Project : ReversIA

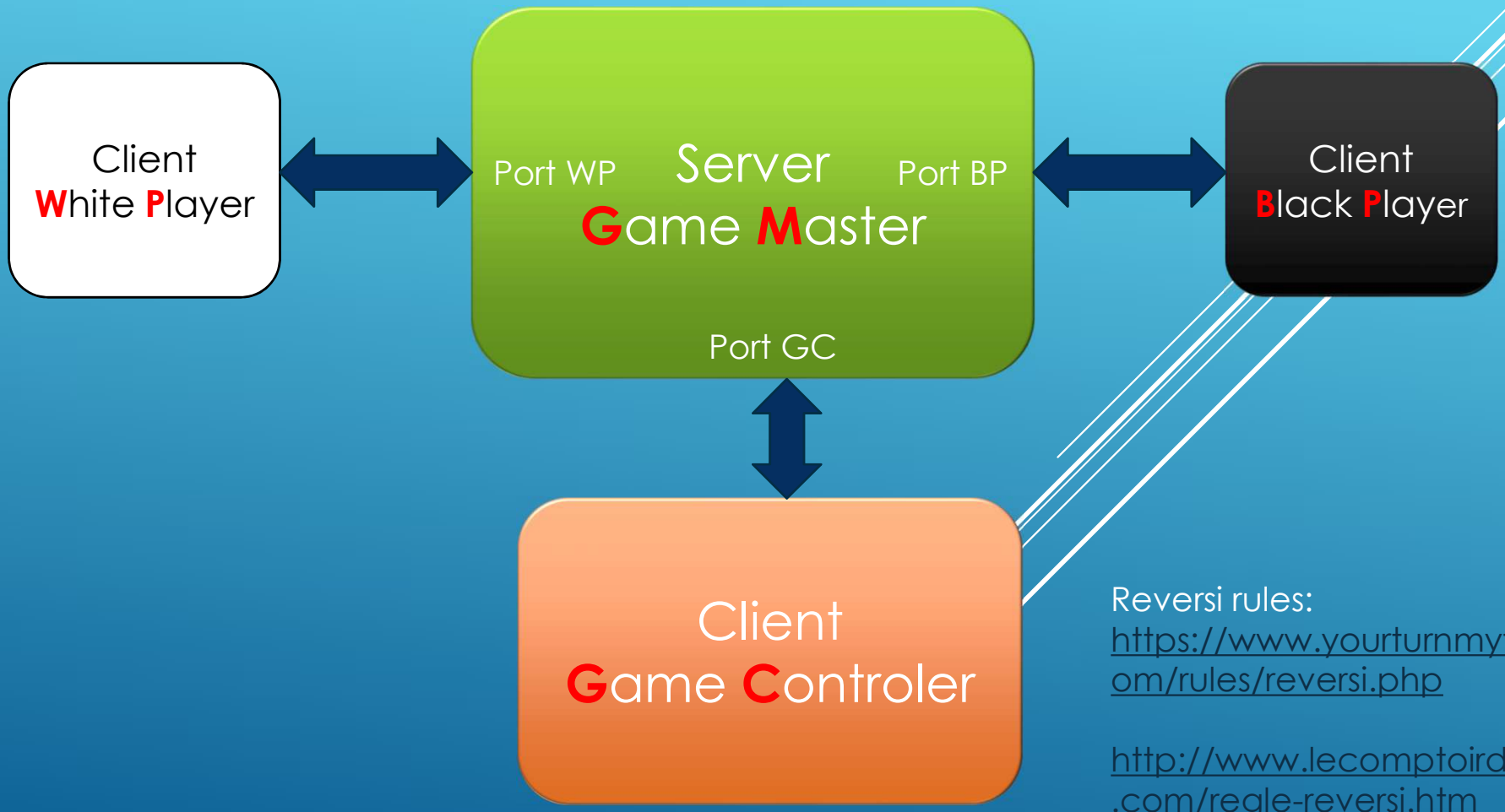
Subprojects:

- Game Master
- Game Player
- Game Controler

“Shall we play a game?” (Joshua)
https://youtu.be/-1F7vaNP9w0

Alain Houelle
V1.5

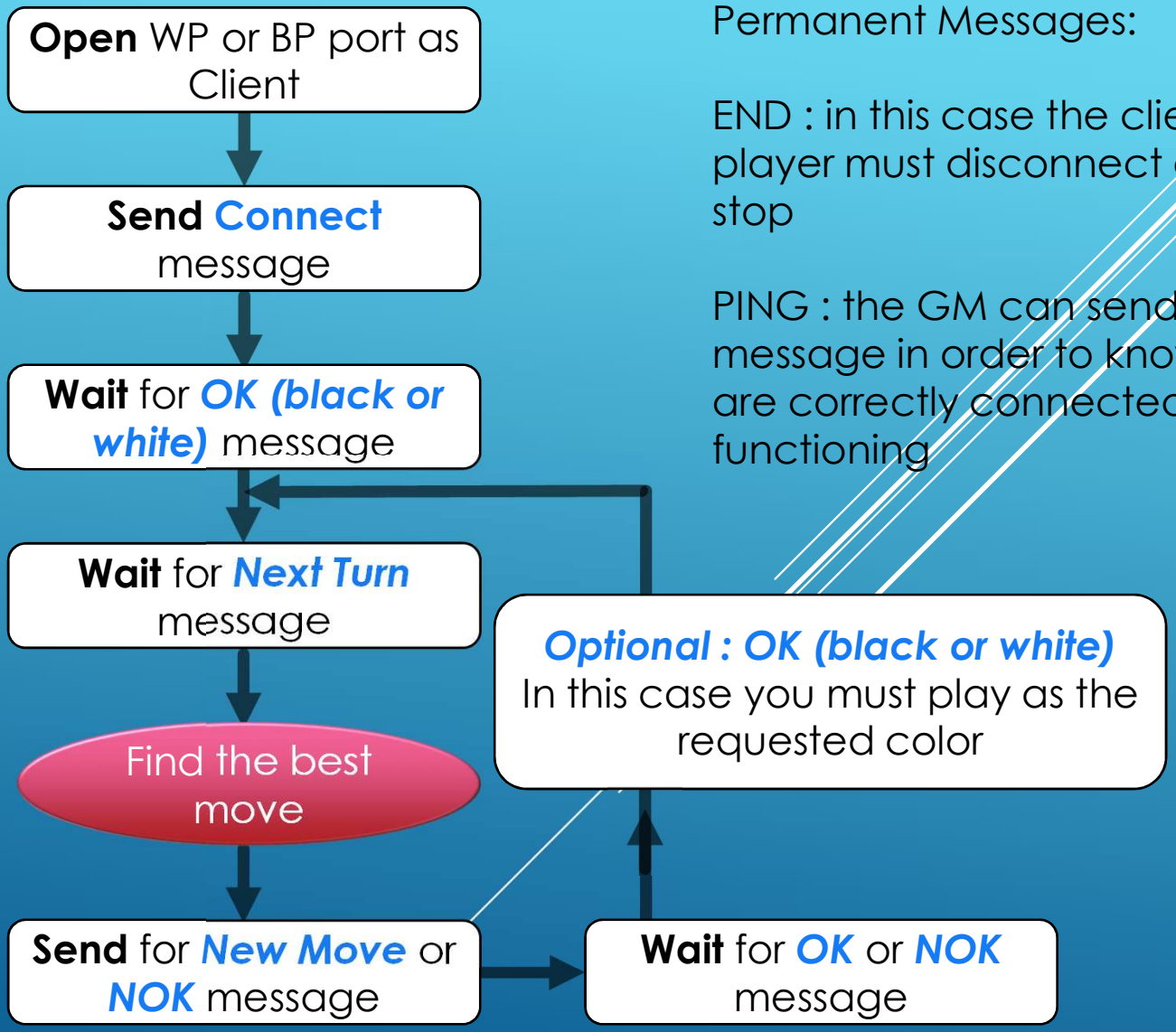
Reversi



Reversi rules:
https://www.yourturnmyturn.com/rules/reversi.php

http://www.lecomptoirdesjeux.com/regle-reversi.htm

Reversi : Client Player



Permanent Messages:

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

PING : the GM can send you this message in order to know if you are correctly connected and functioning

The player must output (printf) every events (move, connection, disconnection, color change, etc…). This output will be used to debug and track the player behavior.

Reversi : Game Master

Listen on WP or BP port

Wait **Connect** message from WP and BP

Send for ***OK (black/white)*** message for the last connected client

1

Send ***Next Turn*** message for the **Black Player**. Start **BP Timer**

Wait for ***New Move*** message

Check the player move. Compute New board

Send ***OK*** or ***NOK*** message. Stop **BP Timer**

Send ***Next Turn*** message for the **White Player**. Start **WP Timer**

Check the player move. Compute New board

Send ***OK*** or ***NOK*** message. Stop **WP Timer**

2

2

1

End Of Game

Send ***End*** message to WP and BP

Game Controller Message:
Reception: At anytime a GC message can be received. The GM must process this message.
Emission: Each time an event (message) is emitted or received to/from WP or BP a ***Status*** Message must be send to the GC
Timeout:
When the Game Master waits for a message coming from the Client Player it must start a timer. When the timer reaches a given value (timeout value) the Game Master must end the game and declares the opponent as the winner.

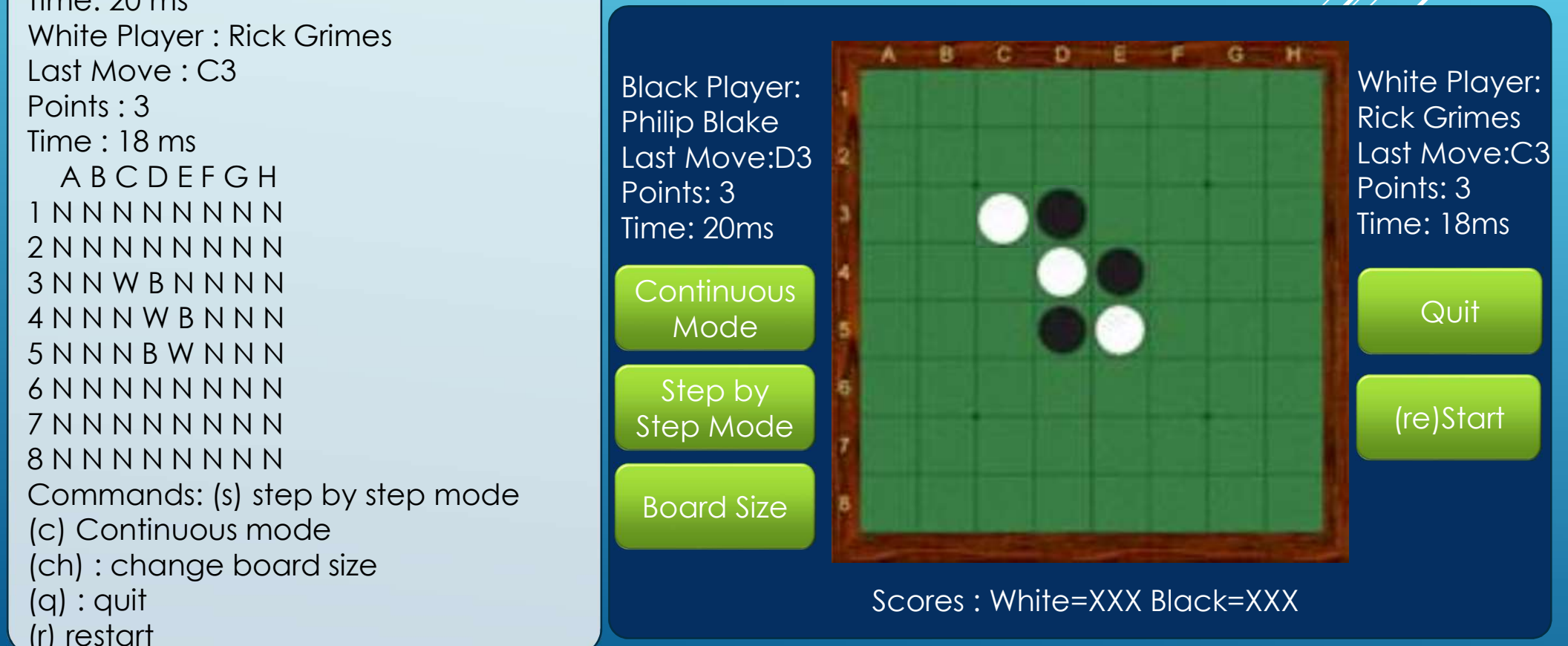
Reversi : Game Controler

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
```

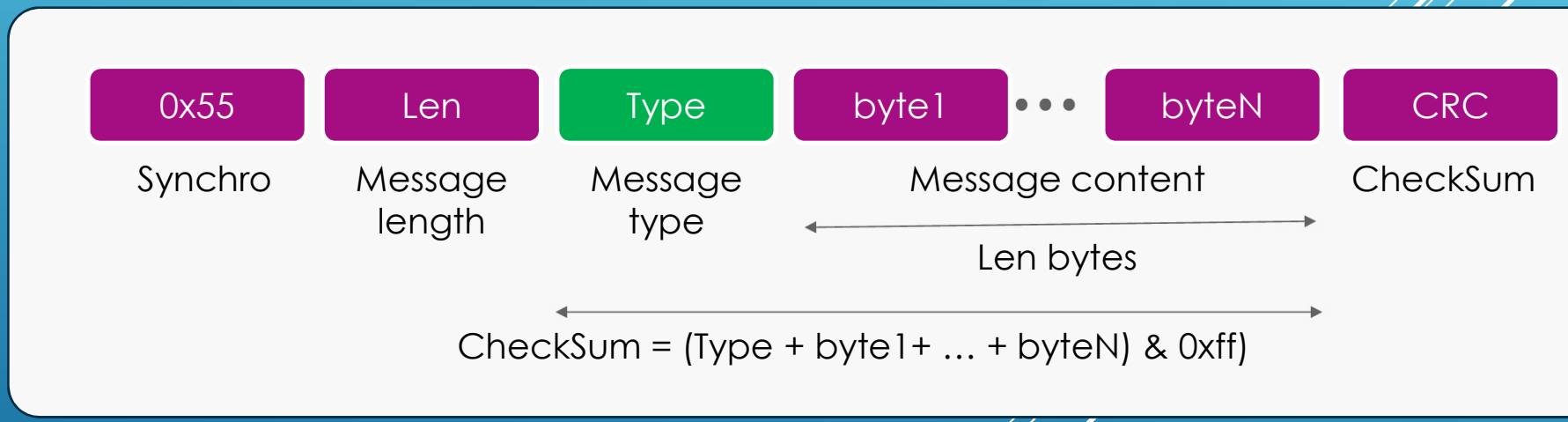
Commands: (s) step by step mode
(c) Continuous mode
(ch) : change board size
(q) : quit
(r) restart

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



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

End Message: 0x04

Ping Message:

0x11

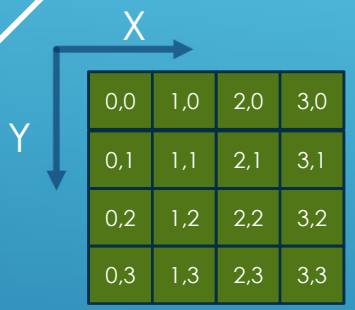
OK/NOK Message:

0x02 | OK=0x01

NOK=0x00

New Move Message:

0x03 | X | Y



When you receive a PING message you must send back a OK message to validate your connection.

If you have no message on a given socket since at least 1sec you must check the connection by sending a Ping message

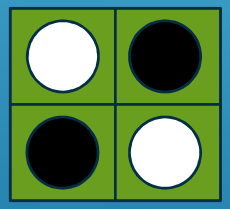
Reversi : Messages (3)
Types (2)

= Type

Next Turn Message:

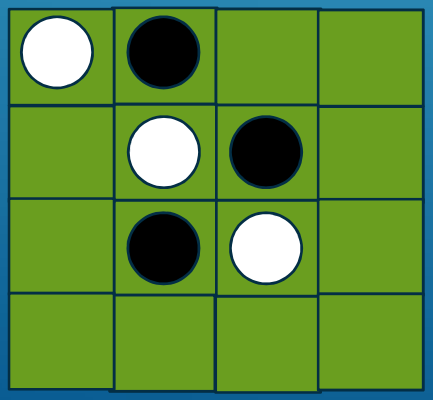
0x05	LastMove X	LastMove Y	Board Size X	Board Size Y	Board State	...	Board State N

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)

Messages from GM to GC

= 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** Score MSB	**BP** Score LSB	**BP** Timeout (seconds)
	BP Name Length	**BP** Name byte1	...	**BP** Name byteN		
	WP Points	**WP** Timer MSB	**WP** Timer LSB	**WP** Score MSB	**WP** Score LSB	**WP** Timeout (seconds)
	WP Name Length	**WP** Name byte1	...	**WP** Name byteN		

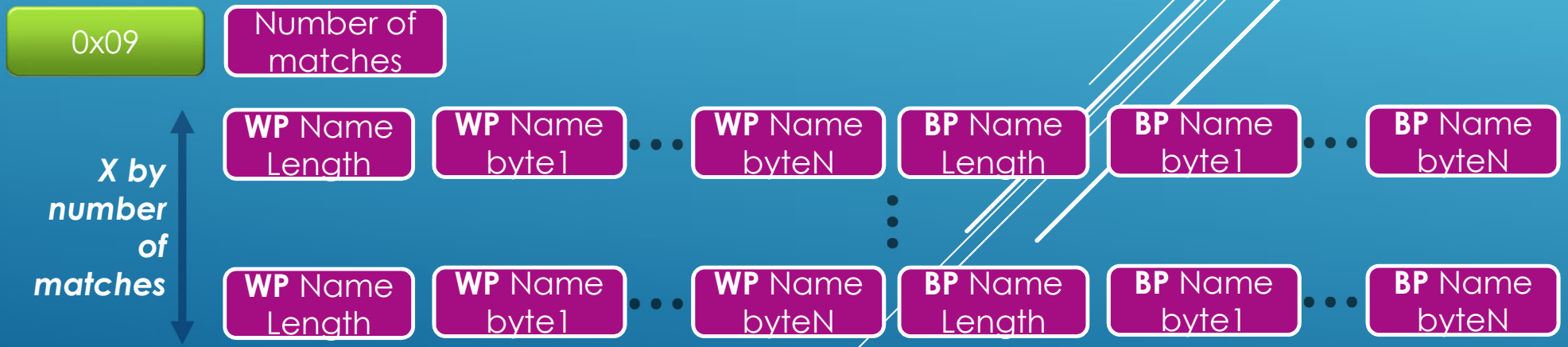
Reversi : Messages (5) Types (4)

= Type

Messages from GM to GC

Message Display Message (used to display a specific message on GC):

Next players to play (used to display the player list for the next matches):



Winner message :

Reversi : Messages (5)
Types (4)

= Type

Control Message :

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

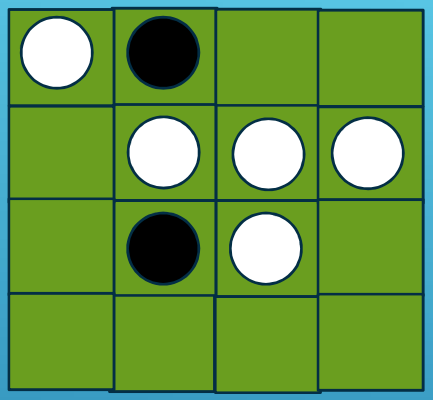
Try to have the 3 components (GM, GC and players) as fast as possible in order to test your program with the other.
Identify and register all (communication) bugs coming from the others teams.
Keep an history of your code versions.

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

Reversi : Score



White Cumulated Time:
8 seconds

Black Cumulated Time:
4 seconds

Black coins : 2
While Coins: 5

Black score = 3
White score = 8

If a player:
- Loses its socket connection (disconnect error)
- Doesn't respond quickly enough (timeout erro)
- Send a corrupted frame (frame and message errors)

He has 0 points. Its opponnent gains a « default win » : 3 points.

If a full game is played :
- The winner wins 8 points
- The « losers wins an « end of game » bonus : 3 points (points for the correct net code)

Bonuses (applicable only when a full game is played):
- If the game if filled with more than 75% of winner's coins, the winner gains a « Big Win » bonus : 3 points.
- The play time of each player is computed. The player with the lesser cumulated time wins a « Time » Bonus : 3 points

The greater possible score for one game is 14 :
win score(8)+ Big Win Bonus (3) + Time Bonus (3)

The lesser possible score for one game is 0:
Disconnect/timeout/frame/message error