

Interfaces graphiques

Fiche de synthèse

① LE PRINCIPE

Ouverture hors programme : `tkinter` est fournie avec Python, `pygame` sert aux jeux, `streamlit` aux applications web. Une **interface graphique** crée une fenêtre et ses composants, les **widgets** (boutons, champs de saisie, textes), associe des fonctions aux actions de l'utilisateur, puis lance une **boucle d'événements**, qui attend ces actions et appelle les fonctions correspondantes.

② TKINTER

```
import tkinter as tk

def convertir(event=None):
    f = float(temperature.get())
    c = 5 / 9 * (f - 32)
    resultat.config(text=f"{c:.2f} °C")

fenetre = tk.Tk()
fenetre.title("Fahrenheit en Celsius")
temperature = tk.Entry(fenetre)
temperature.bind("<Return>", convertir)
resultat = tk.Label(fenetre, text="")
bouton = tk.Button(fenetre, text="Convertir",
                    command=convertir)
temperature.grid(row=0, column=0)
bouton.grid(row=0, column=1)
resultat.grid(row=1, column=0)
fenetre.mainloop()
```

Instruction	Rôle
<code>tk.Tk()</code>	crée la fenêtre principale
<code>tk.Label</code>	affiche un texte
<code>tk.Entry</code>	champ de saisie, lu par <code>get()</code>
<code>tk.Button</code>	bouton ; <code>command</code> donne la fonction appelée
<code>bind("<Return>", f)</code>	appelle <code>f</code> à l'appui sur Entrée
<code>pack()</code> , <code>grid()</code>	place le widget, empilé ou en grille
<code>mainloop()</code>	lance la boucle d'événements

Le paramètre `event=None` permet d'appeler `convertir` depuis le bouton, sans argument, comme depuis la touche Entrée, qui transmet l'événement. `command=fenetre.destroy` ferme la fenêtre.

③ PYGAME

Un programme Pygame est une boucle qui, à chaque tour, lit les événements, met à jour l'état du jeu, dessine l'écran, puis attend le tour suivant.

```
import pygame

pygame.init()
ecran = pygame.display.set_mode((600, 400))
horloge = pygame.time.Clock()
x, y = 280, 180
fini = False
while not fini:
    for evenement in pygame.event.get():
        if evenement.type == pygame.QUIT:
            fini = True
    touches = pygame.key.get_pressed()
    if touches[pygame.K_RIGHT]:
        x = x + 4
    ekran.fill((30, 30, 60))
    pygame.draw.rect(ecran, (250, 200, 40),
                     [x, y, 40, 40])
    pygame.display.flip()
    horloge.tick(60)
pygame.quit()
```

- L'origine des coordonnées est le coin **supérieur gauche** : `y` augmente vers le bas. `flip` affiche l'image dessinée, `tick(60)` limite à 60 images par seconde.
- Un clic produit l'événement `MOUSEBUTTONDOWN`, dont l'attribut `pos` donne la position. Le clic est dans un disque de centre (c_x, c_y) et de rayon r si $d_x^2 + d_y^2 \leq r^2$, avec d_x et d_y les écarts au centre : comparer les carrés évite une racine carrée.

④ STREAMLIT

```
import streamlit as st

st.title("Convertisseur de températures")
f = st.slider("Degrés Fahrenheit", -40, 120, 68)
c = 5 / 9 * (f - 32)
st.write(f"{f} °F font {c:.1f} °C.")
```

Streamlit transforme un script en application web : chaque appel `st.` ajoute un composant à la page. On lance le script par la commande `streamlit run mon_fichier.py`, qui ouvre l'application dans le navigateur ; à chaque action de l'utilisateur, le script est réexécuté.

📌 À retenir.

- Une interface crée des widgets, leur associe des fonctions et lance une boucle d'événements.
- Tkinter : `Label`, `Entry`, `Button`, placés par `pack` ou `grid`, puis `mainloop`.