

# Environnements de développement de logiciel

J.F. Santucci

UMR CNRS 6134 SPE

[santucci@univ-corse.fr](mailto:santucci@univ-corse.fr)

---

# GUI

Une interface graphique est constitué d'un ensemble de composants : fenêtres, boutons, menus, zones à dessiner, zones de texte éditantes ou fixes, etc...

Les fonctions de Racket donnant accès aux classes de tous ces composants graphiques ne sont pas accessibles dans le langage de base racket réservé au mode texte.

Il faut utiliser l'API racket/gui.

Donc pour l'utiliser : (require racket/gui)

---

---

# GUI

Cette API est très complète (nous ne verrons pas tous les éléments en cours).

Trois exemples typiques :

- présentation de résultats textuels de calculs,
  - éditeurs de texte,
  - dessins dans une fenêtre graphique avec ou sans la souris.
-

# GUI

Présentation de résultats de calculs.

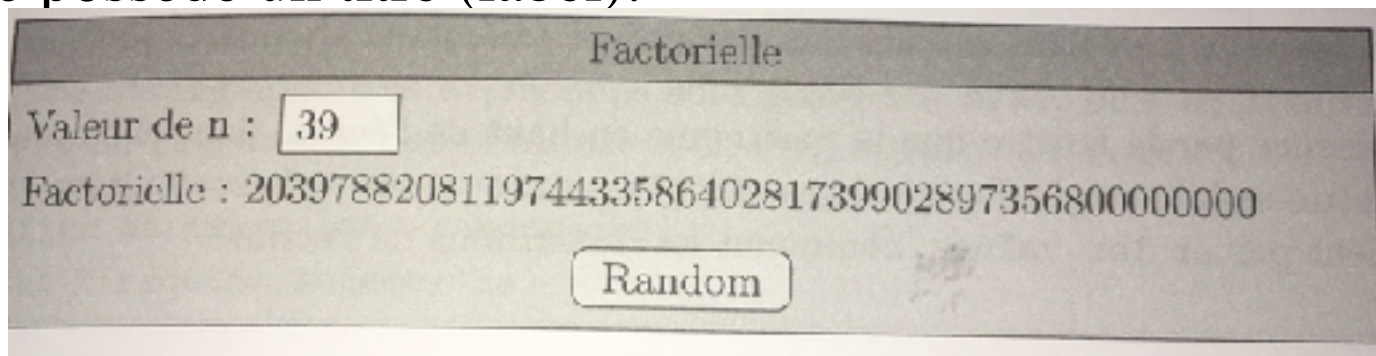
L'utilisateur entre une ou plusieurs données dans des zones de texte éditables.

Ces données sont exploitées pour calculer un résultat.

Ce résultat est présenté dans une zone de texte non éditable.

Le composant principal est une fenêtre : objet de la classe `Frame`

Elle possède un titre (label).



---

# GUI

;;; Exemple — Présenter le résultat d'un calcul

```
#lang racket/gui
```

```
(define FRAME (new frame% (label « Factorielle »)))
```

Cette fenêtre est un composant pouvant lui-même contenir d'autres composants : il s'agit d'un conteneur.

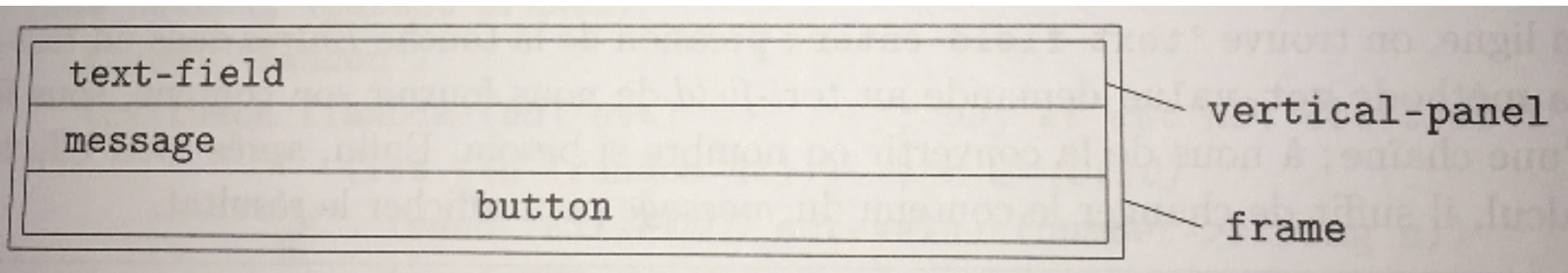
Ses composants sont disposés à la verticale ou à l'horizontale dans un ou plusieurs panneaux invisibles (panels).

---

# GUI

Un panneau étant lui-même un composant et un conteneur, on pourra choisir entre de l'horizontal ou vertical.

Ce choix est fait sur papier.



---

# GUI

La fenêtre va contenir un panneau vertical VPANEL contenant 3 objets:

- une zone de texte éditable TEXTFIELD, de la classe text-field%, dans lequel l'utilisateur va pouvoir entrer des données - zone alignée à gauche
  - un message MSG, de la classe message%. Il s'agit d'un texte sur une seule ligne que l'utilisateur ne peut pas modifier, qui nous servira à présenter le résultat. Ce message est aligné à gauche.
  - un bouton BUTTON de la classe button%. Un click de souris dans un tel bouton provoquera une action par un mécanisme de callback (à noter le bouton est centré).
-

---

# GUI

Mécanisme de callback : fonction prenant en argument l'objet courant et l'événement. L'objet est le text-field lui-meme et l'événement est un objet de la classe `control-event%` qui contient une information importante : le type de l'événement.

La méthode `get-event-type` permet de demander à un événement quel est son type (voir aide en ligne éventuellement : `type d'event` :  
'button' 'check-box' 'choice' 'list-box' 'list-box-dclick' 'list-box-column' 'text-field' 'text-field-enter' 'menu' 'slider' 'radio-box' 'tab-panel' 'menu-popdown' 'menu-popdown-none'

---



---

# GUI

Parmi les choix : 'text-field-enter : pression de la touche dans la touche Entrée dans un text-field.

La méthode get-value demande au text-field de fournir son contenu (sous forme d'une chaîne). Il faut ensuite la convertir en nombre si besoin.

Puis après avoir effectué le calcul, il suffit de changer le contenu du message pour afficher le résultat.

---

---

# GUI

Il faut noter que manipulons des objets qui sont construits en initialisant des champs nommés.

Pour une fenêtre : le titre

Pour un panneau : le conteneur parent et le mode d'alignement à l'horizontal (left, center, right) ou vertical (top, center, bottom).

Utiliser l'aide en ligne afin de voir les champs obligatoires ou optionnels lors de la construction, les valeurs par défaut et les messages qu'il peut recevoir.

---

---

# GUI

L'ordre dans lequel sont définis les composants détermine leur placement et donc très important.

On commence par placer dans la fenêtre le panneau vertical (ensemble potentiel de composants empilés à la verticale et alignés à gauche) :

```
(define VPANEL (new vertical-panel%  
(parent FRAME) (alignement '(left center))))
```

---

# GUI

Dans ce panneau vertical, le premier composant - placé au sommet - sera le text-field que l'on va construire en précisant :

- son titre

- sa largeur minimale

- sa valeur initiale

Attention : le contenu d'un text-field est toujours une chaîne de caractères (même si il doit contenir un nombre).

Pour empêcher le gestionnaire de disposition d'étirer le text-field jusqu'au bord droit de la fenêtre, on précise qu'il n'est pas redimensionnable.

---

# GUI

(define TEXT-FIELD

(new text-field%(parent VPANEL)

(label « valeur de n : »)

(min-width 140) ; largeur minimum

(init-value « 0 ») ; valeur initiale

(stretchable-width #f ; non redimensionnable

(callback (lambda (obj evt)

(when (equal? (send evt get-even-type) 'text-field-enter)

(let ((n (string->number (send TEXT-FIELD get-value))))

(send MSG set-label (format « Factorielle : » (fac n))))))

---

# GUI

Remarque : le texte d'un message ne peut dépasser 200 caractères.

Si il doit être plus long, il faut une barre de défilement horizontal dans la fenêtre.

```
(define MSG  
  (new message% (parent VPANEL)  
    (label « Factorielle. : 1 »)  
    (min-width 500)))
```

---

# GUI

Il nous faut ensuite placer le bouton de nom Random.

Une pression sur ce bouton provoquera le placement d'un entier aléatoire dans le text-field.

Attention : le panneau vertical a ses composants alignés à gauche.

Si on veut centrer le bouton, il faut le placer tout seul dans un panneau horizontal HPANEL(dernier composant du panneau vertical)

# GUI

```
(define HPANEL
```

```
  (new horizontal-panel% (parent VPANEL)
```

```
    (alignment '(center center))))
```

```
(define BUTTON
```

```
  (new button% (parent HPANEL)
```

```
    (label « Random »)
```

```
    (callback (lambda (obj evt). ; obj et evt non utilisés
```

```
      (let ((n (random 41)) ; n compris entre 0 et 40
```

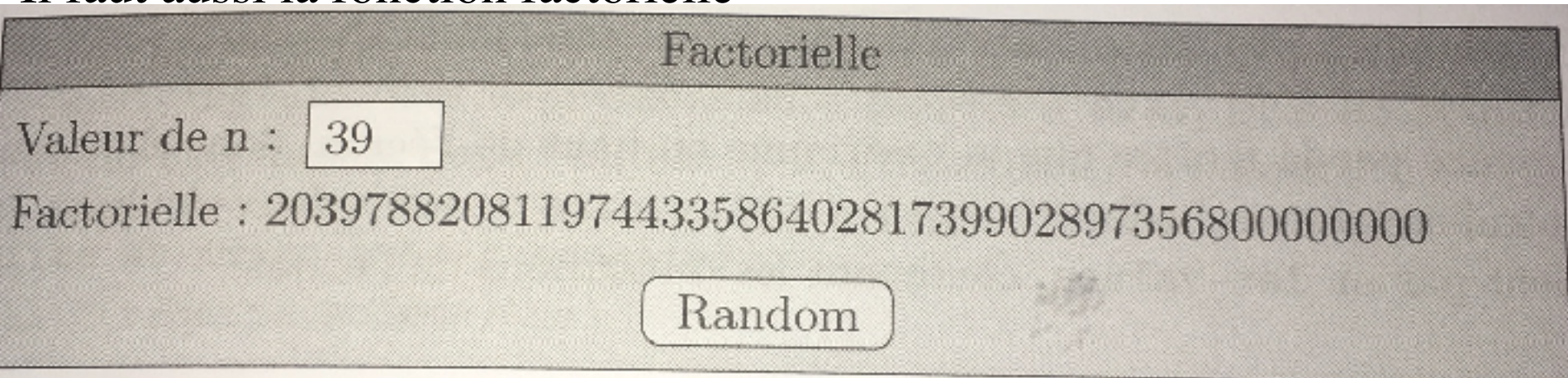
```
        (send TEXT-FIELD set-value (number->string n))
```

```
        (send MSG set-label (format « Factorielle : » (fac n))))))))
```



# GUI

Il faut aussi la fonction factorielle



A screenshot of a Java Swing window titled "Factorielle". The window has a light gray background. At the top, the title "Factorielle" is displayed in a dark font. Below the title, there is a text label "Valeur de n :" followed by a text input field containing the number "39". Below the input field, there is a text label "Factorielle :" followed by a large text area displaying the factorial of 39, which is "20397882081197443358640281739902897356800000000". At the bottom center of the window, there is a button labeled "Random".

Remarques : on doit cliquer sur le bouton (même si une pression sur la touche Entrée a le même effet : mais l'événement est capturé par le callback du text-field et le bouton ne peut pas l'entendre. On pourrait positionner le champs style du bouton qui permettrait de la mettre à l'écoute du clavier avec la valeur border. Mais alors le text-field serait lui sourd!