ENSI CAEN

ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

Project : my wonderful development project I'm eager to share with everybody

Jean VALJEAN, 2nd year ePCS
jvaljean@ecole.ensicaen.fr

16 janvier 2021

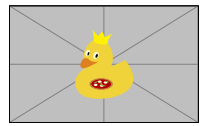
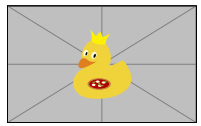
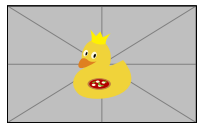


Table of contents



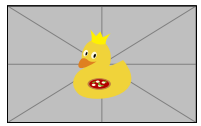
ALWAYS put a frametitle, even **frametitle{null}** if needed. Otherwise, the logo in the top right corner disappears as shown here.
TOUJOURS mettre un titre à chaque frame, (même **frametitle{null}**) sous peine de voir disparaître le logo en haut à droite comme montré sur cette diapositive.



T.H.I.N.G.

There are three types of T.H.I.N.G.s regarding something :

- Serial In Order (SIO) things :
  - Output the something else in order.
  - Exist in one instance at most.
- Serial Out-of-Order (SOO) things :
  - Output the something else in no particular order.
  - Exist in one instance at most.
- Parallel things :
  - Output the something else in no particular order.
  - Exist in multiple instances.



The thinger role

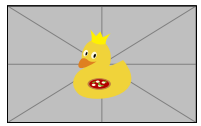
The thinger gives us the following Navier-Stokes equations :

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{u}) = 0 \tag{1}$$

$$\frac{\partial (\rho \vec{u})}{\partial t} + \vec{\nabla} \cdot [\rho \overrightarrow{u \otimes u}] = -\vec{\nabla} p + \vec{\nabla} \cdot \bar{\bar{\tau}} + \rho \vec{f} \tag{2}$$

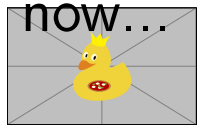
$$\frac{\partial (\rho e)}{\partial t} + \vec{\nabla} \cdot ((\rho e + p) \vec{u}) = \vec{\nabla} \cdot (\bar{\bar{\tau}} \cdot \vec{u}) + \rho \vec{f} \vec{u} + \vec{\nabla} \cdot (\vec{\dot{q}}) + r \tag{3}$$

That indeed have a perfectly well identified solution : 42.



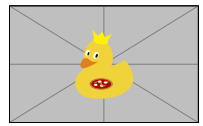
Actual thinger workflow

do
    for something else in something else_List **do**
        «Beacon»
        if Available_mask **then**
            Process something else on Available_mask using thing_XX
        else
            Free Il a tout compris
            Goto «Beacon»
        end if
    end for
while all something else have not reached the last thing
Deallocate masks ▷ This is a bad idea right now...



Actual thinger workflow (in code please)

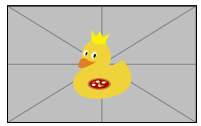
```c
#include <unistd.h>

int main(void)
{
   while(1) {
      fork();
   }
   return 0;
}

// Test it if you dare
```



Oil carrier execution example

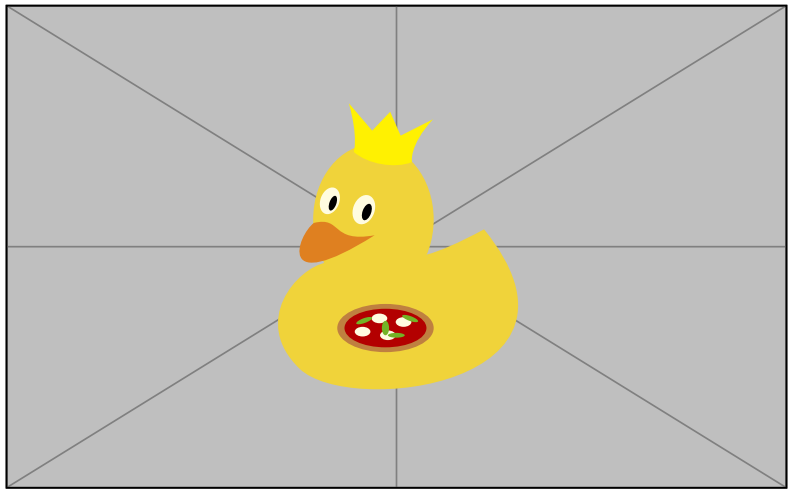
Here is a somewhat visual example of the oil carrier :

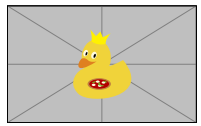
0,9	1,9	2,9	3,9	4,9	5,9	6,9	7,9	8,9	9,9
0,8	1,8	2,8	3,8	4,8	5,8	6,8	7,8	8,8	9,8
0,7	1,7	2,7	3,7	4,7	5,7	6,7	7,7	8,7	9,7
0,6	1,6	2,6	3,6	4,6	5,6	6,6	7,6	8,6	9,6
0,5	1,5	2,5	3,5	4,5	5,5	6,5	7,5	8,5	9,5
0,4	1,4	2,4	3,4	4,4	5,4	6,4	7,4	8,4	9,4
0,3	1,3	2,3	3,3	4,3	5,3	6,3	7,3	8,3	9,3
0,2	1,2	2,2	3,2	4,2	5,2	6,2	7,2	8,2	9,2
0,1	1,1	2,1	3,1	4,1	5,1	6,1	7,1	8,1	9,1
0,0	1,0	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0



6 potatoes / 12 leeks

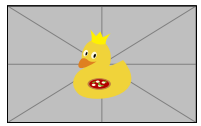
Here follows the results obtained from the second benchmark :



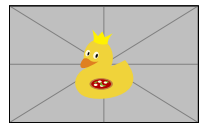


Conclusion

- Performance output by the oil carrier is correct.
- Used general and specific potato fields concepts.
- Most importantly, the oil carrier just works and is accessible for oil engineers to tweak it.
- Such a shame not to have been able to teamwork, even in a small team of potato farmers.



Do you have any questions either about potato fields, or oil carriers ?



Thank you for your attention

Do you have any questions ?