**Informe del Obligatorio de Inteligencia Artificial – Entrega final**

*Realizado por Leonardo Giovanetti, Juan Niosi y Javier Venossa*

Dado el juego “Square Chess” (FangQí) se puede distiguir que el mismo se divide en 3 diferentes etapas de juego. Resumiendo el proceso de jugar:

* El tablero es de 7x8, cada jugador tiene 28 fichas; hay fichas blancas y negras.
* Se empieza por llenar el tablero tratando de formar cuadrados (2x2), un turno por jugador, evitando al mismo tiempo que el contrario haga cuadrados.
* Luego que el tablero se llena, etapa de remover fichas:
  + Si ninguno formó un cuadrado, cada uno remueve una ficha del contrario y se procede a la siguiente etapa.
  + Si se formó uno o más cuadrados, se le remueve esa cantidad de fichas al contrario (no sobrepasando sus fichas libres -que no forman cuadrados-) y se procede a la siguiente etapa.
* Una vez que se removieron fichas, se turna cada jugador para mover una de sus fichas (solamente para arriba, abajo, izquierda y derecha sin poder pasar por encima a ninguna ficha) para formar nuevos cuadrados de manera de quitarle la misma cantidad en fichas al contrario.
* Gana el que le quita todas las fichas al otro.

Dada esta situación, el método de MiniMax c/poda alfa-beta se hace realmente extenso sea cual sea el estado en donde se encuentre, ya que el tablero 56 posiciones (utilizando la variación del tablero de 7 x 8) lo cual al realizar el árbol puede llegar a recorrer 56! estados, lo que hace que el tiempo de decisión sea muy elevado.

Para esto se decidió implementar una profundidad máxima en el árbol de tal manera que llegada a esa profundidad, el árbol no se explora más. Esta implementación requiere que exista una heurística específica para los últimos estados explorados de manera de poder estimar su valor alfa-beta aun que no sean estados finales.

La heurística implementada es la siguiente:

* Para los estados iniciales de llenado del tablero se cuentan los cuadrados y los cuadrados parciales (o sea, fichas que pueden llegar a formar cuadrados). Se hace la diferencia del jugador vs. el oponente y se divide por la cantidad máxima de cuadrados posibles (17).
* Para el resto se hace la diferencia de fichas de jugador menos las del oponente y se divide por la cantidad máxima (28).

Con respecto a la profundidad, la misma se encuentra parametrizada en el valor 3. Luego de varias pruebas encontramos que ese valor es el óptimo considerando el resultado de la jugada elegida y el tiempo que se toma en decidirse.

Por otro lado, existe la posibilidad de que el resultado del juego se vea comprometido por las consecutivas decisiones tomadas en la etapa de remover fichas. Con esto nos referimos a las decisiones que se toman en esta etapa cuando los dos jugadores eligen sus jugadas para que no sea posible remover ninguna ficha. Esto hace que el juego entre en una especie de bucle por lo cual nos vimos obligados a limitar esta situación implementando una cantidad máxima de jugadas sin remover (parametrizada en 20), para que llegado el caso de que se llegue a este valor, las jugadas se interrumpen retornando un empate, ya que nos parece adecuado suponer que ningún jugador va a cambiar su modalidad de juego para dejar que el contrario tenga chances de ganar y él mismo de perder.

A continuación se lista los cambios realizados desde la pre-entrega:

* Se revisaron los controles de cambios de turno para evitar situaciones extremas. Por ejemplo, cuando un jugador se encuentra bloqueado por fichas del oponente no puede realizar ningún movimiento por lo que se procede a saltear su turno.
* Cambio en los nombres de los movimientos disponibles; se estandarizaron.

Para la segunda parte de este obligatorio se decidió construir la heurística automáticamente utilizando computación evolutiva como metodología. La implementación fue realizada con JGAP.

Los individuos se representan mediante vectores de 6 números decimales, representando los siguientes conceptos:

* Factor cantidad de fichas del jugador
* Factor cantidad de fichas del oponente
* Factor cantidad de cuadrados parciales del jugador
* Factor cantidad de cuadrados parciales del oponente
* Factor cantidad de cuadrados completos del jugador
* Factor cantidad de cuadrados completos del oponente

La función de aptitud se define como la cantidad de partidas ganadas en un torneo contra otros agentes genéticos, agentes MiniMax c/poda alfa-beta y agentes aleatorios. Por lo tanto se busca maximizar esta función para ir obteniendo las heurísticas que más ganen.

La población se inicializa con 10 individuos iniciados al azar con factores que varían entre -1 y 1 de tal manera que los factores del jugador son positivos y los del oponente son negativos. Esto se definió así porque como es de esperar, los factores que pertenecen al jugador deben tener un impacto positivo en la heurística así como los factores del oponente deben impactar negativamente.

Con respecto a los operadores genéticos, se utilizó la configuración por defecto de JGAP. La evolución termina luego de 20 generaciones y en última instancia se realiza un torneo con los mejores de la última generación para obtener la mejor heurística. Por último se realiza una partida contra un agente miniMax c/poda alfa-beta para testear la efectividad de la heurística obtenida.