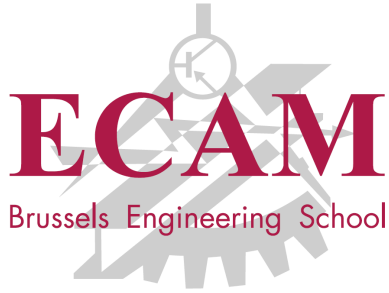


IN20 Informatique

Séance 4

Interface graphique et programmation évènementielle

Sébastien Combéfis, Quentin Lurkin

lundi 5 octobre 2015

Ce(tte) œuvre est mise à disposition selon les termes de la Licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Pas de Modification 4.0 International.

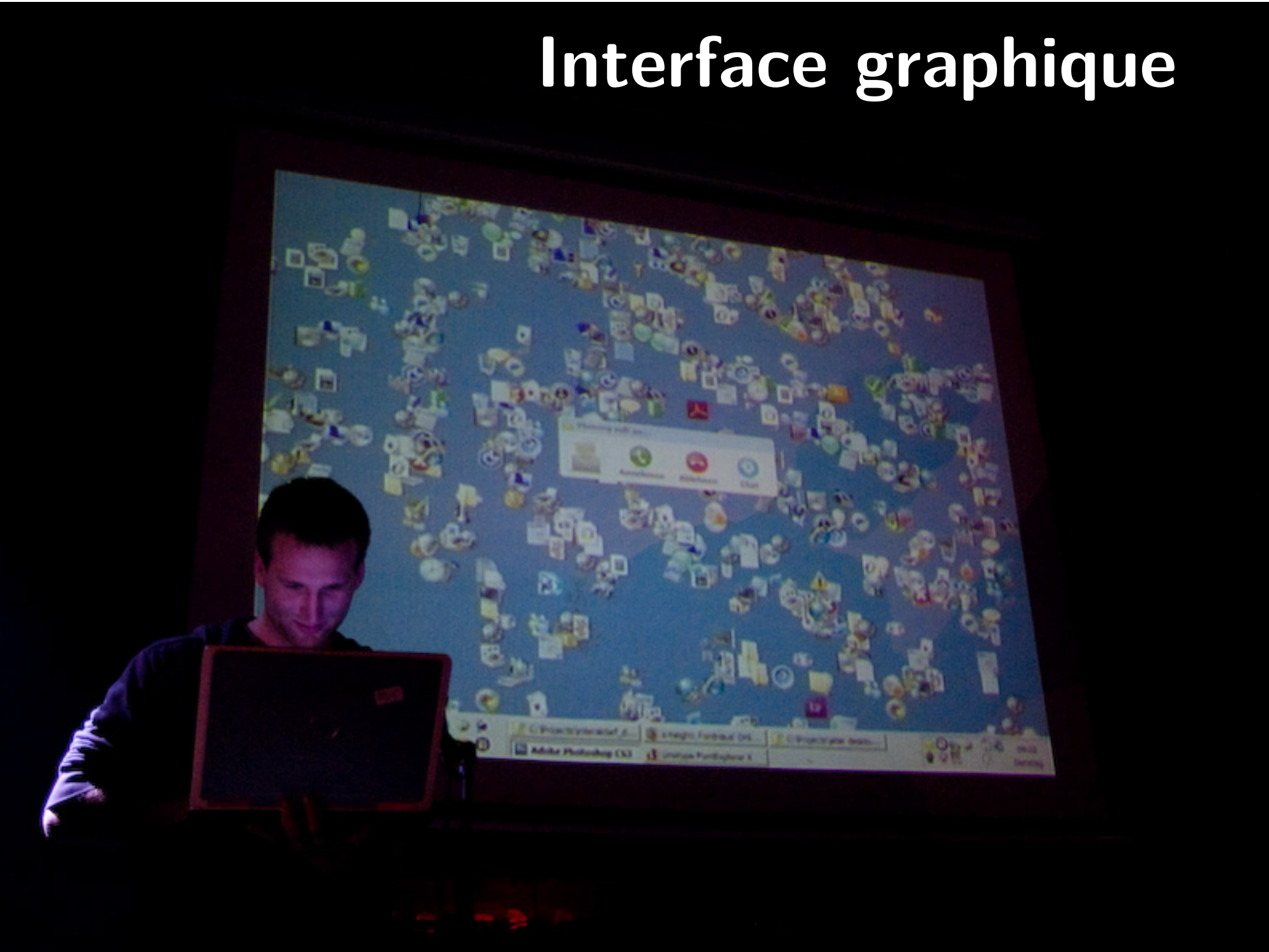
Rappels

- Classe
  - Définition d'une classe (variable d'instance et constructeur)
  - Définition d'un constructeur et instanciation d'un objet
  - Définition de méthode et appel
- Programmation orientée objet
  - Représentation d'un objet avec `__str__`
  - Surcharge d'opérateur
  - Égalité des objets (==) et des identités (`is`)
  - Accesseur et mutateur

Objectifs

- Interface graphique
  - Construction d'une interface graphique
  - Widgets graphiques
- Programmation évènementielle
  - Boucle d'évènement
  - Action et gestionnaire d'évènement
  - Canvas et animation

Interface graphique



Graphical User Interface

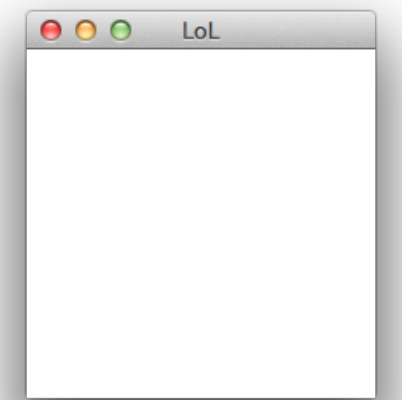
- Utilisation de la librairie graphique Tk pour les GUI

  Importation de `tkinter`

- Création d'une nouvelle fenêtre avec un objet `Tk`

  Titre changé via la méthode `title`

```
1  from tkinter import *
2
3  window = Tk()
4  window.title('LoL')
5  window.mainloop()
```



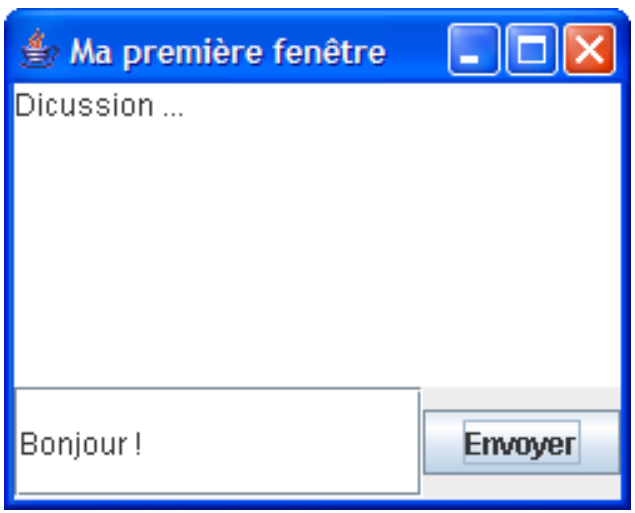
Widget

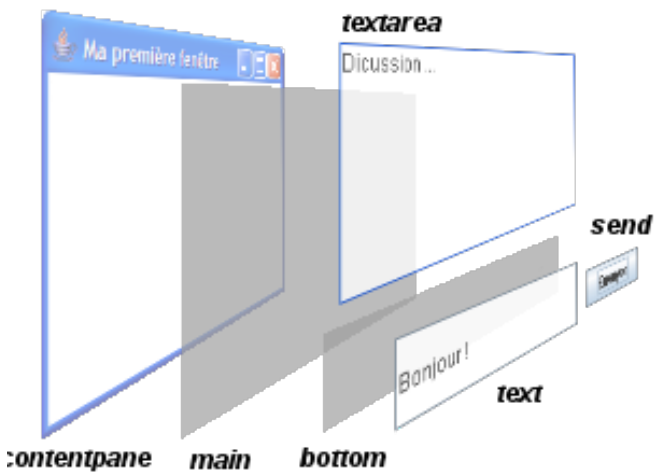
- Interface graphique construite à partir de widgets

  Composants tels que label, bouton, fenêtre, liste, case à cocher...

- Widgets placés les uns dans les autres

  Chaque composant a un composant parent



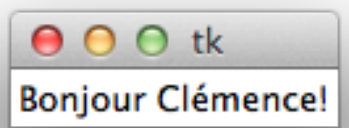


Label

- Un label est une zone de texte non modifiable

  On définit son texte avec le paramètre `text`

```
1  window = Tk()
2
3  text = Label(window, text='Bonjour Clémence!')
4  text.pack()
5
6  window.mainloop()
```

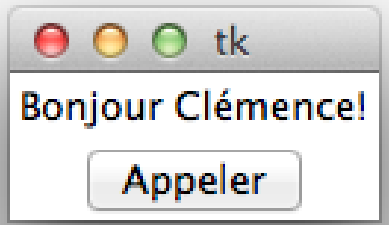


Bouton

- Un bouton peut être cliqué pour déclencher une action

  On définit son texte avec le paramètre `text`

```
1  window = Tk()
2  text = Label(window, text='Bonjour Clémence!')
3  text.pack()
4
5  btn = Button(window, text='Appeler')
6  btn.pack()
7
8  window.mainloop()
```

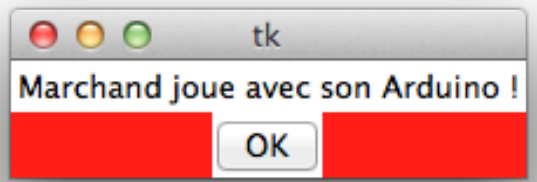


Frame

- Une frame est une zone rectangulaire

  Permet d'accueillir d'autres composants

```
1  window = Tk()
2
3  f = Frame(window, background='red')
4  f.pack()
5  text = Label(f, text='Marchand joue avec son Arduino !')
6  text.pack()
7  ok = Button(f, text='OK')
8  ok.pack()
9
10 window.mainloop()
```



Notation compacte

- Pas nécessaire de stocker une référence par widget

  Sauf si on doit y faire référence ailleurs ou appeler des méthodes

- Appel direct à `pack` après création du widget

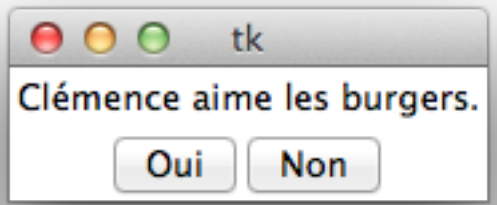
```
1  window = Tk()
2
3  f = Frame(window, background='red').pack()
4
5  Label(f, text='Marchand joue avec son Arduino !').pack()
6  Button(f, text='OK').pack()
7
8  window.mainloop()
```

Construction d'interface

- Spécification du composant parent

  Pour chaque nouveau composant créé

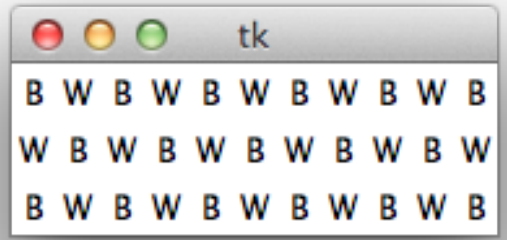
```
1  window = Tk()
2  Label(window, text='Clémence aime les burgers.').pack()
3  f = Frame(window)
4
5  Button(f, text='Oui').pack(side=LEFT)
6  Button(f, text='Non').pack(side=LEFT)
7
8  f.pack()
9  window.mainloop()
```



Gestionnaire de mise en page

- Trois manières de gérer la mise en page
  - `pack` permet du positionnement relatif
  - `place` permet de spécifier les coordonnées précises
  - `grid` permet de placer les composants en grille

```
1  window = Tk()
2  text = 'B'
3  for r in range(3):
4      for c in range(11):
5          Label(window, text=text).grid(row=r, column=c)
6          text = 'W' if text == 'B' else 'B'
7  window.mainloop()
```

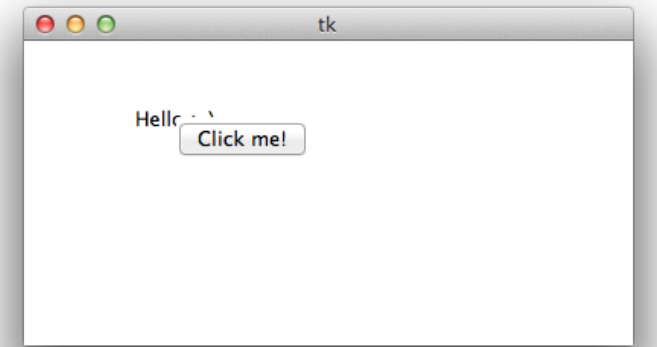


Placement précis

- Positionnement avec des coordonnées précises

  Attention, les composants peuvent se chevaucher

```
1  window = Tk()
2  window.geometry('400x200')
3
4  Label(text='Hello :-)').place(x=70, y=40)
5  Button(text='Click me!').place(x=100, y=50)
6
7  window.mainloop()
```

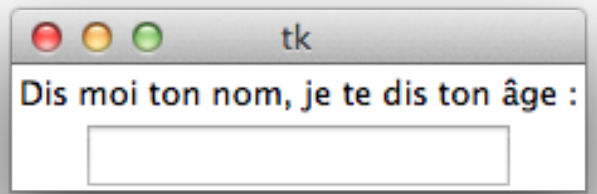


Zone de texte

- Un zone de texte permet à l'utilisateur de saisir un texte

  On définit son contenu avec le paramètre `text`

```
1  window = Tk()
2  text = Label(window, text='Dis moi ton nom, je te dis ton âge :')
3  text.pack()
4
5  Entry(window).pack()
6
7  window.mainloop()
```



Application graphique (1)

- Encapsulation de l'interface graphique dans une classe

Création d'une classe représentant la Frame principale

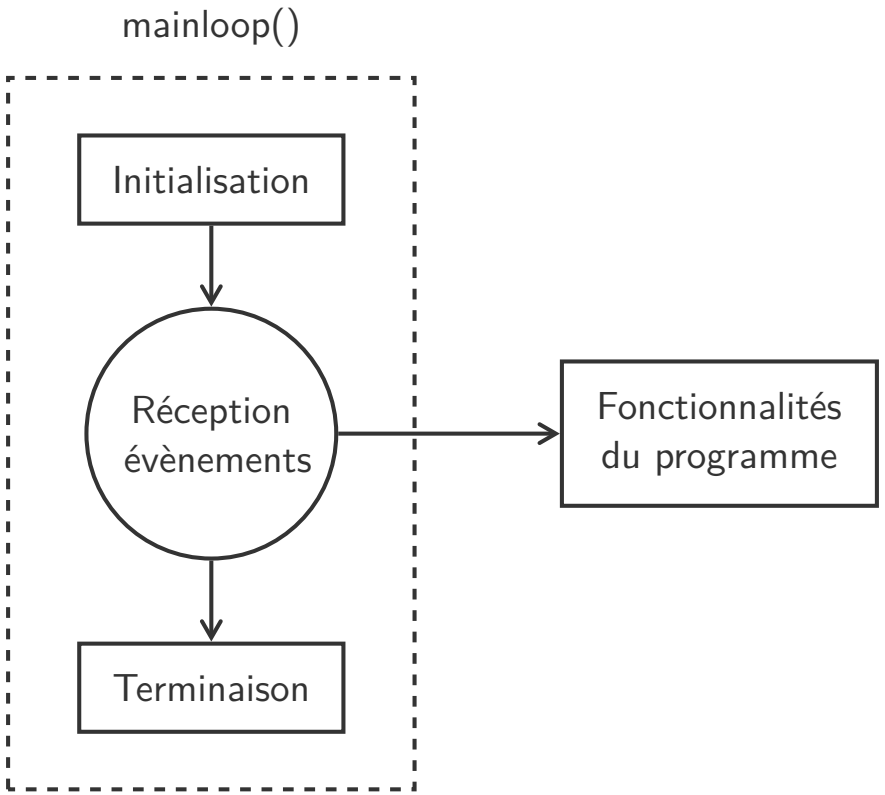
```
1  class App(Frame):
2      def __init__(self, parent):
3          Frame.__init__(self, parent)
4          self.pack()
5          self.createWidgets()
6
7      def createWidgets(self):
8          text = Label(self, text='Dis moi ton nom, je te dis ton âge
            :')
9          text.pack()
10         Entry(self).pack()
11
12 window = Tk()
13 app = App(window)
14 app.mainloop()
```

Programmation évènementielle

Boucle d'évènement

- Des évènements sont produits par l'utilisateur

  Il faut associer les évènements à un gestionnaire

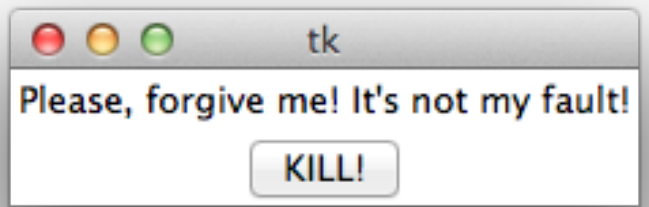


Gestionnaire d'évènement

- Associer une fonction à une action d'un composant

  Via le paramètre command lors de la création

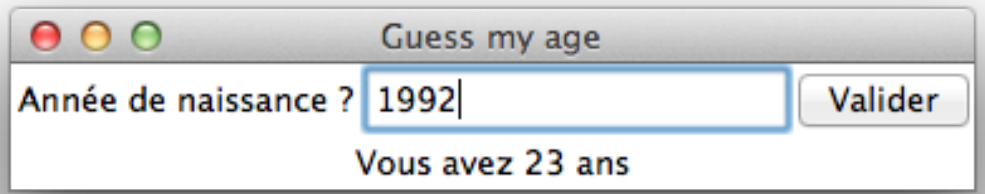
```
1  window = Tk()
2  Label(window, text="Please, forgive me! It's not my fault!").pack()
3  Button(window, text='KILL!', command=window.destroy).pack()
4
5  window.mainloop()
```



Application graphique (2)

- Construire toute l'application graphique dans une même classe
  - Initialisation de la Frame
  - Création et placement des widgets
  - Association des gestionnaires d'évènements

```
1 window = Tk()
2 app = App(window)
3 window.title('Guess my age')
4 app.mainloop()
```



Initialisation de la `Frame`

- Création de la `Frame` et initialisation

  En appelant la méthode `__init__` de `Frame`

- Mise en page, puis ajout des widgets

  En appelant `pack` puis `createWidgets`

```
1  class App(Frame):
2      def __init__(self, parent):
3          Frame.__init__(self, parent)
4          self.pack()
5          self.createWidgets()
6
7      def createWidgets(self):
8          # ...
9
10     # ...
```

Création du gestionnaire d'évènements

- Ajout d'une méthode par gestionnaire d'évènements

  La méthode a accès aux composants de la `Frame`

- Le gestionnaire peut mettre à jour l'interface

  Appels de méthodes des widgets

```
1  def computeage(self):
2      age = 2015 - int(self.__year.get())
3      self.__answer['text'] = 'Vous avez {} ans'.format(age)
```

Création des composants

- Création des composants et ajout dans la `Frame`

  La référence `self` contient la `Frame`

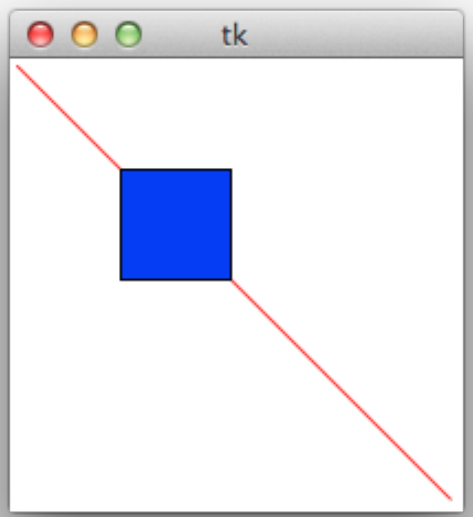
```
1  def createWidgets(self):
2      # Création d'un panneau de contrôle
3      controls = Frame(self)
4      # Création du label
5      Label(controls, text='Année de naissance ?').pack(side=LEFT)
6      # Création de la zone de texte
7      self.__year = Entry(controls)
8      self.__year.pack(side=LEFT)
9      # Création du bouton
10     Button(controls, text='Valider', command=self.computeage).pack
       ()
11     controls.pack()
12     # Zone de réponse
13     self.__answer = Label(self, text='...')
14     self.__answer.pack()
```

Canvas

- Un canvas est un widget pour faire du dessin 2D

  Des méthodes permettent de dessiner des objets graphiques

```
1  window = Tk()
2  c = Canvas(window, width=200,
   height=200)
3  c.pack()
4
5  c.create_line(0, 0, 200, 200,
   fill='red')
6
7  c.create_rectangle(50, 50,
   100, 100, fill='blue')
8
9  window.mainloop()
```



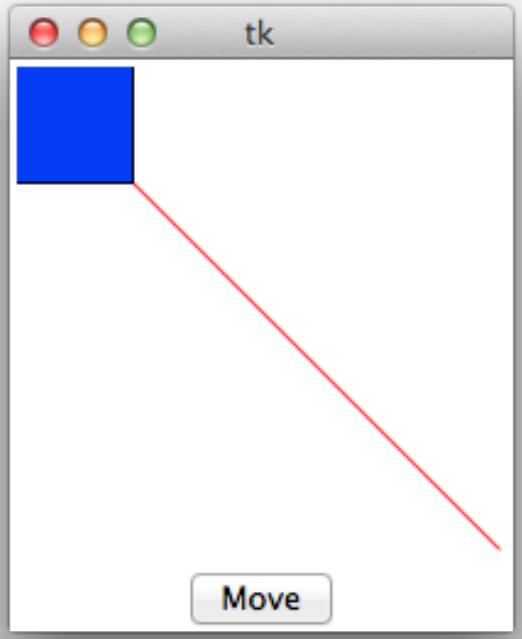
Animation (1)

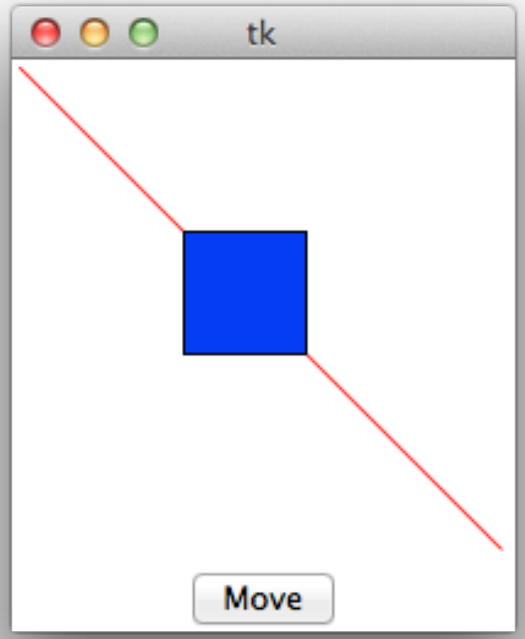
- Création d'une animation par une succession d'images

  Mettre à jour de manière régulière le canvas

```
1  class App(Frame):
2      def __init__(self, parent):
3          Frame.__init__(self, parent)
4          self.pack()
5          self.createWidgets()
6
7      def createWidgets(self):
8          self.__c = Canvas(self, width=200, height=200)
9          self.__c.pack()
10         self.__c.create_line(0, 0, 200, 200, fill='red')
11         self.__rect = self.__c.create_rectangle(0, 0, 50, 50, fill=
           'blue')
12         Button(text='Move', command=self.move).pack()
13
14     def move(self):
15         self.__c.move(self.__rect, 10, 10)
```

Animation (2)





Crédits

- https://www.flickr.com/photos/silvertje/1934375123
- https://www.flickr.com/photos/125720812@N02/15529452622