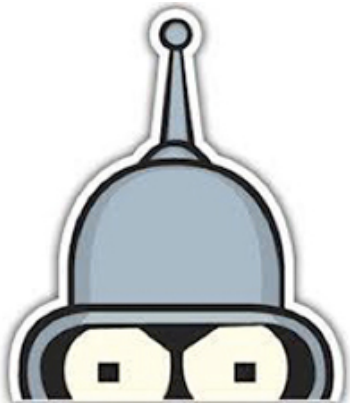
B2 - Stumpers

B-CPE-210

Connect4

Duo Stumper



2.0

{EPITECH.}

Connect4

binary name:	connect4
repository name:	CPE_duostumper_$STUMPERNUMBER_$ACADEMICYEAR
repository rights:	ramassage-tek
language:	C
compilation:	via Makefile, including re, clean and fclean rules

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

The $STUMPERNUMBER is always on one digit: `1`, `2`, ...

For this project, the **only** authorized functions are those of the standard `libC`.

Connect4 is a game where the goal is to line up four game pieces on a grid made up of x rows and y columns. Each player has different colored game pieces (here they will be represented by 'X' (uppercase) and 'O' (uppercase)).

With each round, players place a game piece in the column of their choice.
The game piece slides down said column until it can't go any further. Next, it's the opponent's turn to play.
The winner is the first player to line up (horizontally, diagonally, vertically) at least four of his/her own colored game pieces.
If all of the grid's spaces are filled in, and neither player manages to align their game pieces, the round will end in a tie.

Your program must be able to take several options as parameter:
-w WIDTH: defines the grid's width. Default: 7.
-h HEIGHT: defines the grid's height. Default: 6.
-p1 AVATAR: defines the caracter the represents player 1. Default: 'X'.
-p2 AVATAR: defines the caracter the represents player 2. Default: 'O'.
-a REFEREE: defines the character that represents a game piece in a winning position. Default: '#'

You must comply with the following rules:

- Player 1 always starts the round.

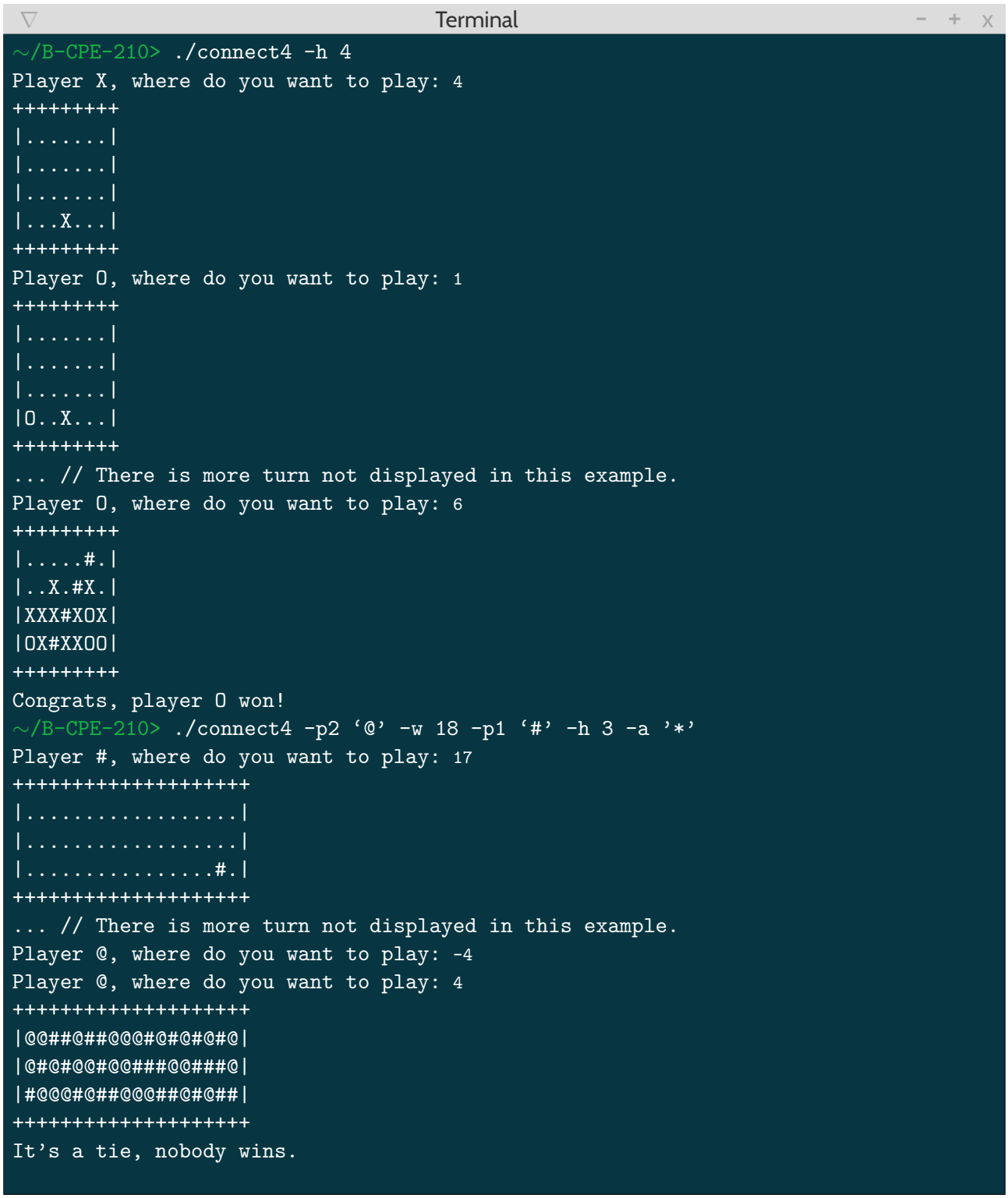
- During a winning game board configuration, the winning pieces must be displayed with the referee character.
- The columns' indexes go from 1 to 7 (or -w) columns.
- At each round, your program will display the grid and each player's game pieces' position. Then it will put a user's entry on hold. If the user enters a number that doesn't correspond to any columns, it must display the request message again.

- Your program must automatically detect if a player wins.

The grid size cannot exceed 80 columns and 16 rows. Also, the grid must have at least 20 slots. If you encounter one of these cases, display an error message and quit.
Both players cannot have the same avater (and the avatars cannot be the same as the referee). In these cases, display an error message and quit.

Terminal

```
~/B-CPE-210> cat -e moves
4$
1$
~/B-CPE-210> ./connect4 < moves
Player X, where do you want to play: +++++++++
|.......|
|.......|
|.......|
|.......|
|.......|
|...X...|
+++++++++
Player O, where do you want to play: +++++++++
|.......|
|.......|
|.......|
|.......|
|.......|
|O..X...|
+++++++++
Player X, where do you want to play: ~/B-CPE-210>
```


Terminal
~/B-CPE-210> ./connect4 -h 4
Player X, where do you want to play: 4
+++++++++
|.......|
|.......|
|.......|
|...X...|
+++++++++
Player O, where do you want to play: 1
+++++++++
|.......|
|.......|
|.......|
|O..X...|
+++++++++
... // There is more turn not displayed in this example.
Player O, where do you want to play: 6
+++++++++
|.....#.|
|..X.#X.|
|XXX#XOX|
|OX#XXOO|
+++++++++
Congrats, player O won!
~/B-CPE-210> ./connect4 -p2 ‘@’ -w 18 -p1 ‘#’ -h 3 -a ’*’
Player #, where do you want to play: 17
++++++++++++++++++++
|..................|
|..................|
|................#.|
++++++++++++++++++++
... // There is more turn not displayed in this example.
Player @, where do you want to play: -4
Player @, where do you want to play: 4
++++++++++++++++++++
|@@##@##@@@#@#@#@#@|
|@#@#@@#@@###@@###@|
|#@@@#@##@@@##@#@##|
++++++++++++++++++++
It’s a tie, nobody wins.