintas.
2. Implementar un modo de juego de humano contra la máquina. Mediante una interfaz de texto, se tendrá que ofrecer la posibilidad de escoger el bando, realizar movimientos indicando la casilla de partida y la casilla de destino, y muestre el estado del tablero en cada jugada por pantalla. Esto servirá para depurar el algoritmo implementado en el punto anterior con el fin de evitar, en la medida de lo posible, que la máquina tome decisiones no lógicas a corto plazo.
 3. Implementar un modo de juego máquina contra máquina, donde se permita de manera opcional observar los movimientos de los jugadores y el estado del tablero en cada jugada. También deberá de tener un modo silencioso donde no se muestre nada por pantalla. Este modo de juego será necesario para el punto 5.
 4. Definir una estructura para almacenar resultados estadísticos de varias partidas, y crear un método para imprimir por pantalla los campos de la estructura:
 - Número de partidas totales.
 - Número de partidas ganadas por las blancas.
 - Número de partidas ganadas por las negras.
 - Proporción de partidas ganadas por las blancas (número de partidas ganadas por las blancas entre número de partidas totales).
 - Proporción de partidas ganadas por las negras (número de partidas ganadas por las negras entre número de partidas totales).
 - Número de movimientos de media que necesitan las blancas para ganar.
 - Número de movimientos de media que necesitan las negras para ganar.
 5. Definir una función (**genera-estadísticas n-partidas f-e-negras f-e-blancas**) en dónde la máquina jugará *n-partidas* usando una función de evaluación estática *f-e-negras* para el jugador que controla las negras y otra función de evaluación estática *f-e-blancas* para el jugador que controla las blancas. La función devuelve datos estadísticos según la estructura realizada en el apartado anterior.

6. Escribir una memoria del trabajo y presentarla en formato *.pdf* junto con el código fuente en formato *.lisp* incluyendo:
- Descripción de las principales funciones implementadas en el código.
 - Descripción de los diferentes métodos de función de evaluación estática empleados. Se valorará la implementación y comparativa de varios métodos.
 - Análisis estadístico del equilibrio del juego, comentando los resultados obtenidos en el apartado 5 y concluyendo si el juego está equilibrado o no.
 - Análisis estadístico del equilibrio de las variantes del juego *Ard Ri* y *Tablut* (ver referencia 1), comentando brevemente los cambios que han sido necesarios realizar en el código anterior, y los resultados obtenidos, concluyendo si están equilibrados.

Criterios de evaluación

- Se valorará la eficiencia en la implementación y el buen estilo de programación en Lisp.
- Se valorará no sólo la calidad de los resultados obtenidos, sino también la claridad con que se presenten en la memoria y en la defensa.
- Se valorará cualquier aportación original que se incluya en el trabajo de manera adicional, aparte de los requisitos mínimos aquí descritos.

Referencias:

1. <http://es.wikipedia.org/wiki/Hnefatafl>
2. <http://www.cs.us.es/cursos/ia1/temas/tema-07.pdf>
3. <http://www.cs.us.es/cursos/ia-2004/temas/anexo-04.pdf>
4. [http://www.ragnarokgames.com/ARCHIVOS/Reglas%20Hnefatafl%20\(Espanol\).pdf](http://www.ragnarokgames.com/ARCHIVOS/Reglas%20Hnefatafl%20(Espanol).pdf)