

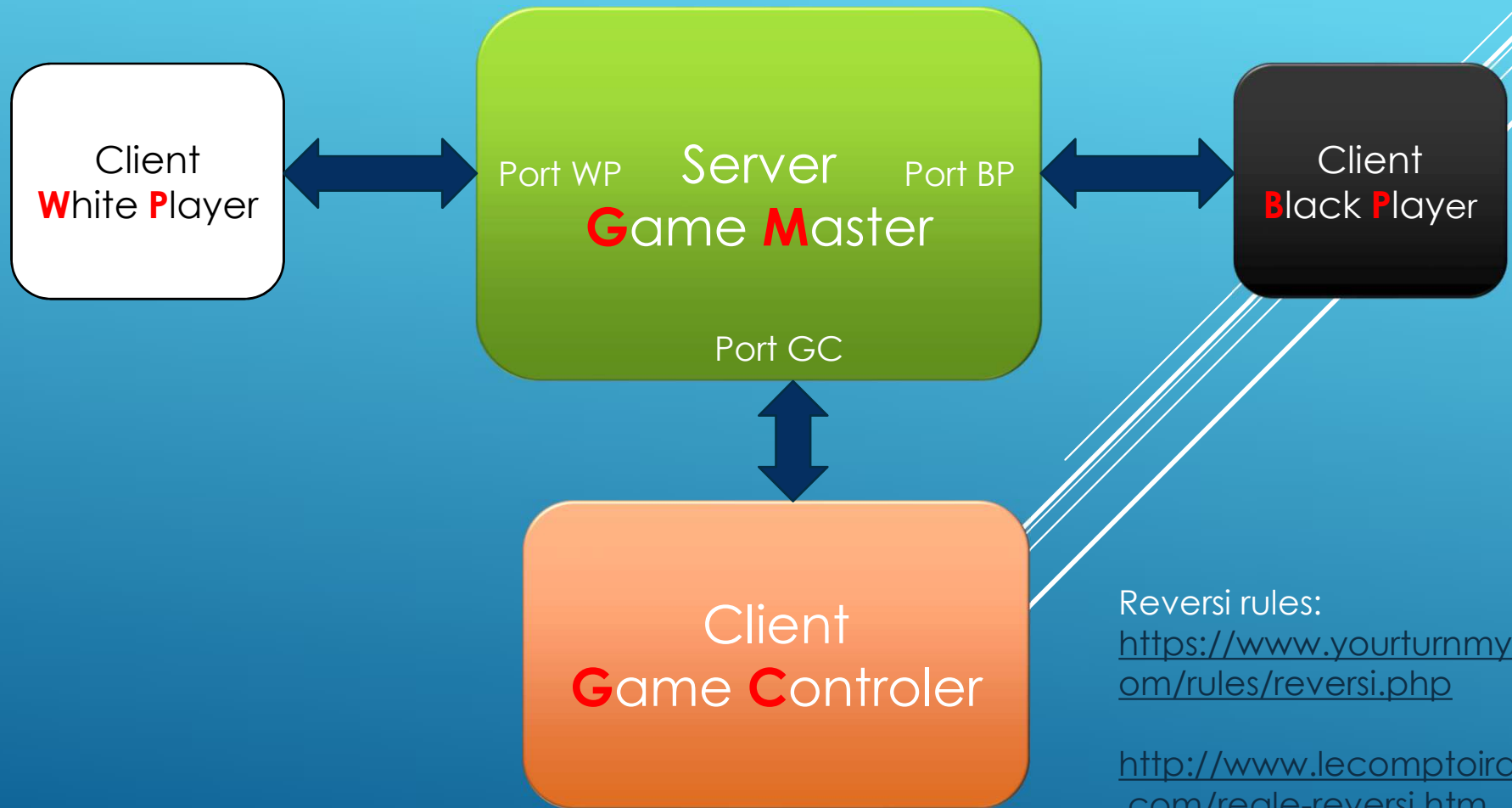
Project : **Reversi**

Subprojects:

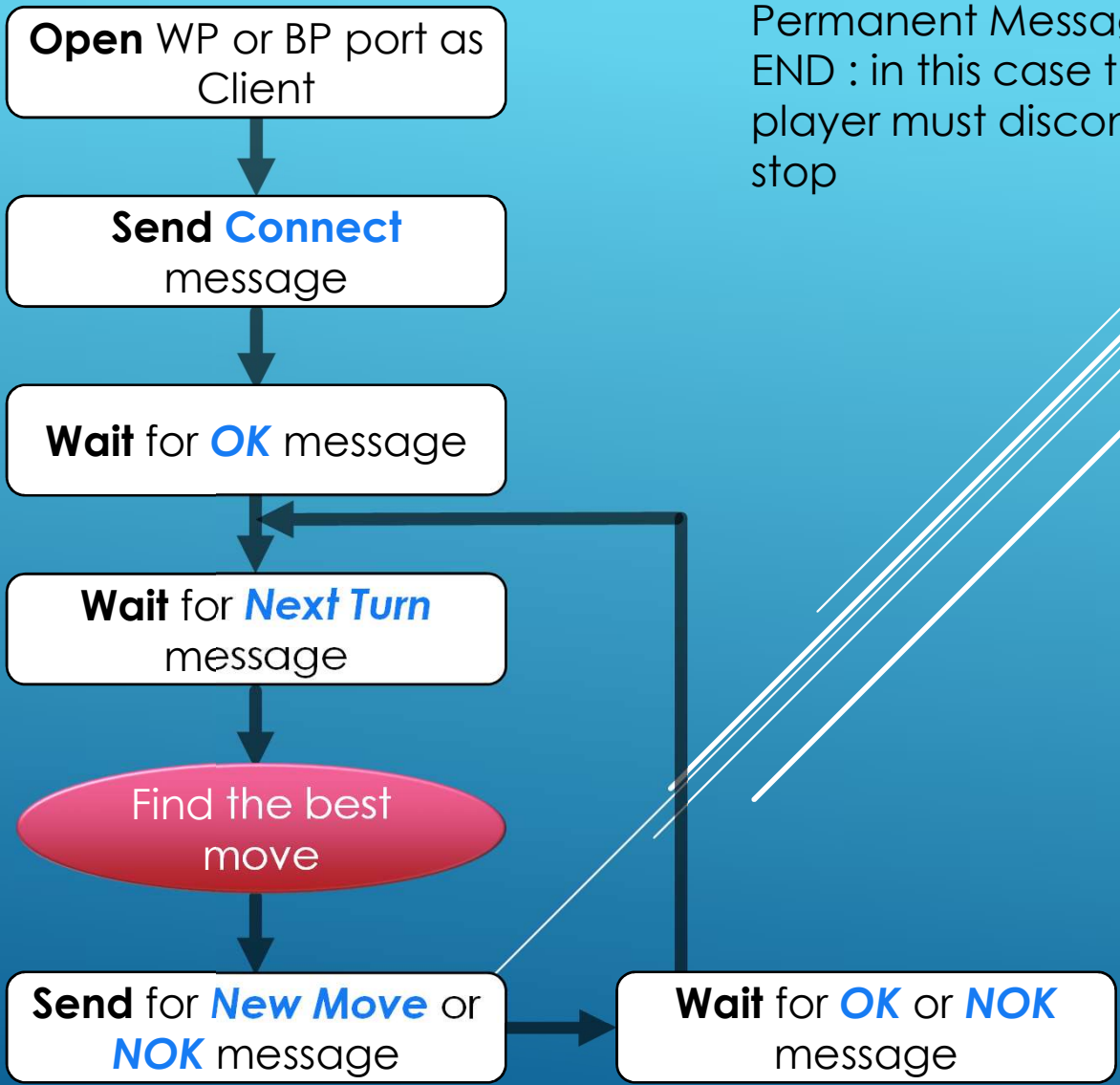
- **Game Master : 2 players**
- **Black & White Player : 2x2 players**
- **Game Controller : 2 players**

“Shall we play a game?”

Reversi

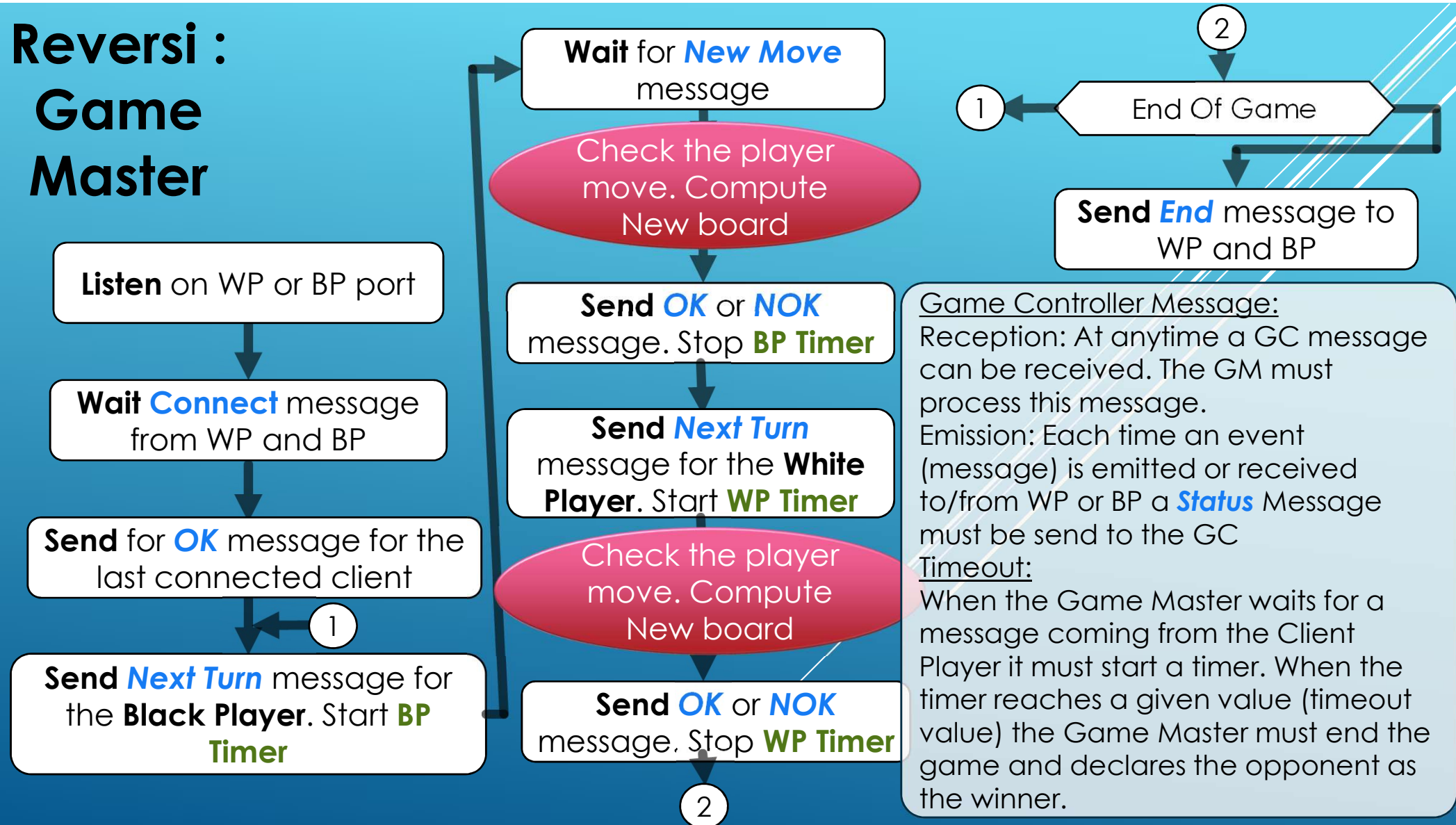


Reversi : Client Player



Permanent Messages:
END : in this case the client
player must disconnect and
stop

Reversi : Game Master



Reversi : Game Controller

ASCII Mode:

Black Player : Philip Blake

Last Move : D3

Points: 3

Time: 20 ms

White Player : Rick Grimes

Last Move : C3

Points : 3

Time : 18 ms

	A	B	C	D	E	F	G	H
1	N	N	N	N	N	N	N	N
2	N	N	N	N	N	N	N	N
3	N	N	W	B	N	N	N	N
4	N	N	N	W	B	N	N	N
5	N	N	N	B	W	N	N	N
6	N	N	N	N	N	N	N	N
7	N	N	N	N	N	N	N	N
8	N	N	N	N	N	N	N	N

Commands: (s) step by step mode

(c) Continuous mode

(ch) : change board size

(q) : quit

(r) restart

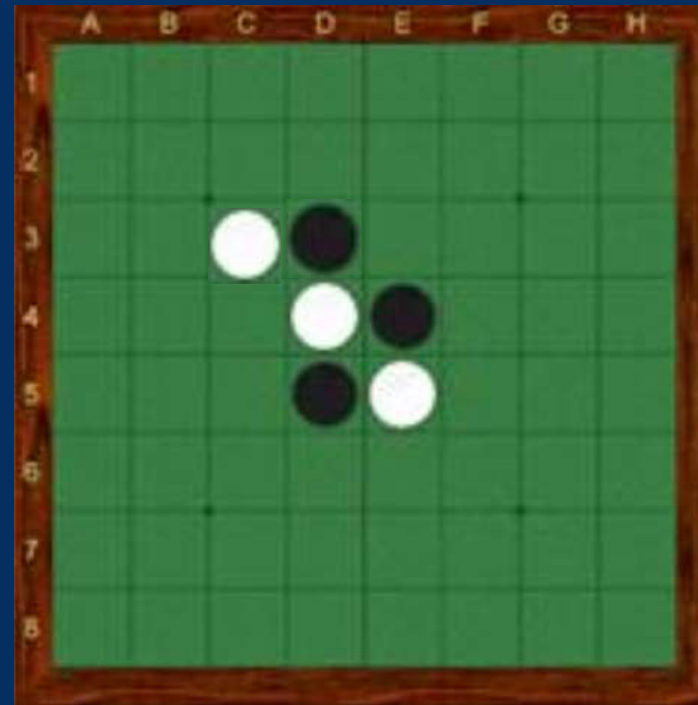
The Game Controller allows to visualize and control the game. Two modes are available ASCII or GRAPHIC.

Black Player:
Philip Blake
Last Move:D3
Points: 3
Time: 20ms

Continuous
Mode

Step by
Step Mode

Board Size



White Player:
Rick Grimes
Last Move:C3
Points: 3
Time: 18ms

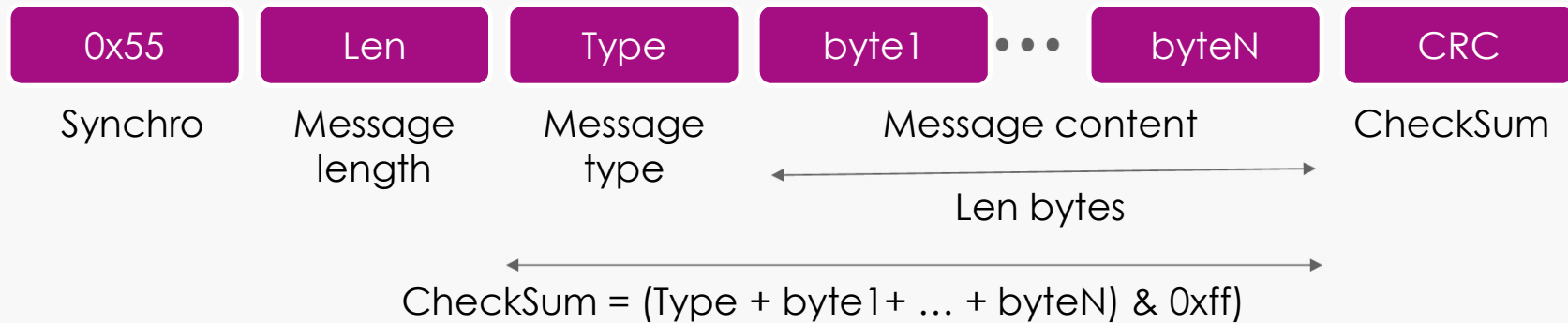
Quit

(re)Start

Reversi : Messages (1)

Global form

 = 1 byte



Reversi : Messages (2) Types (1)

= Type

Connect Message:

0x01

NameLen

NameByte1

...

NamebyteN

Player OK Message:

0x10

OK Black=0x01

OK White=0x02

OK/NOK Message:

0x02

OK=0x01

NOK=0x00

New Move Message:

0x03

X

Y

End Message:

0x04

Reversi : Messages (3)

Types (2)

 = Type

Next Turn Message:

0x05

LastMove X

LastMove Y

Board Size X

Board Size Y

Board State

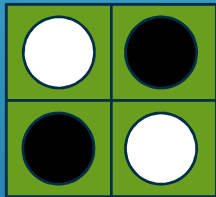
...

Board State N

Board State format :

Cell state coded on 2 bits : 00=empty cell, 01=Black, 10= White, 11=not used

1 byte can code 4 cells (8bits/2bits). Board is coded by scanning horizontal lines from left to right and from top to bottom. Example board size = 2x2 :



Only one byte is needed to code this board :

coodonates : white(0,0),black(0,1),white(0,0),black(0,1)

Bit code : 10,01,01,10

Byte= 10010110 = 0x96

0x05

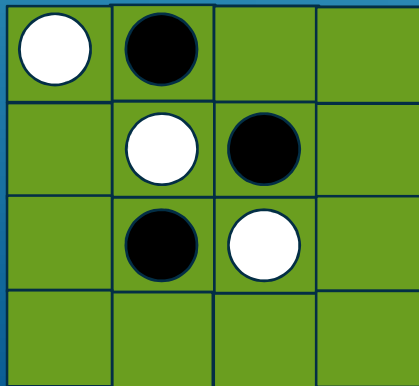
-

-

2

2

0x96



4 bytes are needed to code this board :

coodonates : white(0,0),black(0,1),empty(0,2), empty(0,3), empty(1,0), white(1,1), black(1,2), empty(1,3),empty(2,0), black(2,1), white(2,2), empty(2,3), empty(3,0), empty(3,1), empty(3,2), empty(3,3)

Bit code : 10,01,00,00,00,10,01,00,00,01,10,00,00,00,00,00

Byte= 10010000 00100100 00011000 00000000 = 0x90 0x24 0x18 0x00

0x05

-

-

4

4

0x90

0x24

0x18

0x00

Reversi : Messages (4) Types (3)

 = Type

Status1 Message (same as Next Turn Message) :

0x06

LastMove X

LastMove Y

Board Size X

Board Size Y

Board State

... Board State N

Status2 Message :

0x07

BP Points

BP Timer
MSB

BP Timer
LSB

BP Name
Length

BP Name
byte1

... **BP** Name
byteN

WP Points

WP Timer
MSB

WP Timer
LSB

WP Name
Length

WP Name
byte1

... **WP** Name
byteN

Reversi : Messages (5) Types (4)

 = Type

Control Message :

 0x08  New Board
Size X  New Board
Size Y  Game
Mode  restart

New Board Size :

Dimension size of the new board. This new dimension will be taken into account after the next or current **RESTART**. 0 and 0 means no new dimension.

Game Mode:

0x00 : Continuous mode

0x01 : Step by step mode

Restart:

0x00 : no restart

0x01 : restart the game with the new dimensions

Reversi : Some advises

Try to have the 3 components (GM, GC and players) as fast as possible in order to test your program with the other.

Identify and register all (communication) bugs coming from the others teams.

Keep an history of your code versions.

Game Master:

First develop the most simple Game Master:

- Open the 3 ports as server (BP, WP, GC).
- Develop a "yes" move checker (no verification).
- Implements the messages generator/parser.
- Check the connection with the GC, WP, BP.

Game Player:

- Implement a stupid Gamer : He plays the first possible move.
- Check the connection with the Game Master

Game Controler:

- Develop an ASCII controller which is able to display the game status