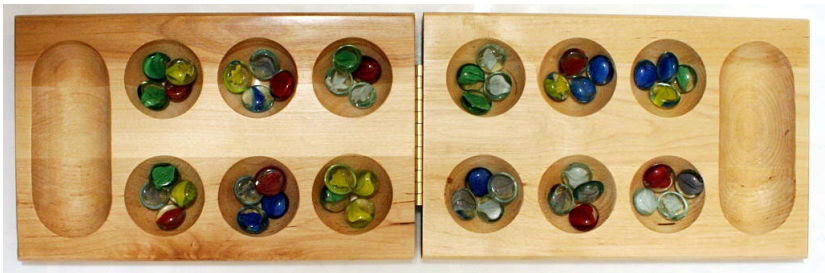
Assignment 4

Mancala is one of the oldest known board games that are still widely played today. It is a two-players turn-based strategy game which consist of obtaining the major number of stones into your storage. An image of the game can be seen below.



The object of the game is to capture more stones than your opponent.

How to play:

1. The Mancala board is made up of two rows of six holes, or pits, each.
2. Four stones are placed in each of the 12 holes.
3. Each player has a store (called a Mancala) to the right side of the Mancala board.
4. The game begins with one player picking up all of the pieces in any one of the holes on their side.
5. Moving counter-clockwise, the player deposits one of the stones in each hole until the stones run out.
6. If you run into your own store, deposit one piece in it. If you run into your opponent's store, skip it.
7. If the last piece you drop is in your own store, you get a free turn.
8. If the last piece you drop is in an empty hole on your side, you capture that piece and any pieces in the hole directly opposite.
9. Always place all captured pieces in your store.
10. The game ends when all six spaces on one side of the Mancala board are empty.
11. The player who still has pieces on his side of the board when the game ends capture all of those pieces.
12. Count all the pieces in each store. The winner is the player with the most pieces.

The rules are taken from: https://www.thesprucecrafts.com/how-to-play-mancala-409424. There is also a video with the rules.

What to do: Your assignment now is to apply Minimax algorithm with your own utility function to play against three different bots **(5 points)**:

- getA3Player (if you win, you get a 3)
- getA4Player (if you win, you get a 4)
- getA5Player (if you win, you get a 5)

Your report has to cover the key parts as follows:

1. Explain the used utility function. Additionally, you have to add the code of the utility function to the report, explaining it with comments.

How to use the given code:

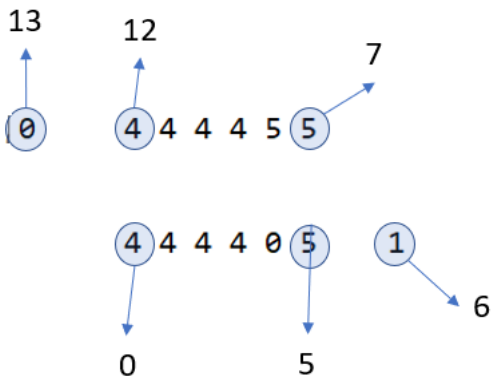
- Your bot will receive the information of the board (Number of stones on each whole) and the player turn (which part of the board is yours). Then, it will return which whole you are emptying. The variables are:
    - board: Information of the board.
    - playerTurn: 1, you can empty from position 0 to 5 (1 to 6 in matlab). 2, you can empty form position 7 to 12 (8 to 13 in matlab).
    - move: Which move will you make. Always from 1 to 6 independently from the playerTurn.
- When trying your bot, the first step is to run "Mancala". Within 10 seconds, you have to run both players. If, for some reason does not work, run it again. If after 3 times does not work, then talk to the teacher.
- When playing a match, a folder will be created with the information of the different games.
- Here you can change your player name:

```
25  # VARIABLES
26  playerName = 'Your player name'
```

- Here you have to change the code:

```
73  # Using your intelligent bot, assign a move to "move"
74  #
75  # example: move = '1';  Possible moves from '1' to '6' if the game's rules allows those moves.
76  # TODO: Change this
77  ################
78  move = '0'
79  ################
80  send(s, move)
81
```

- When printing the game, you can see the indexes in the following position:



- If the C# code does not work change the following lines of code:
  IPHostEntry ipHost = Dns.GetHostEntry("127.0.0.1");
  IPAddress ipAddr = ipHost.AddressList[0];
  To → IPAddress ipAddr = IPAddress.Parse("127.0.0.1");

Pre-conditions in order to present your code:

- Your Minimax algorithm is allowed to check up to d = 3 (meaning that you can check your first action, the first action of the opponent and your second action). After that, you have to use the utility function. Example moves:
    - MAX, MIN, MAX
    - MAX, MAX, MAX
    - MAX, MIN, MIN
    - …
- You can only start from the provided code, meaning that only Matlab, python or C# can be used for this assignment.
- **Change your playerName to "Name_LastName"**
- The code is prepared so that your bot has to give a desired movement in less than 0.2 seconds. The match consists of 10 games.
- If you win against getA3Player and getA5Player, but lose against getA4Player, you get a 3.
- In order to win against a player, you have to have more won games than lost games (I can run the code several times and you have to always win).
- You will not pass if you just code a random generator. So, you have to propose an utility function.

Conditions to approve the assignment 4

In order to receive the points for the code, you need to submit the report and once you have submitted and it is correct, you can get the points from the assignment.

- Minimum score to pass assignment 4: 3 points.

SCORES 3-5

- Your score < 3: U
- 3 <= your score <= 3.75: 3
- 3.75 < your score < 4.75: 4
- 4.75 <= your score: 5

SCORES A-F

- Your score < 3: F
- 3 <= your score < 3.4: E
- 3.4 <= your score < 3.8: D
- 3.8 <= your score < 4.2: C
- 4.2 <= your score < 4.6: B
- 4.6 <= your score: A

SUBMISSION DETAILS

Maximum Score: 5 points.

Time to submit: 20 January – 20 March & 26 April & 31 May

I will not consider a submission without the following format:

File names for the submission:

StudentName_StudentLastName_Assignment4_Code.zip (Only by email)

StudentName_StudentLastName_Assignment4_Report.pdf (Only in canvas)

Email subject for the submission:

[DVA340] StudentName StudentLastName Assignment4 Code

ADDITIONAL INFORMATION!

You are allowed to send each part of the assignment a maximum of three times. When you approve the assignment, you are not allowed to submit the assignment again.

Before defending the code to Miguel Leon, you have to send the code by email to Miguel.leonortiz@mdh.se.

Before submitting the report, you should present this assignment first to Miguel Leon. After that, and only if everything is correct, you are able to submit the report in canvas.

You cannot get the points of the report without presenting the code first.

Extra information

Use the code given in: “assignment 4 files.zip”.