

Diferentes funciones de interés para la heurística

- a) cómo saber si se trata de un movimiento asociado a comer ficha: la siguiente función devuelve true si la acción que origina el estado actual fue de comer ficha.

```
estado.isEatingMove()
```

Y sabiendo esto podemos además analizar si la pieza comida es una de las nuestras o no, para penalizar, si queréis, estos movimientos. Las funciones necesarias se describen a continuación.

- b) determinación de las piezas comidas en la última acción.

Se asume que actual es el objeto Parchis que se está considerando. Se observa que la función eatenPiece devuelve un par de valores: el primero es el color y el segundo es el identificador de la pieza (para cada color hay 2 piezas y los ids serán 0 y 1):

```
// se obtienen las piezas comidas con la ultima accion  
pair<color, int> piezaComida = actual.eatenPiece();
```

- c) cálculo de la distancia de la pieza comida, cuya información se almacena en la variable piezaComida de la función previa:

```
// se determina la distancia a meta de la pieza comida  
int distancia = actual.distanceToGoal(piezaComida.first,  
                                       piezaComida.second);
```

Se observa que el primer argumento de la función distanceToGoal es el color y el segundo el id. de la ficha. Con esto puedo valorar de forma diferente si la pieza comida está cerca de meta o no y tenerlo en cuenta para la valoración.

- d) cómo saber si la pieza comida es propia. Para ello se obtienen los colores del jugador que se evalúa (para ello se usa el id del jugador usado, el que se recibe como argumento en las llamadas de minimax o de la poda alfa - beta)

```
// se determina si es de uno de los colores propios  
vector<color> colores = actual.getPlayerColors(jugador);
```

y ahora se comprueba si se corresponde con el de la pieza comida (se usa la misma variable indicada antes)

```
// se comprueba si color de la pieza comida es de los propios  
bool propia = false;  
for(int i=0; i < colores.size() && !propia; i++){  
    if(colores.at(i) == piezaComida.first){  
        propia = true;  
    }  
}
```

si la pieza es propia puedo tenerlo en cuenta para restar puntuación a la jugada.