



Piscine 101

Python rush

Summary: This document is the subject for the PYTHON rush of the Piscine 101 @ 42Tokyo.

Contents

I	Instructions	2
II	Foreword	3
III	Introduction	4
IV	General Instructions	5
V	Mandatory part	6
V.1	The Filler	6
V.2	The Board	7
V.3	The tokens	7
V.4	The Topic	7
V.4.1	The Player	7
V.4.2	Multi Players	8
V.4.3	How the game rolls	9
V.4.4	VM	10
VI	Bonus part	11

Chapter I

Instructions

- Only this page will serve as reference; do not trust rumors.
- Watch out! This document could potentially change up to an hour before submission.
- These exercises are carefully laid out by order of difficulty - from easiest to hardest. We will not take into account a successfully completed harder exercise if an easier one is not perfectly functional.
- Make sure you have the appropriate permissions on your files and directories.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- Your reference guide is called `Google / man / the Internet /`
- Examine the examples thoroughly. They could very well call for details that are not explicitly mentioned in the subject...
- If no other explicit information is displayed, you must assume the following versions of languages : `Python - python3.9.0`.
- Catch every exceptions.
- Each exercises must be done using `virtualenv`.

```
?>python3 -m venv myenv  
?>source myenv/bin/activate
```

- You should use `guacamole.42tokyo.jp` to validate exercises.

Chapter II

Foreword

<https://bit.ly/3q0lX5C>

Chapter III

Introduction

Filler is an algorithmic game which consists in filling a grid of a known size in advance with pieces of random size and shapes, without the pieces being stacked more than one square above each other and without them exceeding the grid.

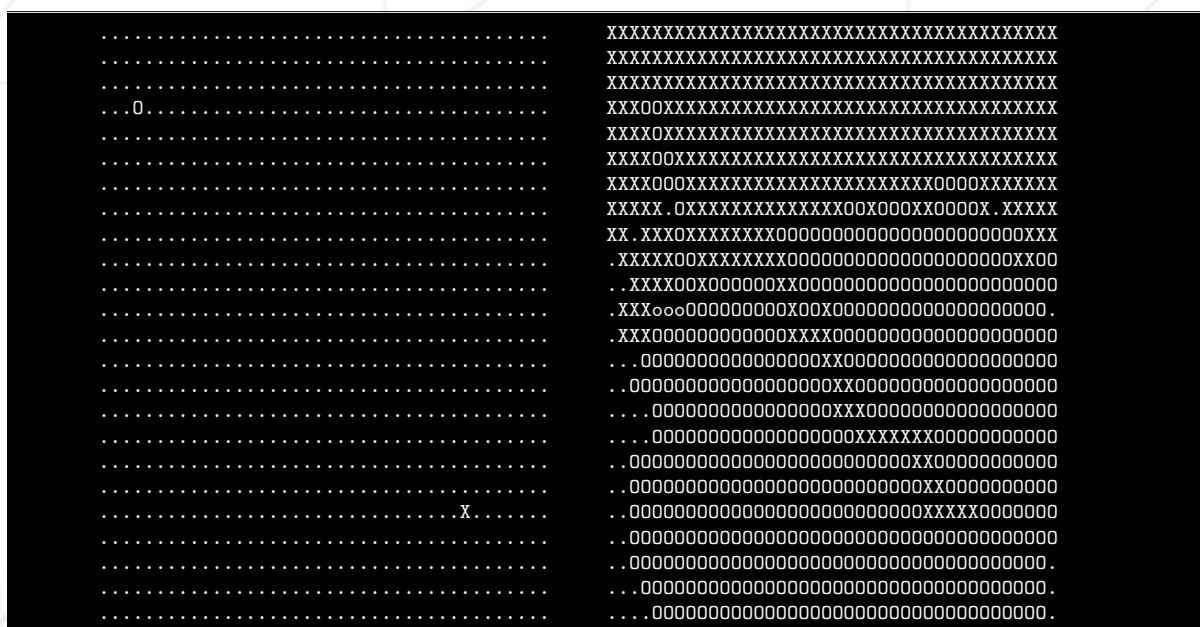
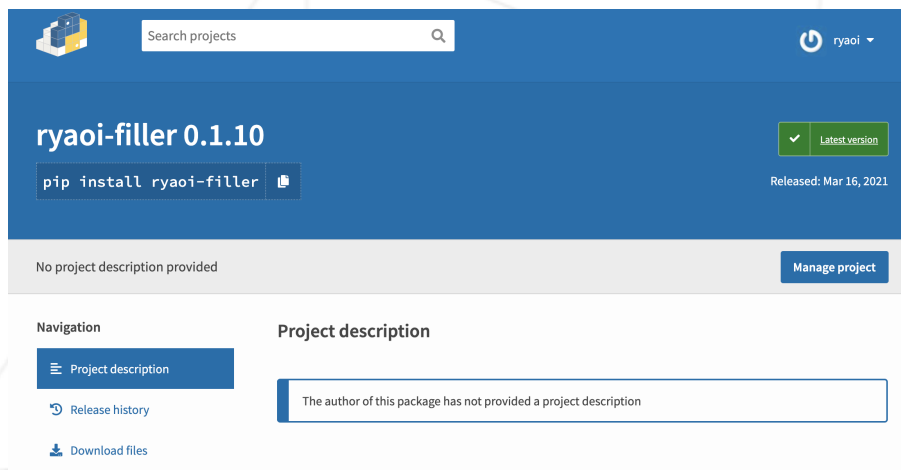


Figure III.1: On the left: the initial state of the grid, at the right: the result after the fight between two players.

Chapter IV

General Instructions

- This project will only be corrected by humans. You are therefore free to organize and name your files as you wish, while respecting the constraints listed here.
- You must create a program named `<choose_a_name>_filler`. You are free to replace this part with whatever comes up to your mind if the letter contains only alphanumeric value and the length of the string is less than 20 characters: `<choose_a_name>`.
- You must create a python package and deploy it to pypi.



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

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

- You must submit all the necessary files to the git repository specified on the project page.
- VM and players are inside `resources.tar`.



poetry

Chapter V

Mandatory part

V.1 The Filler

The goal is to collect as many points as possible by placing the highest number of pieces on the the game board.

- In this game, two players fight each other. They play one after the other.
- Each player got 3 seconds to play.
- The board is defined by X columns and N lines, it will then become $X*N$ cells.
- At the beginning of each turn, you will receive your game piece.
- A game piece is defined by X columns and N lines, so it will be $X*N$ cells. Inside each game piece, will be included a shape of one or many cells.
- To be able to place a piece on the board, it is mandatory that one, and only one cell of the shape (in your piece) covers the cell of a shape placed previously (your territory).
- The shape must not exceed the dimensions of the board
- When the game starts, the board already contains one shape.
- The game stops at the first error: either when a game piece cannot be placed anymore or it has been wrongly placed.

V.2 The Board

A board is a two-dimensional grid with an arbitrary number of rows and columns. To launch the game an initial board must be passed as an argument to the VM. This initial board must have a starting form for each player.

Here is an example of an initial board of 14 by 30 :

```
Plateau 14 30:
012345678901234567890123456789
000 .....
001 .....
002 ..X.....
003 .....
004 .....
005 .....
006 .....
007 .....
008 .....
009 .....
010 .....
011 .....0..
012 .....
013 .....
```

V.3 The tokens

The tokens are managed randomly by the VM. You can't predict their size or shape until the VM transmits them to your program. Here are some arbitrary examples of possible tokens to give you an idea:

```
Piece 4 7 Piece 4 5: Piece 3 6:
...*... .**.. .****.
...*... .***. **....
...*... .**.. *......
...***. ....
```

V.4 The Topic

V.4.1 The Player

- The executable that will enable you to play the filler is attached to this subject.
- For this project, you will have to create a filler player. Your goal is to win:
 - It will read the board and the game pieces placed on the standard output.
 - Each turn the filler rewrites the board map and includes a new piece to be placed.
 - In order to place the game piece on the board, the player will have to write it's coordinates on the standard output.
 - The following format must be used "Y X\n".
 - You will collect points each time you place a piece.


```

Plateau 14 30:
012345678901234567890123456789
000 .....
001 .....
002 .....
003 .....
004 .....
005 .....
006 .....
007 .....
008 .....0.....
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 must 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 piece last placed on the board. At the following turns, that same piece will be represented by the uppercase letters ("X" or "O"), as it won't be the piece last placed anymore.
 - You will collect points each time you place a piece.
- How the game works
 - At each turn, the filler will send the updated map and a new token to the player concerned.
 - The player concerned will write on the standard output his or her piece's coordinates to place it on the board.
 - The filler will send the map and a new piece to the other player.

V.4.3 How the game rolls

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

```
$> python3 -m virtualenv venv
$> source venv/bin/activate
(venv) $> pip install ryaoi-filler
Collecting ryaoi-filler
  Downloading ryaoi_filler-0.1.10-py3-none-any.whl (7.3 kB)
Installing collected packages: ryaoi-filler
Successfully installed ryaoi-filler-0.1.10
(venv) $> ./resources/filler_vm -t 3 -p1 ryaoi_filler -p2 resources/players/carli.
        filler -f resources/maps/map00
# ----- VM version 1.1 ----- #
#                                     #
# 42 / filler VM Developped by: Hcao - Abanlin #
#                                     #
# ----- #
launched ryaoi_filler
$$$ exec p1 : [ryaoi_filler]
launched players/carli.filler
$$$ exec p2 : [players/carli.filler]
Plateau 15 17:
  01234567890123456
000 .....
001 .....
002 .....
003 .....
004 .....
005 .....
006 .....
007 .....
008 ..0.....
009 .....
010 .....
011 .....
012 .....X..
013 .....
014 .....
Piece 2 3:
.**
.**
<got (0): [8, 1]
Plateau 15 17:
  01234567890123456
000 .....
001 .....
002 .....
003 .....
004 .....
005 .....
006 .....
007 .....
008 ..oo.....
009 ..oo.....
010 .....
011 .....
012 .....X..
013 .....
014 .....
Piece 1 3:
.**
<got (X): [12, 13]
Plateau 15 17:
  01234567890123456
000 .....
001 .....
002 .....
003 .....
004 .....
005 .....
006 .....
007 .....
```

```
008 ..00.....
009 ..00.....
010 .....
011 .....
012 .....xx.
013 .....
014 .....
Piece 2 3:
..*
***
<got (0): [9, 0]
[...]
== 0 fin: 50
== X fin: 123
```

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 bonus points, we will take into account:

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