Challenge 2 - F1 - Bird's-eye Circuit

« Prev Next »

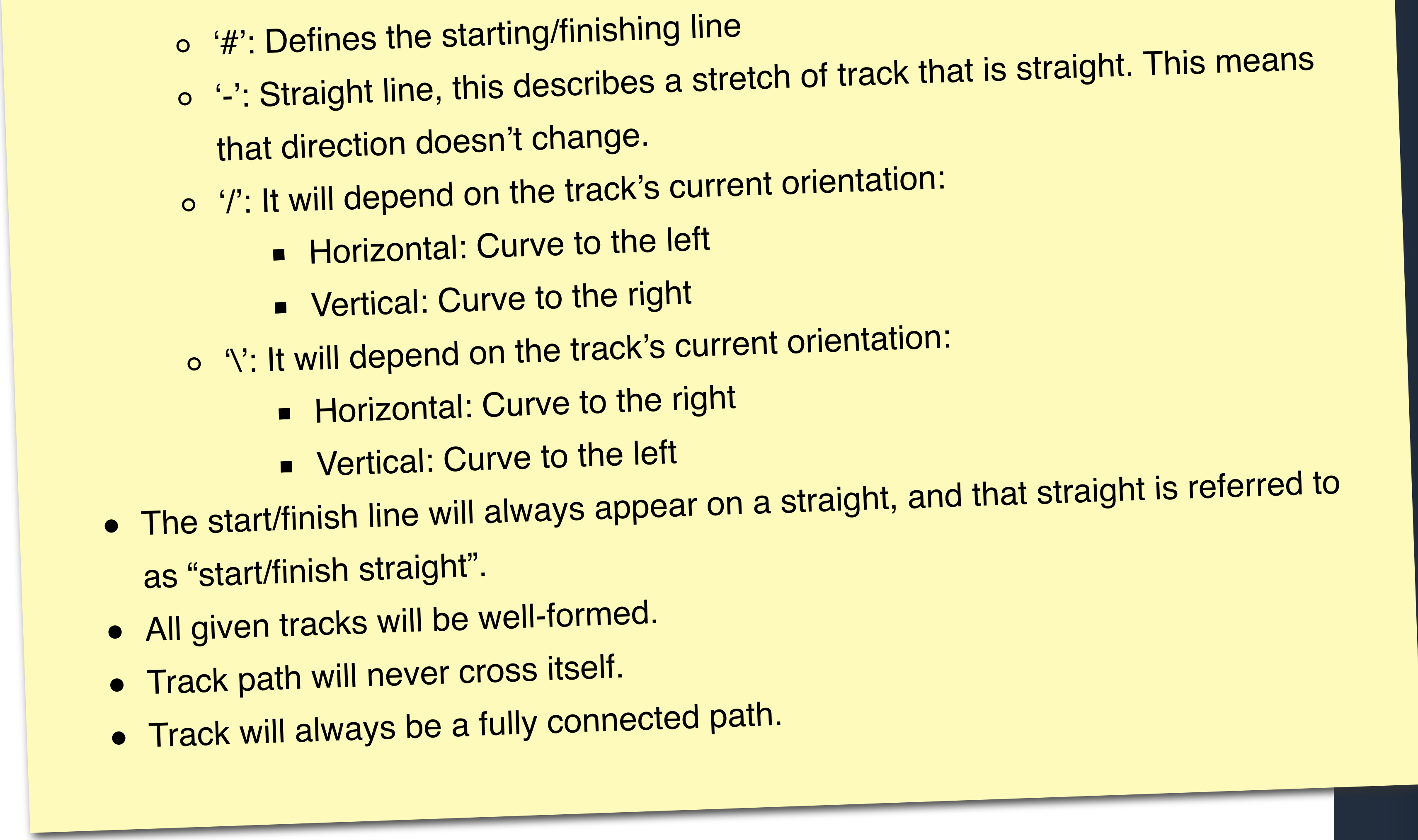
tuenti móvil

F1 Racing Team

Tuenti Móvil F1 Racing Team is an ambitious project that has already been born. You are an engineer in our brand new F1 team and as a welcome (after a big round of applause) we have a task for you.
We have just received next season's tracks in plain text format (yes, FIAL is really slapdash with these kinds of things). We want to beautify the graphic representations in order to have a visual approximation of what each track will look like in real life from an overhead view. This is your task and we are certain that you will do your best!

Oh, I almost forgot! This morning we found a post-it note on our desk written by an engineer from another team containing some basic information that is helpful for understanding the data received:

- Each track is represented as a plain text line.
- There are only four valid characters:



  - ‘#’: Defines the starting/finishing line
  - ‘-’: Straight line, this describes a stretch of track that is straight. This means that direction doesn’t change.
  - ‘/’: It will depend on the track’s current orientation:
    - Horizontal: Curve to the left
    - Vertical: Curve to the right
  - ‘\’: It will depend on the track’s current orientation:
    - Horizontal: Curve to the right
    - Vertical: Curve to the left
- The start/finish line will always appear on a straight, and that straight is referred to as “start/finish straight”.
- All given tracks will be well-formed.
- Track path will never cross itself.
- Track will always be a fully connected path.

We have only a few **requirements** for the representation (output):

- We want an overhead of the circuit in which the start/finish straight must be represented horizontally and from left to right.
- All the lines of the resulting graphic representation must have the same length and empty spaces may be added if needed to match the longest line length.
- Characters used for the graphic representation:
  - Dash ‘-’: Horizontal stretch of track.
  - Pipe ‘I’: Vertical stretch of track.
  - Slash ‘/’: Curve.
  - Back-slash ‘\’: Curve.
  - Space ‘ ‘: Empty space.
  - Line feed (as EOL) ‘\n’: This is used to change to start next line of the drawing.

In order to clarify what we want and how we want it, below are some examples of last year’s tracks:

Input

```
#----\-----/-----\-----/
```

Expected output

```
/#----\
|     |
|     |
|     |
|     |
|     |
\-----/
```

Input

```
------\-/-/-\-----#-------\--/----------------\--\----\---
/---
```

Expected output

```
/---------\
|         |
|       /-/
|       |
\----\  \-----#-------\
     |                |
     |                |
     \----------------/
```

Submit & test your code

To test and submit code we provide a set of tools to help you. Download contest tools if you haven't already done that. You will then be able to test your solution to this challenge with the challenge tokens.

```
Challenge tokens: CHALLENGE_2, CHALLENGE_SUBMIT_2
```

To test your program

```
./test_challenge CHALLENGE_2 path/program
```

A nice output will tell you if your program got the right solution or not. You can try as many times as you need.

To test your program against the input provided in the submit phase

```
./test_challenge CHALLENGE_SUBMIT_2 path/program
```

During the submit phase, in some problems, we might give your program harder inputs. As with the test token, a nice output will tell you if your program got the right solution or not. You can try as many times as you need.

In the actual contest you first need to solve the test phase before submitting the code, you must provide the source code used to solve the challenge and you can only submit once (once your solution is submitted you won't be able to amend it to fix issues or make it faster).

If you have any doubts, please check the info section.

« Prev Next »