

COMP5028 - Assignment 1

Question 1:

Analysing the requirements:

How to analyses requirements:

The first step is reading the whole document and find out what are common things, for example (what are ordinary things that may exist in the game of Othello).

- board, white and black stones, and 64 stones, and two players, and different colours, and some rules, and so on.

The send step is discerning the essence of the items. In the Othello, the following items are necessary:

- game board, with 8X8 grids
- two players - each player chooses own unique colour(white/black), and takes 32 stones.
- each player is able to move the stone but has to follow rules.
- when the game starts what the initial configuration is.
- once a player's disc has been placed on the square and the disc is able to outflank the foe's discs, then those discs must be flipped.
- some special rules:
 - black discs move first
 - checking whether a player has chance to move
 - allowing players to move their stones in multiple directions.
 - checking player whether against that he skips own colour disc to outflank enemy discs
 - checking that whether has discs that may only be outflanked as a direct result of a move and must fall in the direct line of the disc placed down
 - checking a player whether flips disc.
 - once disc placed on a square, then it cannot move another square
 - if player still has chance to move the disc even if he out of discs
 - this function is checking that if either player no longer to move disc, then finds out winner who has majority colour discs left.

The step three is modelling, where been displayed on below **modelling section**.

OBJECTS:

A 8X8 grids(board), white/black(circles), players, 32 discs/player, colour, rules

FUNCTIONS:

```
/****** Common functions *****/  
Board(); //implementing the its class function - +InitialPlace(); and +DrawBoard();  
  
InitialPlace(); // initial whole board when the game starts
```

SID: 440042783

STN: XUN ZHAO

```
DrawBoard(); // Drawing the graph of board
```

```
/****** Controller *****/
```

```
move(int x, int y, String colour); //movement function that allowed players to move their stone into a exact coordinate.
```

```
inputCoordinate(); // request player to input cordite
```

```
placeInfo(); // display the info that after user inputed
```

```
/****** Player rules *****/
```

```
initialMove(); // to determiner the player who has black stone
```

```
outflanking(int Ox, int Oy, int Ex, int Ey); // this function is outflanking enemy discs, and overturning the outflanked disc(s) to Own colour.
```

```
outflank_and_flip_check(); //this function is checking whether the player has chance to move.
```

```
multi_outflank(); // this function is allowed player to outflank any number of discs in multiple directions, such as in horizontally, vertically or diagonally
```

```
skip_own_color_check(); //this function is check whether player has against that he skips own colour disc to outflank enemy discs
```

```
specialCondition_outflank_check(); //this function checks that whether has discs that may only be outflanked as a direct result of a move and must fall in the direct line of the disc placed down
```

```
flipped_check(); //this function is checking that player whether flips disc.
```

```
cannot_move_check(); //once disc placed on a square, then it cannot move another square
```

```
opportunity_check(); //if player still has chance to move the disc even if he out of discs
```

```
winner_check(); //this function is checking that if either player no longer to move disc, then finds out winner who has majority colour discs left.
```

Modelling (Step three)



