

August, 30 2002

VERY IMPORTANT NOTE : please, note that this article is my personal raw translation of the one i've just wrote in french. So, keep cool ;-) if you find any misspelling.

The results will not be changed after a good translation !

## P4D Challenge #2 : . . . .

[snip] ;-)

### Previous Challenge: OrganizeRectangles

The challenge published in *Planet 4D* N°1 was the first « international » contest. It was the same statement for both editions, *Planète 4D* and *Planet 4D*. The count of developers who subscribed to the mini-mailing list has grown in a fantastic manner ! See the sidebar for infos about that mailing list.

Challengers had to organize rectangles of different sizes (names *small rectangles*) in a larger one (named *the area*). They must arrange as many *small rectangles* as possible in *the area*. It seems that this challenge was very difficult: only 4 subscribers have sent some code. Probably, the statement was not impossible, but too much time-consuming.

Readers of the french edition will recognize the names of the 4 challengers : they were usual and strong challengers of the previous season (3 of them was the 3 winners of the previous season). Probably, readers of the english edition feel a bit intimidated !

So, let's go back to our work. Challengers had to write a method using the following prototype :

```
OrganizeRectangles(->rectsToOrganize;areaLeft; areaTop; areaRight; areaBottom; -  
>rectsInArea;->rectsNotInArea)
```

Parameters are described in the statement of the challenge (see *Planet 4D* N°1). here is an abstract of their use.

` Pointer to a BLOB holding informations about the *small rectangles*

**C\_POINTER(\$1)**

` Coordinates of *the area*.

**C\_INTEGER(\$2;\$3;\$4;\$5)**

` Pointer To a BLOB containing informations about the *small rectangles* in *the area*

**C\_POINTER(\$6)**

` Pointer To a BLOB containing informations about the *small rectangles* that could not fit in *the area*

**C\_POINTER(\$7)**

The information stored in the BLOBs was in this order: for every *small rectangles*, there

was the number of the rectangle, then its coordinates left/top/right/bottom. All the information was put in the BLOB using **INTEGER TO BLOB** with the Macintosh byte ordering.

They were some constraints : it was forbidden to resize the rectangles, to make them overlap, to turn them. The challenger must put as many *small rectangles* as possible into the *area*.

Unfortunately, every challenger did not comment the code, and I thank those who did! Meanwhile, algorithms was similars : 1/ sort the rectangles in a wayt that permit to eliminate the one that will no fit the area, and then 2/ put the survivors in the are. The sort was done by surface, and some chalengers took care of the width and height of the rectangles. Indeed, un samll rectangle can have a little surface, but be larger than the area (ie : an area of 100x100 pixels and a small rectangle of 200x1 pixel).

As stated, 2 tests have been done. To fit one of the original goals of this challenge, the second test was this one : the small rectangles was of the same width, and they have to be put in an area of this width, but much higher (just like a day-calendar).**[This sentence has to be rewritten in ggod english, for sure !]** But it was too simple : all challengers put the same count of rectangles ! Note that testing has been done with 4D 673, compiled structure, Windows NT, CPU Athlon 700 MHz.

Here are the results. Time is in milliseconds. The database is on the ftp of *Planet 4D*. I added an interfacve that make bad rectangles flash, to quicky see what's wrong.

#### Test 1 (45 rectangles)

	Error	Rect.	Time	Points
Hugues Gombert	Overlap	29	9,2	0
Jacques Fadeuilhe	-	32	11,4	32
Lackder Si-Youcef	-	30	11980	30
Silvain Traynard	-	29	980,4	29

#### Test 2 (35 rectangles)

	Error	Rect.	Time	Points
Hugues Gombert	-	17	3,9	20
Jacques Fadeuilhe	-	17	5	19
Lackder Si-Youcef	-	17	44,6	18
Silvain Traynard	-	17	120,8	17

#### Totaux

<b>1°</b>	Jacques Fadeuilhe	51 rectangles
<b>2</b>	Lackder Si-Youcef	48
<b>3</b>	Silvain Traynard	46
<b>4</b>	Hugues Gombert	20

So, congratulations to Jacques, who wins this challenge ! The code of Lackder is very slow in test #1. I guess it is because this code wants abolutly put more rectangle than Silvain ;-).

After this first challenge of the new season, Jacques has 25 points, Lackder 15, Silvain 10 and Hugues 6. Please, note that if a 5<sup>th</sup> challenger had contest, even with an empty code, he should get the 3 points that go to the 5<sup>th</sup> of a challenge !

Here is Jacques's winning code.

```
|| [To be copy/psate from n english version of 4D]
```

<Sidebar 1>

### Some rules

- 1/ Nothing to win. Honor is the key, being *the best* at the end of the season.
- 2/ By sending a response, the challenger is OK that Editions des Sources publish his code, for free. He is OK that his code can be downloaded from the *Planet 4D/Planète 4D* ftp site.
- 3/ To win a challenge, nothing more simple : send an answer that solves it, and that solves it faster than the other challengers. The P4D Challenge is – with triggers - the only place where tracking milliseconds is mandatory.
- 4/ No plugins are allowed except those 3 one (if the challenger finds usefull to use them) : 4D PACK, 4D Chart, internetCommands. This is because those 3 plugin come with 4D Standard Edition.
- 5/ The code must compiled without any error. If the error is a little one (missing parenthesis, ...°, I'll fix it. Else, I'll send the error file to the challenger.
- 6/ There is a timeout. If a code is still running after 5 minutes, we'll consider that it is in an infinite loop. This goes as long as the statement does not say that you have more time.

</ Sidebar 1>

< Sidebar 2>

### Points

- 1/ Solve a challenge :
  - 25 points for the winner
  - 15 for the second
  - 10 for the third
  - 6 for the fourth
  - 3 pfor the fifth

2/ Propose a challenge that is submitted to challengers: 2 points. Every idea is welcome, but remember that a challenge must avoid taking « nice forms » into account (this is too subjective), and that a challenge should not take 10 days work to be solved.

</ Sidebar 2>

< Sidebar 3>

### **Mailing list**

To avoid late responses from challengers due to postal problems of the magazine, we decided to create a kind of « mini-mailing list ». If you are interested in challenging, just write to [challenge@planet-4d.com](mailto:challenge@planet-4d.com) or [challenge@planete4d.com](mailto:challenge@planete4d.com) putting « subscribe » or « unsubscribe » as the subject of the mail. Once on the list, you'll receive the announcements of the challenge, plus questions/answers when things in a challenge need clarification. Don't be afraid: there will be only something like 2-3 messages per challenge!

</Sidebar 3>