

## Utilisation de la librairie Graphics de OCAML

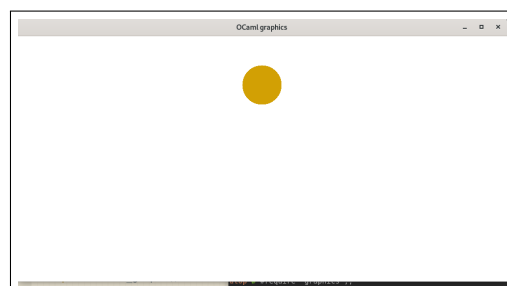
La librairie Graphics de OCAML permet de mettre en place rapidement un affichage au moyen d'une fenêtre graphique. La fonctionnalité de base du module Graphics est de fournir un **canvas** dans lequel on peut dessiner. Le présent descriptif n'a pas vocation à remplacer la documentation : <https://ocaml.github.io/graphics/graphics/Graphics/index.html>, on fournit ici seulement les quelques instructions qui permettent de démarrer, par l'exemple.

```
1 let () =
2   (* Ouvre une fenêtre de 1000*500 pixels (attention espace néc.)* )
3   Graphics.open_graph " 1000x500" ;
4   (* Change la couleur courante *)
5   Graphics.set_color (Graphics.rgb 168 27 3) ;
6   (* Dessine un rectangle *)
7   Graphics.fill_rect 10 100 490 200 ;
8   (* Attend qu'une touche du clavier soit appuyée *)
9   let _ = Graphics.wait_next_event [Key_pressed] in
10  (* On supprime le contenu de la fenêtre graphique *)
11  Graphics.clear_graph ();
12  (* Change la couleur courante *)
13  Graphics.set_color (Graphics.rgb 210 160 4) ;
14  (* Dessine un rectangle *)
15  Graphics.fill_circle 500 400 40;
16  (* Attend qu'une touche du clavier soit appuyée *)
17  let _ = Graphics.wait_next_event [Key_pressed] in
18  (* On ferme la fenêtre graphique *)
19  Graphics.close_graph ()
```

À l'exécution de ce programme, une fenêtre graphique avec un rectangle rouge s'ouvre. Après appui sur une lettre du clavier, l'image est effacée et s'affiche alors un disque jaune. Après un nouvel appui sur une lettre, la fenêtre est fermée (avec utop éviter d'utiliser la croix pour fermer). Si on commente la .11, la 2ème image présentera à la fois le rectangle et le disque.



Premier affichage



Second affichage

**ATTENTION** : Pour utiliser Graphics avec utop on évaluera `#use "topfind";;` puis `#require "graphics";;` afin de charger le module, et ce avant de charger le fichier qui y fait appel. Ce chargement est à renouveler à chaque fois (comme les `#use`).

Pour compiler un fichier code.ml utilisant Graphics en un exécutable code, taper :

```
ocamlfind ocamlpt -o code -linkpkg -package graphics code.ml
```