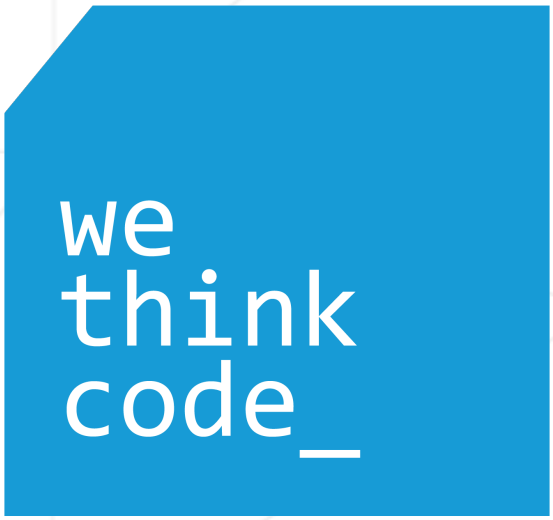

Filler

Pedago pedago@wethinkcode.co.za

Summary: Will you be the best?

Contents

Chapter I

Forewords

Many people are asking, “What are the seven deadly sins?” The seven deadly sins viewed by society and literature are:

- Lust (Luxuria in Latin) – to have an intense desire or need: “But I tell you that anyone who looks at a woman lustfully has already committed adultery with her in his heart” (Matthew 5:28).
- Gluttony (Gula in Latin) – excess in eating and drinking: “for drunkards and gluttons become poor, and drowsiness clothes them in rags” (Proverbs 23:21).
- Greed (avaritia in Latin)- excessive or reprehensible acquisitiveness: “Having lost all sensitivity, they have given themselves over to sensuality so as to indulge in every kind of impurity, with a continual lust for more” (Ephesians 4:19).
- Laziness (Acedia in Latin) – disinclined to activity or exertion: not energetic or vigorous: “The way of the sluggard is blocked with thorns, but the path of the upright is a highway” But it is mostly about the Laziness of the soul, boredom, to walk away from God, prayers. (Proverbs 15:19).
- Wrath (Ira in Latin) – Also known as Anger. Strong vengeful anger or indignation: “A gentle answer turns away wrath, but a harsh word stirs up anger” (Proverbs 15:1).
- Envy (Invidia in Latin) – painful or resentful awareness of an advantage enjoyed by another joined with a desire to possess the same advantage: “Therefore, rid yourselves of all malice and all deceit, hypocrisy, envy, and slander of every kind. Like newborn babies, crave pure spiritual milk, so that by it you may grow up in your salvation” (1 Peter 2:1-2).
- Pride (superbia in Latin)- quality or state of being proud – inordinate self esteem: “Pride goes before destruction, a haughty spirit before a fall” (Proverbs 16:18).

Chapter II

Introduction

Create your player to fight other students on the famous (or not) Filler board. The concept is simple: two players compete on a board and each must place, turn after turn, a token that the master of the game (given under the shape of an executable Ruby) gives them, earning in the process more points. The game stops as soon the token cannot be placed. A small fun project!

Chapter III

Goals

The goal of this project is to bring you to basic algorithm and to make you manipulate inputs/outputs.

Chapter IV

General Instructions

- You'll have to submit a file called `author` containing your username followed by a '\n' at the root of your repository.

```
$>cat -e author
xlogin$
```

- The executable file must be named `filler`.
- It must be at the root of the repository.
- You must submit a `Makefile`.
- Your `Makefile` must compile the project and must contain the usual rules. It must recompile and re-link the program only if necessary.
- If you are clever, you will use your library for your player. Submit also your folder `libft` including its own `Makefile` at the root of your repository. Your `Makefile` will have to compile the library, and then compile your project.
- Your project must be written in accordance with the Norm.
- You have to handle errors carefully. In no way can your program quit in an unexpected manner (Segmentation fault, bus error, double free, etc).
- Within your mandatory part, you are allowed to use the following functions:
  - `read`
  - `write`
  - `malloc`
  - `free`
  - `perror`
  - `strerror`
- Good luck and GOOD FIGHT to all!

Chapter V

Mandatory part

V.1 The Filler

- Two opponents will fight it out in this game. They play one after the other.
- The goal is to win the most points possible by filling the game board with as many elements as possible.
- The board is defined by X columns and N lines, it will then become X*N cells.
- You will receive your token at each turn.
- A token is defined by X columns and N lines, so it will be X*N cell. In each token, a shape of one or many cell is represented.
- To be able to place a token, it is mandatory that one, and only one cell of the shape covers a cell on a shape already placed.
- The shape must fit completely in the board.
- The board contains one first shape to initiate the game.
- The game will stop at the first mistake: as soon as a token cannot be placed or has been wrongly placed.

V.2 The Board

- A 14x30 board

```
Plateau 14 30:
012345678901234567890123456789
000 ..............................
001 ..............................
002 ..............................
003 ..............................
004 ..............................
005 ..............................
006 ..............................
007 ..............................
008 ..............O...............
009 ..............................
010 ..............................
011 ..............................
012 ..............................
013 ..............................
```

V.3 The tokens

- An example of a 4x7 token

```
Piece 4 7:
...*...
...*...
...*...
..***..
```

- An example of a 4x5 token

```
Piece 4 5:
.**..
.***.
..*..
.....
```

- An example of a 3x6 token

```
Piece 3 6:
.****.
**....
*.....
```

V.4 The Topic

V.4.1 The Player

- The executable that allows you to play the filler is accessible in the attachment to the topic.
- For this project, you will have to create a filler player. Your goal is to win:
  - It will read the board and the tokens on the standard output.
  - Each turn the filler displays the map with a new token to be placed.

- The player will have to put on the standard output the coordinates to place the token.
- The following format must be used “X Y\n”.
- Each token placed will earn you points.

```
Plateau 14 30:
012345678901234567890123456789
000 ..............................
001 ..............................
002 ..............................
003 ..............................
004 ..............................
005 ..............................
006 ..............................
007 ..............................
008 ..............O...............
009 ..............................
010 ..............................
011 ..............................
012 ..............................
013 ..............................
Piece 4 7:
...*...
...*...
...*...
..***..
```

Watch out! You must write the coordinates of the token and not those of the shape.

V.4.2 Multi Players

- Player number:
  - The first two lines of the filler will be in the following format:

```
$$$ exec pPLAYER_NUMBER : [PLAYER_NAME]
```

  - The filler will only send the line that concerns your program. You’ll have to get your player number.
  - If you are Player 1 your program will be represented by “o” and “O”. If you are Player 2, your program will be represented by “x” and “X”. The first step will be to get your player number.
  - The lowercases (“x” or “o”) will highlight the last token placed. For the following turns, it won’t be the last anymore, so it will then be represented in uppercases (“X” or “O”).
  - Each token placed will earn you points.
- How the game works

- At each turn, the filler will send the updated map and a new token to the player involved.
- The player involved will write on the standard output the coordinates to place his/her token.
- The filler will send the map and a new token to the other player.

V.4.3 How the game rolls

- Here is an example on how a game will roll out.

```
$>./filler_vm -p1 user1 -p2 user2 -v -f samples/w1.flr
$$$ exec p1 : [user1]
$$$ exec p2 : [user2]
Plateau 14 30:
01234567890123456789012345678 9
000 ..............................
001 ..............................
002 ..............................
003 ..............................
004 ......X.......................
005 ..............................
006 ..............................
007 ........................O...
008 ..............................
009 ..............................
010 ..............................
011 ..............................
012 ..............................
013 ..............................
Piece 3 6:
.****.
**....
*.....
<got (O) : [7 24] (7,24)
Plateau 14 30:
01234567890123456789012345678 9
000 ..............................
001 ..............................
002 ..............................
003 ..............................
004 ......X.......................
005 ..............................
006 ..............................
007 ........................oooo.
008 ......................oo....
009 ......................o.....
010 ..............................
011 ..............................
012 ..............................
013 ..............................
Piece 3 8:
......*.
......**
.......*
<got (X) : [4 0] (4,0)
Plateau 14 30:
01234567890123456789012345678 9
000 ..............................
001 ..............................
002 ..............................
003 ..............................
004 ......x.......................
005 ......xx......................
006 .......x......................
007 ........................OOOO.
008 ......................OO....
009 ......................O.....
010 ..............................
011 ..............................
012 ..............................
013 ..............................
[...]
== X fin : 175 [1018918090]
== O fin : 168 [1018918090]
```

V.4.4 VM

If you experience problems with the VM, please contact us on slack. Really make sure the problem is indeed coming from the VM and not from your program.

Chapter VI

Bonus part

As a bonus will be taken into account:

- A graphic visualizer.
- Any additional bonus that you will consider useful and that your peers will approve and enjoy.

Chapter VII

Submission and peer correction

Submit your work on your `GiT` repository as usual. Only the work on your repository will be graded.