Environnements de développement de logiciel

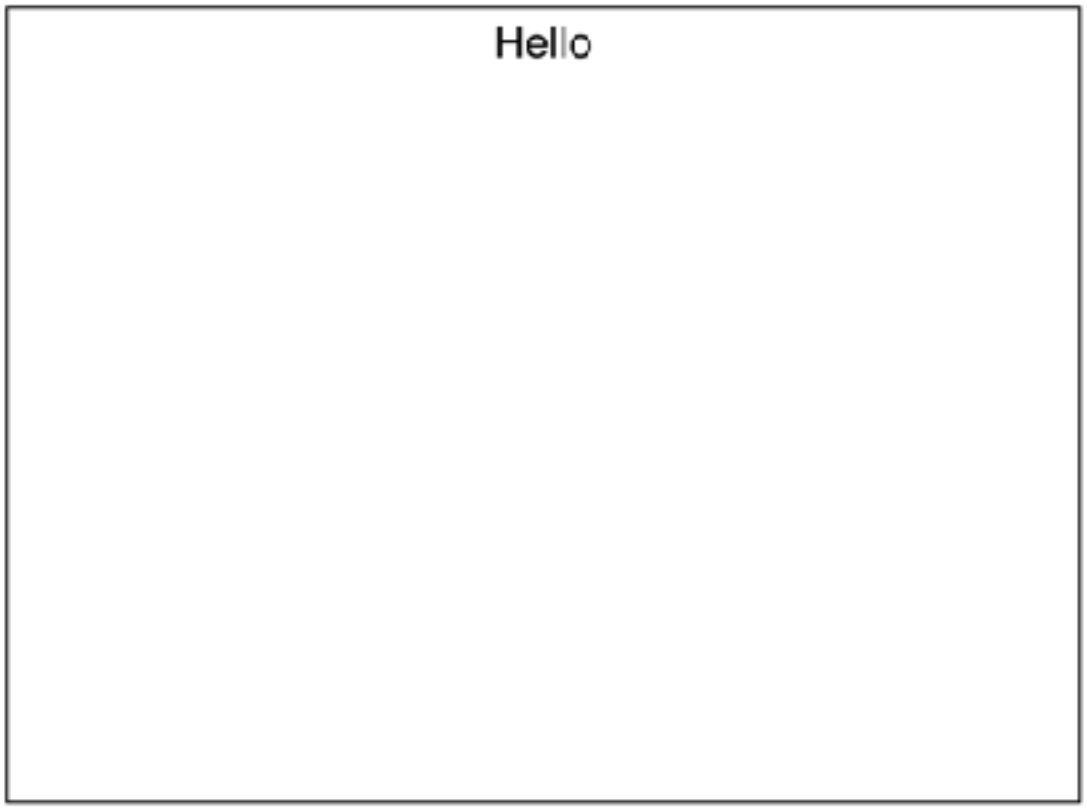
J.F. Santucci
UMR CNRS 6134 SPE
santucci@univ-corse.fr

SlideShow

- Slideshow est une librairie dédiée à la création de présentation.
- Le concept se rapproche de la classe Beamer pour LaTeX.
- On crée un programme qui va décrire le document de présentation et produire en sortie un fichier PDF par exemple.
- Comme pour les aspects GUI, le cours présente les principes mais évidemment la documentation vous offre énormément d'autres possibilités.

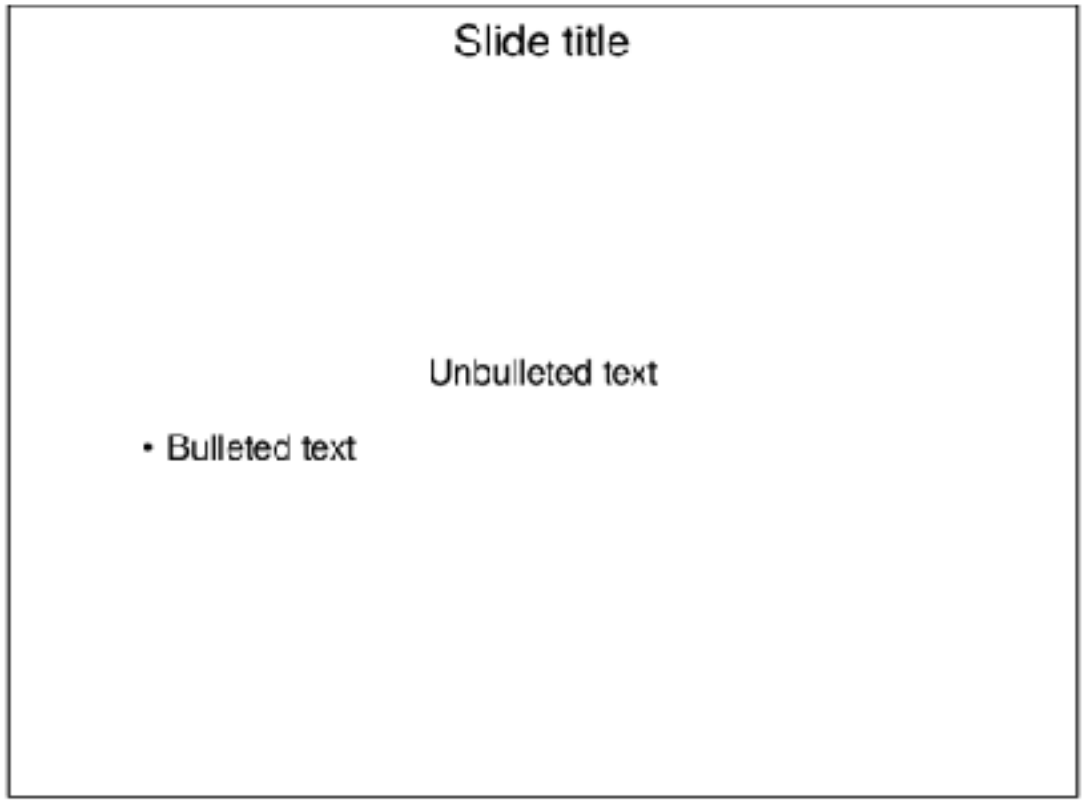
Slideshow

```
(require slideshow)
(slide #:title "Hello")
```



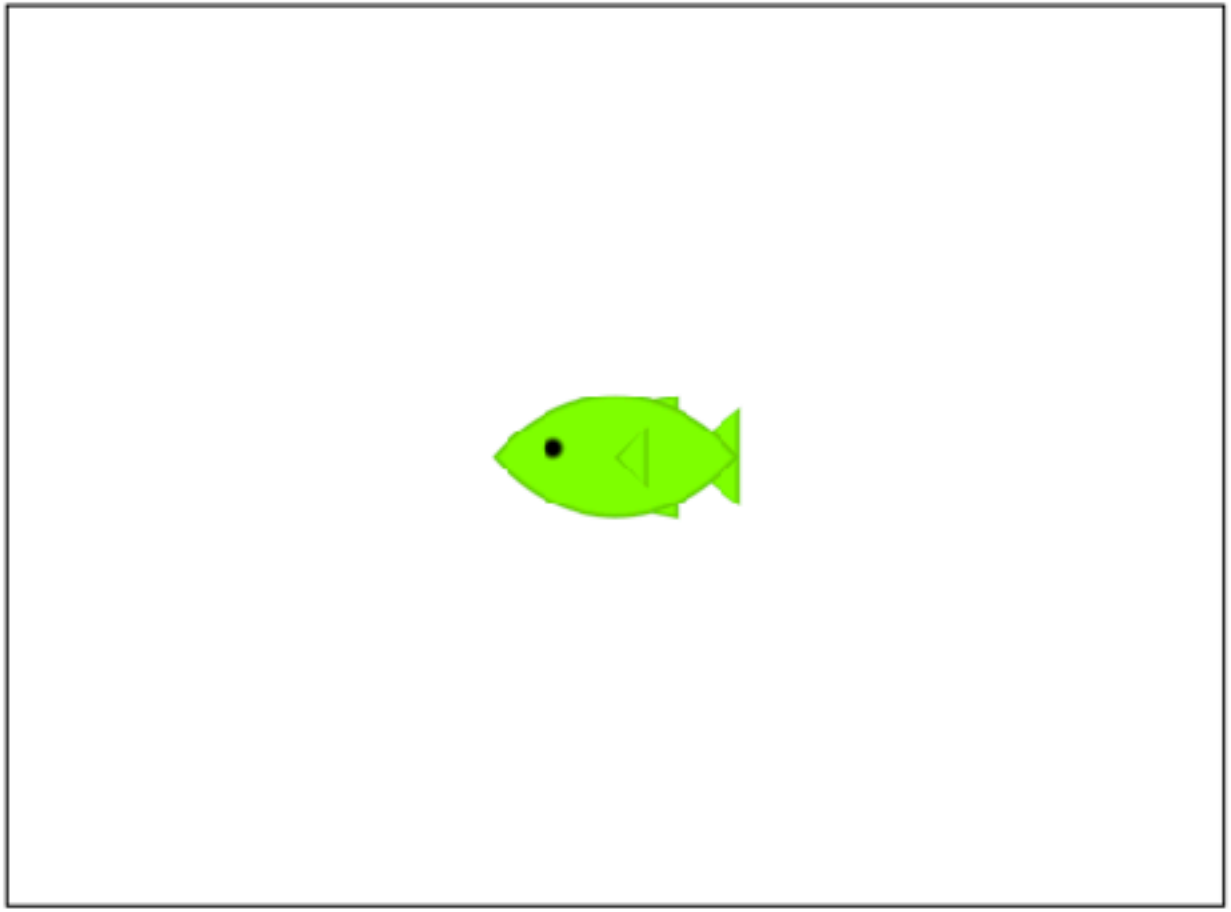
Slideshow

```
(slide #:title "Slide title"
  (t "Unbulleted text")
  (item "Bulleted text"))
```



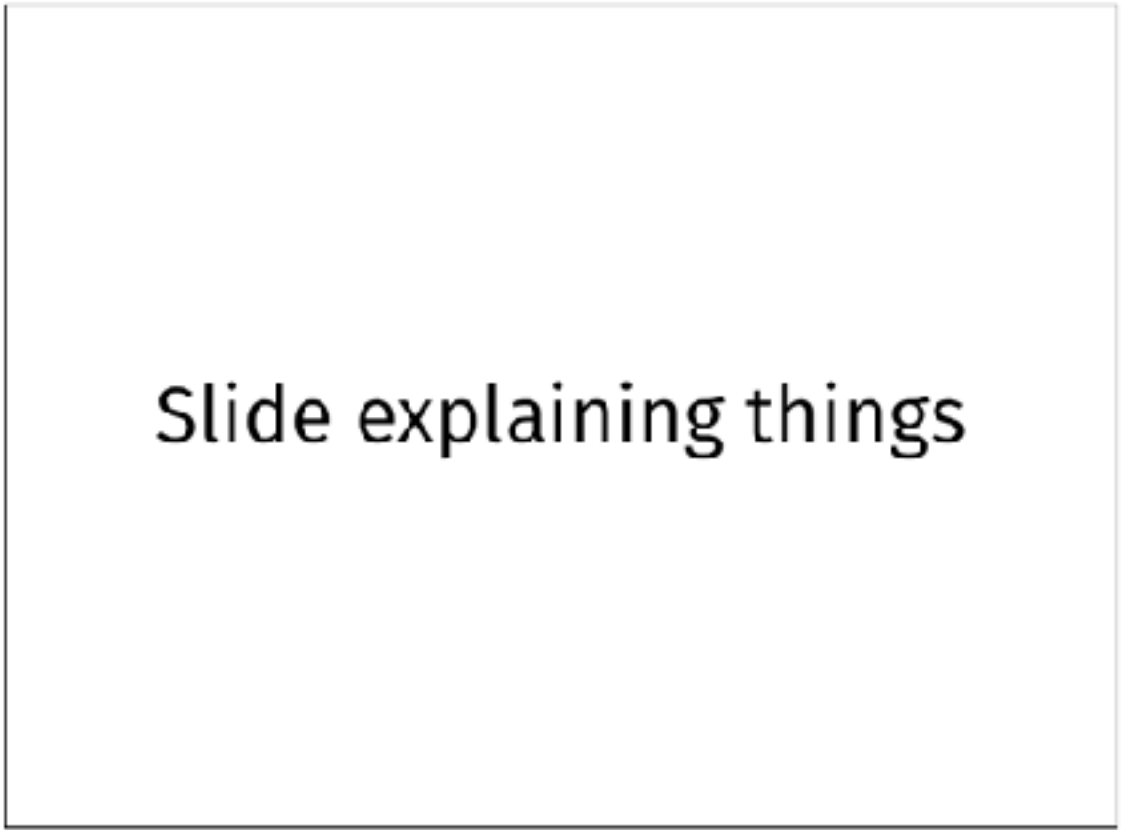
Slideshow

```
(begin (require pict)
       (slide (standard-fish 200 100 #:color "chartreuse")))
```



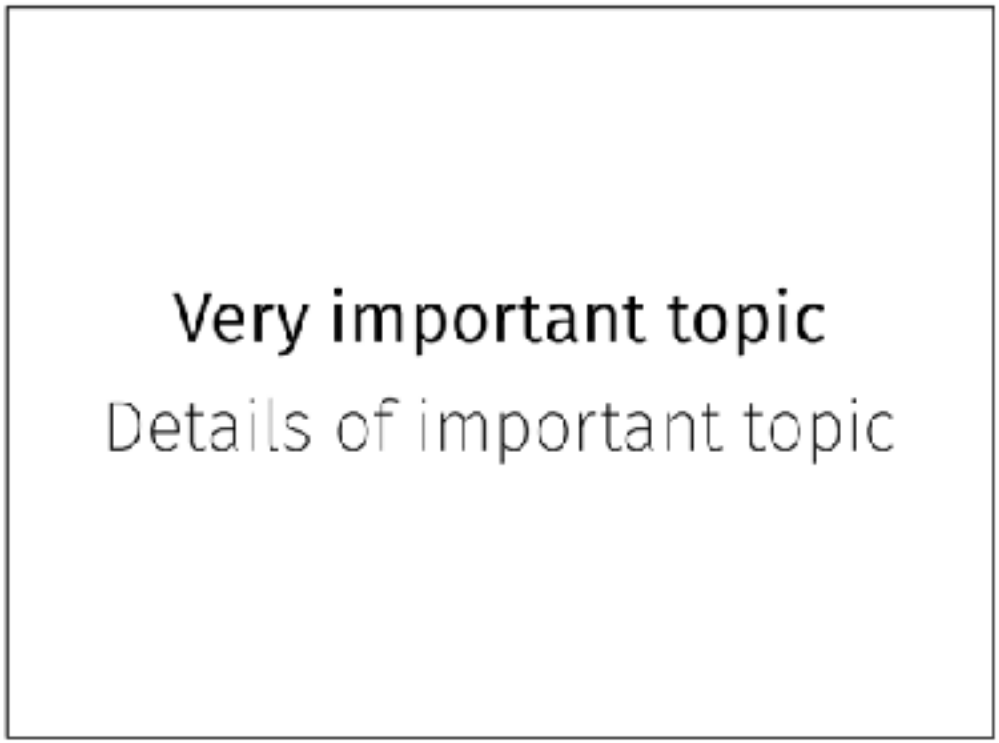
Slideshow

```
(begin (current-main-font "Fira Sans")
       (current-font-size 70)
       (slide (t "Slide explaining things")))
```



Slideshow

```
(begin (current-main-font "Fira Sans")
       (current-font-size 70)
       (slide (t "Very important topic")
              (text "Details of important topic"
                    "Fira Sans, ExtraLight"
                    (current-font-size))))
```



Slideshow

```
(begin (require slideshow-text-style)
       (with-text-style
         #:defaults (#:face "Fira Sans")
         ([heading #:size 70 #:bold? #t]
          [auth #:size 50 #:color "firebrick"])

         (slide (heading "Title of My Talk")
                (auth "Alice the Programmer"))))
```

Attention :
installer le package
slideshow-text-style

Slideshow

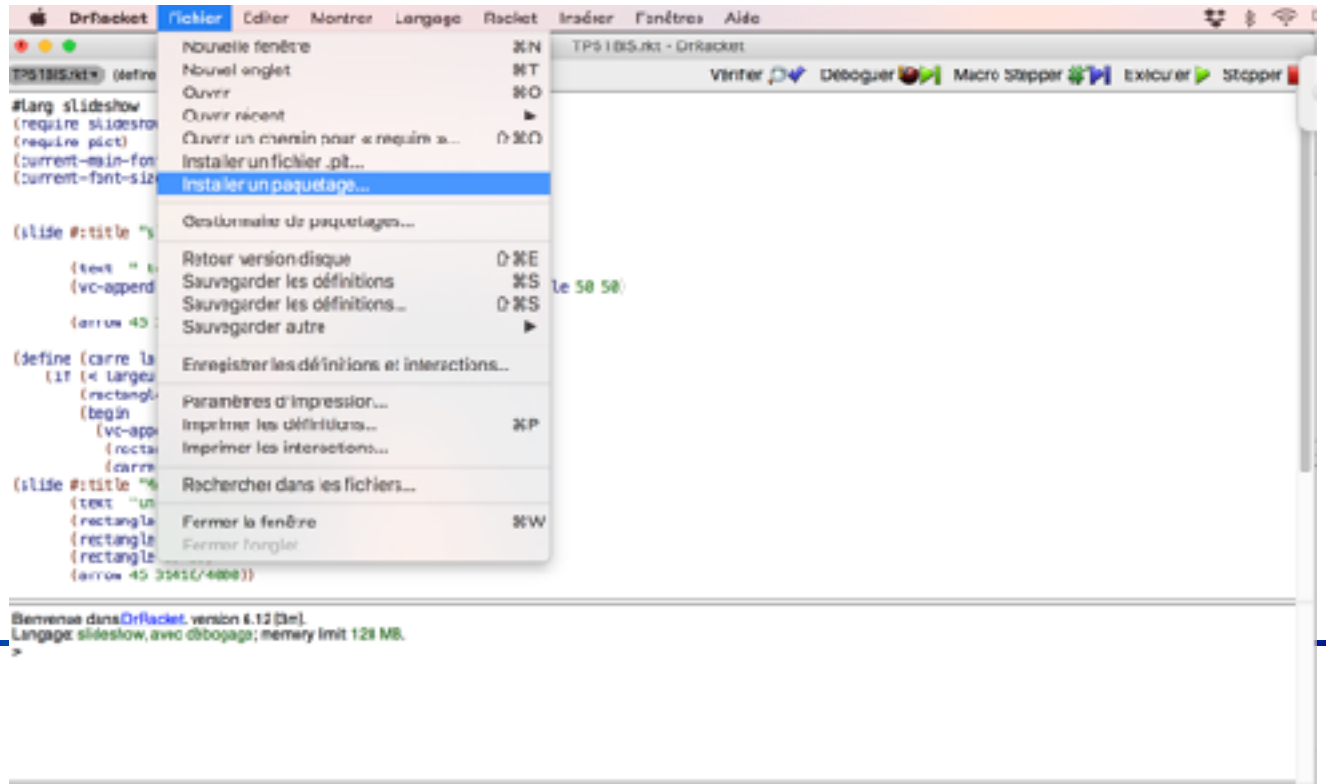
```
(begin (require ppict/2 ppict/slideshow2)
       (with-text-style
         #:defaults (#:face "Fira Sans, Condensed")
         ([h #:size 70]
          [t #:size 50 #:face "Fira Sans, Light"])

         (pslide #:go (coord 0.1 0.1 'lt)
                 (h "Research idea")
                 #:go (coord 0.2 0.5 'lc)
                 (t "A DSL for standard fishes")
                 #:go (coord 0.8 0.8)
                 (standard-fish 200 100 #:color "tomato"))))
```

Attention si la paquetage ppict n'existe pas il faut l'installer : pour cela plusieurs méthodes

Slideshow

Installation de paquetage ;

On peut utiliser les fonctionnalités de DrRacket par exemple



Slideshow

Research idea

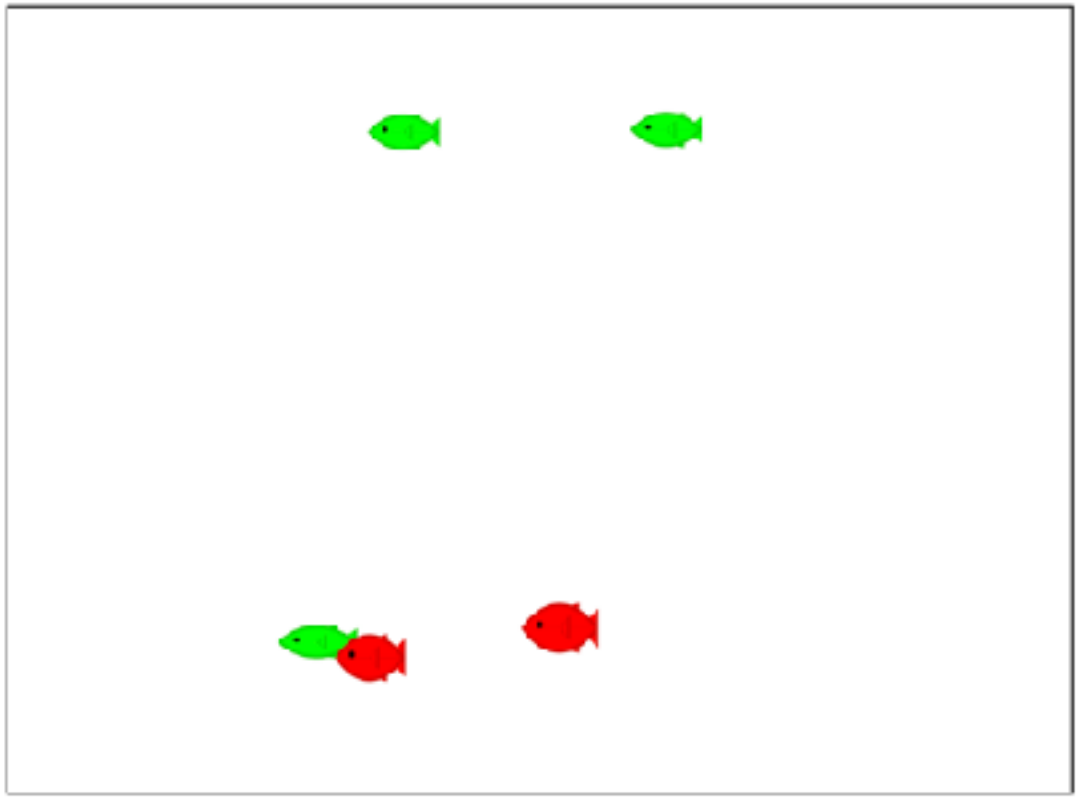
A DSL for standard fishes

Slideshow

```
(begin (require ppict/2 ppict/slideshow2)
       (require racket/random)
       (define fishes
         (for/fold ([pic (blank 500 500)])
                   ([n 5])
           (ppict-do
            pic
            #:go (coord
                  (random)
                  (random)
                  (random-ref '(lt lb lc rt rb rc cc)))
            (standard-fish (random 60 90)
                           (random 30 50)
                           #:color
                           (random-ref
                            '("red" "green" "blue"))))))
       (pslide fishes))
```

Slideshow

https://pkgs.racket-lang.org/package/ppict



Slideshow

Présentation en couleurs

https://docs.racket-lang.org/draw/color-database___.html

Slideshow

On peut aussi utiliser des formes (rectangles, cercles, etc..).
La librairie pict offre des primitives et des outils de combinaison de formes.
La primitive `filled-rectangle` permet de dessiner un rectangle utilizable comme fonds et que l'on peut combiner avec `cc-superimpose` (superpose des objets les uns sur les autres) superimposes a pict on top of another). La variable `client-w` définit la largeur de l'ecran.

Slideshow

```
(begin (require pict racket/draw ppict/slideshow2)

       (pslide #:go (coord 0.5 0.5)
                (cc-superimpose
                 (filled-rectangle
                  client-w 300 #:color "firebrick")
                 (colorize
                  (text "Conclusion: 1 + 1 = 2"
                        "Fira Sans, Condensed, Heavy" 80)
                  "white"))))
```

Slideshow

On peut aussi construire ses propres primitives en utilisant dc.

Voir exemple suivant qui implementer des degrés d'affichage de couleurs (donc affichage en dégradé).
Pas besoin de voir les détails d'implementation.

```
(begin (require pict racket/draw ppict/slideshow2)

       (define (rectangle/2t
                width height
                #:border-width [border-width 1]
                #:border-color [border-color "black"]
                #:color-1 [color-1 "white"]
                #:color-2 [color-2 "black"])
         (dc (λ (dc dx dy)
               (define old-brush
                 (send dc get-brush))
               (define old-pen
                 (send dc get-pen))
               (define gradient
                 (make-object
                  linear-gradient%
                  dx dy
                  dx (+ dy height)
                  `((0 ,(make-object color% color-1))
                    (1 ,(make-object color% color-2)))))
               (send dc set-brush
                 (new brush% [gradient gradient]))
               (send dc set-pen
                 (new pen% [width border-width]
                           [color border-color]))
               (send dc draw-rectangle dx dy width height)
               (send dc set-brush old-brush)
               (send dc set-pen old-pen))
             width height))

       (pslide #:go (coord 0.5 0.5)
               (cc-superimpose
                (rectangle/2t client-w 300
                              #:color-1 "mistyrose"
                              #:color-2 "white")
                (vl-append 20
                 (text "AFFICHAGE:"
                       "Fira Sans, Heavy" 60)
                 (text « DEGRADE ou DEGRADE????"
                       "Fira Sans" 50)))))
```

Slideshow

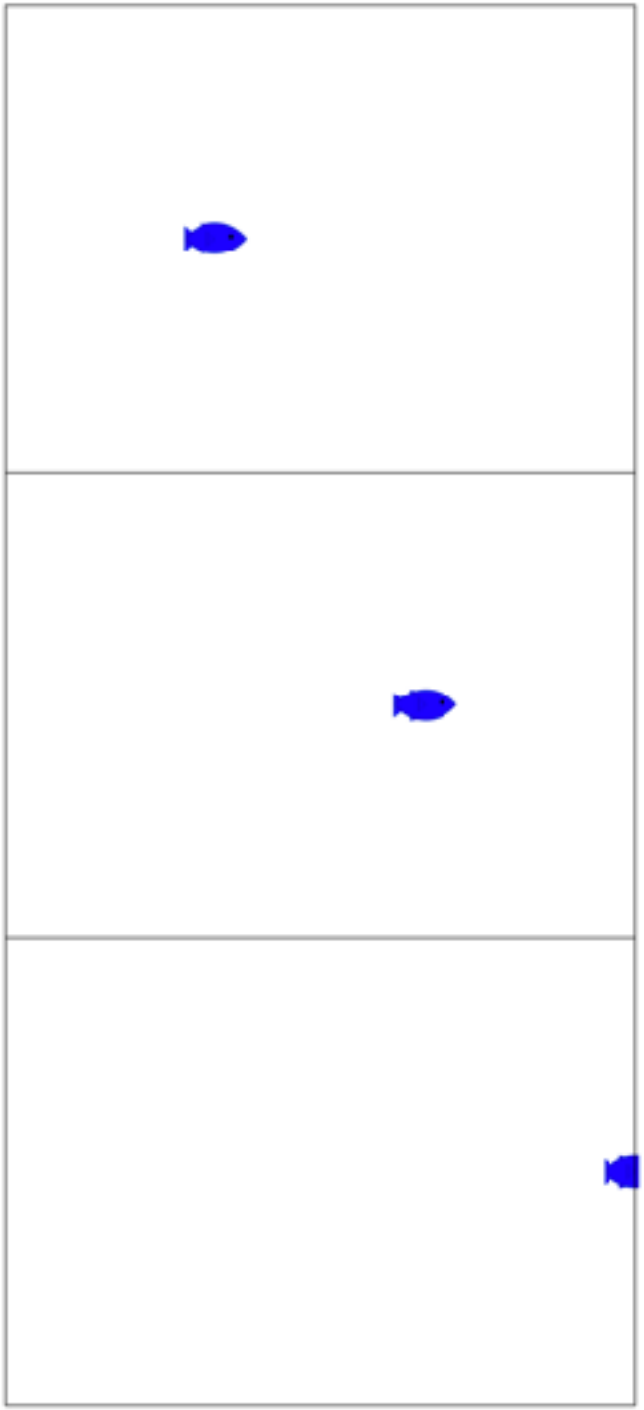
Future work:

standard-fish + standard-fish = ???

Slideshow

Animations des slides.
La fonction play appartenant à la librairie `slideshow/play` permet d'ajouter des animations dans les slides .

```
(begin (require ppict/2
                slideshow/play)

       (play #:steps 3
        (lambda (x-coord)
          (ppict-do (blank client-w client-h)
           #:go (coord x-coord 0.5)
           (standard-fish 100 50
            #:direction 'right)))))
```



TP N°5