

A través de la funció millorTirada cridem la funció minimax la qual requereix de 4 arguments (la profunditat, l'alpha "-10000", la beta "10000" i si vol el valor més alt o el més baix (isMaximizing)

```
public Tirada millorTirada(){
    Tirada millorTirada = null;
    Integer bestScore = 0;
    Integer score = null;
    ArrayList<Tirada> res = new ArrayList<>();
    res = obtenirTiradesPossiblesJugador(_torn);
    Boolean isMaximizing = null;

    if(_torn == 1) {
        isMaximizing = true;
        bestScore = -9999;
    }
    else if (_torn == -1) {
        isMaximizing = false;
        bestScore = 9999;
    }

    for(Tirada tirada : res){
        aplicaMoviment(tirada,true);
        score = minimax(_maxDepthAI-1,-10000,10000, isMaximizing);
        desferTirada(tirada);
        if(isMaximizing){
            if(score > bestScore){
                bestScore = score;
                millorTirada = tirada;
            }
        }
        else if (!isMaximizing){
            if(score < bestScore){
                bestScore = score;
                millorTirada = tirada;
            }
        }
    }

    return millorTirada;
}
```

Amb l'ajuda de la funció obtenirTiradesPossiblesJugador() aconseguim totes les tirades possibles que pot fer el jugador que esta tirant, un cop obtingudes passem a traves de totes ells i les apliquem al tauler a través d'un for i cridem la funció minimax.

L'alpha i la beta s'utilitzen per poder descartar tirades (podar l'arbre) si el resultat que n'obtindriem seria indiferent i així fer més ràpida la recursió.