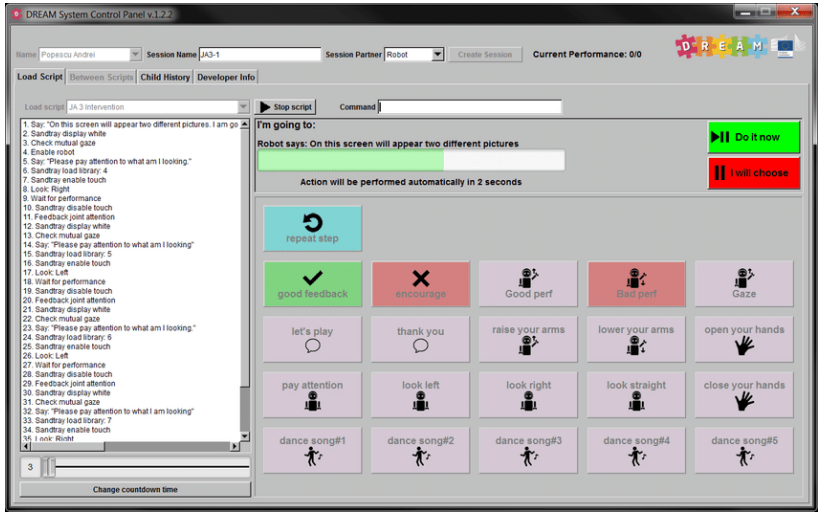
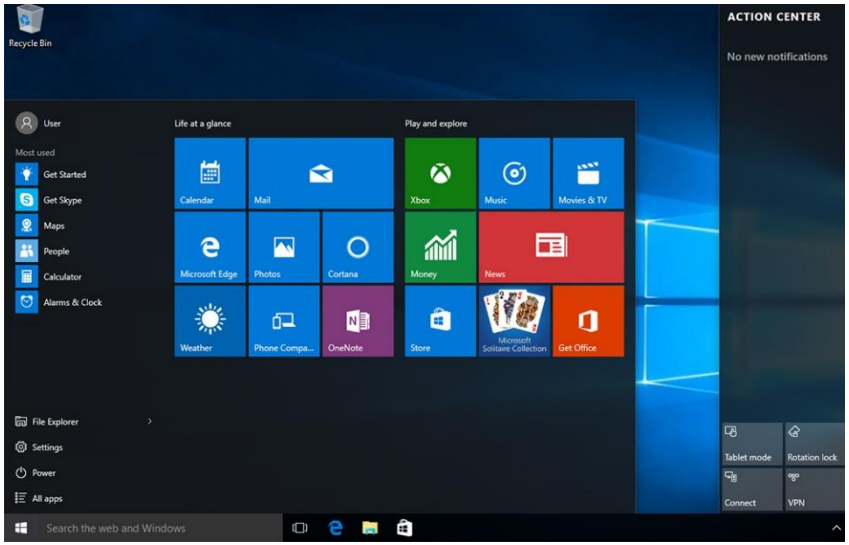
HEH.be Sciences et technologies

Base de programmation

- BA1 Informatique
  Johan Depréter – johan.depreter@heh.be

HEH be
Sciences
et technologies





WALLONIE-BRUXELLES ENSEIGNEMENT
pôle hainuyer

Chapitre 9

Programmation évènementielle

Introduction

- Programmation fondée sur les évènements
- Opposition avec la programmation séquentielle :
  - Programmation séquentielle : Exécute une suite d'instructions
  - Programmation évènementielle : Réagis aux différents évènements qui peuvent se produire

Séquentielle

- L'application a le contrôle
- L'utilisateur va faire ce que lui demande l'application
- Exemples :
  Rentrer une valeur
  Choisir quelle action effectuer
  …

Séquentielle

```
// PROGRAMME
Main()
{
    string saisie;
    Console.WriteLine("Entrez une valeur");
    saisie = Console.ReadLine();
    int valeur = Convert.ToInt32(saisie);
    int carre = valeur * valeur;
    ...
}
```

Evènementielle

- L'utilisateur a le contrôle
- S'il ne fait rien, l'application fonctionne mais rien ne se produit
- Exemples :
  - Cliquer sur un bouton pour ouvrir une fenêtre
  - Afficher le contenu d'un fichier s'il est modifié
  - …

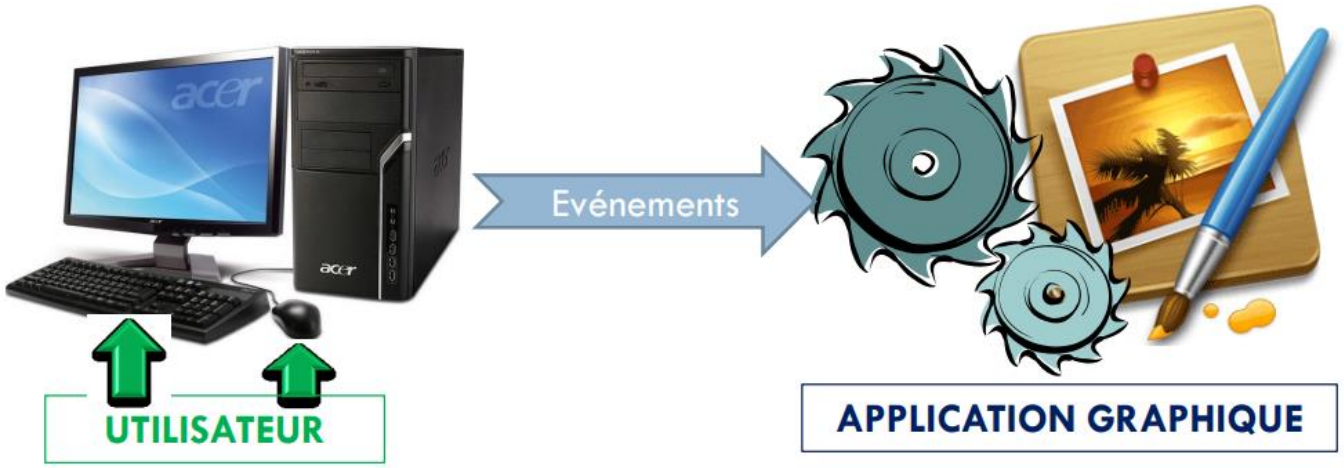
Evènementielle

```
// PROGRAMME
Main()
{
   ...
   while(true)  // tantque Mamie s'active
   {
      // récupérer son action (faire une maille ...)
      e = getNextEvent();
      // traiter son action (agrandir le tricot ...)
      processEvent();
   }
   ...
}
```

Interface graphique

- Application « esclave » de l'utilisateur

- Conséquences :
  - Application prête à réagir



Evènements

- C’est quoi un évènement ?
- Changement d’état dans l’environnement ou intervention explicite de l’utilisateur
- 2 types principaux :
  - Liés aux périphériques
  - Liés au système

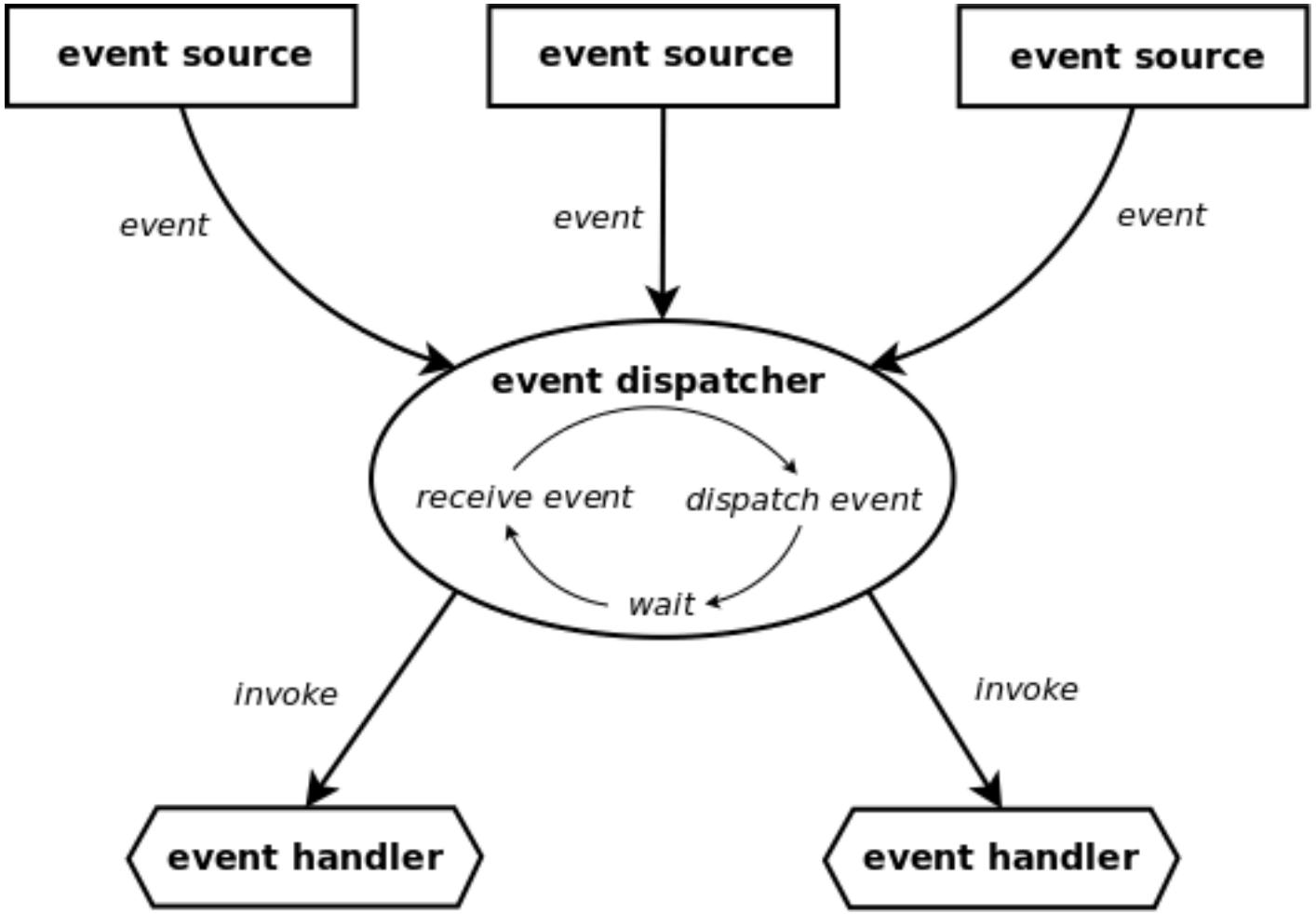
Liés aux périphériques

- Clique de souris sur un bouton
- Frappe au clavier
- Focus/défocus d'une fenêtre
- Sélection d'un objet dans une liste déroulante
- …

Liés au système

- Création / Ouverture / … d'une fenêtre
- Tic d'horloge
- Mise en veille de la machine
- …

Fonctionnement



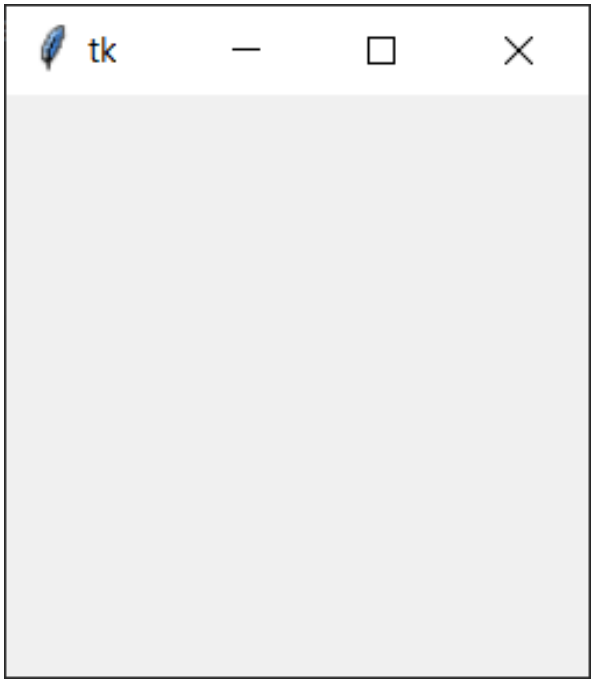
Fonctionnement

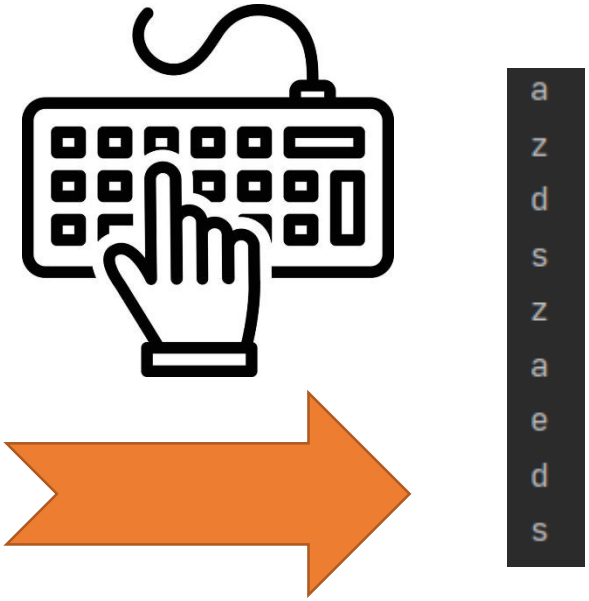
- *Event Loop* : Boucle durant laquelle l'application va vérifier si un évènement a lieu
- *Event Dispatcher* : Envoie l'évènement à l'Event Handler correspondant
- *Event Handler* : Fonction qui va s'exécuter quand l'évènement correspond arrive
- *Bind* : Lier un *Event Handler* à un objet graphique

Exemple

```
import tkinter as tk


def handle_keypress(event):
    """Print the character associated to the key pressed"""
    print(event.char)


window = tk.Tk()

window.bind("<Key>", handle_keypress)

window.mainloop()
```

Exemple





Exception handler

- Les exceptions sont un type d'évènements
- Même fonctionnement qu'un *Event Handler*
- On attend qu'elles arrivent pour gérer

Exception handler

```python
import sys

randomList = ['a', 0, 2]

for entry in randomList:
    try:
        print("La valeur est", entry)
        r = 1/int(entry)
    except:
        print("Oops!", sys.exc_info()[0], "s'est produite.")
        print("Prochaine valeur.")
        print()
print("L'inverse de", entry, "est", r)
```

```
La valeur est a
Oops! <class 'ValueError'> s'est produite.
Prochaine valeur.

La valeur est 0
Oops! <class 'ZeroDivisionError'> s'est produite.
Prochaine valeur.

La valeur est 2
L'inverse de 2 est 0.5
```

Exception handler

```
try:
    # do something
    pass

except ValueError:
    # handle ValueError exception
    pass

except (TypeError, ZeroDivisionError):
    # handle multiple exceptions
    # TypeError and ZeroDivisionError
    pass

except:
    # handle all other exceptions
    pass
```