Let´s Build AI

By Adam Czerwinski

Robotics Engineer – AI Specialist

What is what ??!

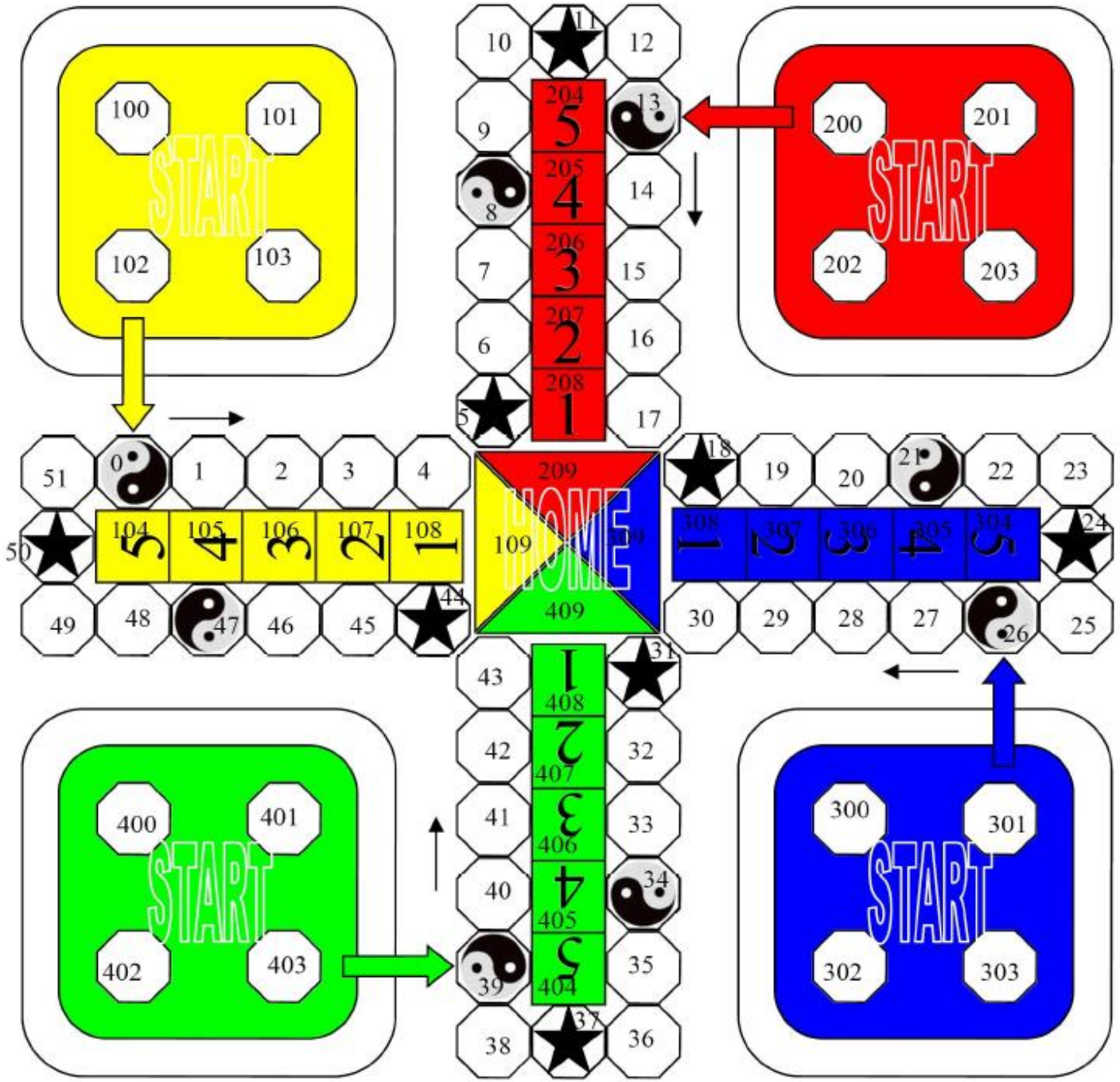
An “Intelligent” agent

Deducing the LUDO Game ☺

Relating the deduction onto the AI agent

Let’s Create !

What to consider ?? :D



```
public void play() {
    board.print("Random player playing");
    board.rollDice();
    int nr=-1;
    double best = 0;
    for(int i=0;i<4;i++) // find a random moveable brick
    {
        if(board.moveable(i)) {
            double temp = rand.nextDouble();
             if(temp>best) {
                 best = temp;
                 nr = i;
             }
        }
    }
    if(nr!=-1) board.moveBrick(nr);
    //else nothing to do - no moveable bricks
}
```

Interface Summary	
LUDOPlayer	Interface which any automatic ludo player must implement.

Class Summary	
AggressiveLUDOPlayer	Example of automatic LUDO player
FifoLUDOPlayer	Example of automatic LUDO player
LUDO	Main class the LUDO simulator - "controls" the game.
LUDOBoard	The LUDOBoard class is the core class of the LUDO simulator.
ManualLUDOPlayer	Example of automatic LUDO player
PacifisticLUDOPlayer	Example of automatic LUDO player
RandomLUDOPlayer	Example of automatic LUDO player
SemiSmartLUDOPlayer	Example of automatic LUDO player

Take 10 before

the workshop ☺

1) Unzip the LUDO.zip in your eclipse workspace folder (*e.g./home/User/workspace*)

2) Open eclipse

3) Press the “new” button (Most left button in bar)

4) Select General -> Project and press Next

5) Type project name e.g. LUDO and press Finish

6) In project explorer navigate to “*LUDO/src/LUDOSimulator/LUDO*”

7) Press “Run” (round green button with a play icon in bar)

8) Play a game of LUDO and see how it works ☺

9) For more information about the classes and examples of implementing a new player see the LUDO-Simulator.pdf

LUDO-AI/src/LUDOSimulator/SemiSmartLUDOPlayer.java

```
1   public float analyzeBrickSituation(int i) {
2           if(board.moveable(i)) {
3               int[][] current_board = board.getBoardState();
4               int[][] new_board = board.getNewBoardState(i, board.getMyColor(), board.getDice());
5
6               if(hitOpponentHome(current_board,new_board)) {
7                   return 5+rand.nextFloat();
8               }
9
10              //*****Add more code here *******/////
11
12
13              else {
14                  return 1+rand.nextFloat();
15              }
16          }
17          else {
18              return 0;
19          }
20      }
```

Method Summary	
boolean	**almostHome**(int index, int color) If a given index corresponding to color are in colored(safe) area close to home.
boolean	**atField**(int index) if a given index is at the field(white) area.
boolean	**atHome**(int index, int color) If index corresponding to color are in home area(brick completed game).

boolean	**inStartArea**(int index, int color) If brick corresponding to color and nr are in starting area.
boolean	**isDone**(int color) If all bricks of a particular color is home(game completed)
boolean	**isGlobe**(int index) if index is a globe
boolean	**isStar**(int index) if index is a star

Boolean | hitOpponentHome(current_board,new_board)

Let’s Build !

```
    else if(hitMySelfHome(current_board,new_board)) {
    return (float)0.1;
}
else if(board.isStar(new_board[board.getMyColor()][i])) {
    return 4+rand.nextFloat();
}
else if(moveOut(current_board,new_board)) {
    return 3+rand.nextFloat();
}
else if(board.atHome(new_board[board.getMyColor()][i],board.getMyColor())) {
    return 2+rand.nextFloat();
}
```

Play!

Test your AI Agent ☺