

Paint4D : Ma palette de dessin

Algorithmes pour la Programmation Graphique
L2 Informatique / UP8

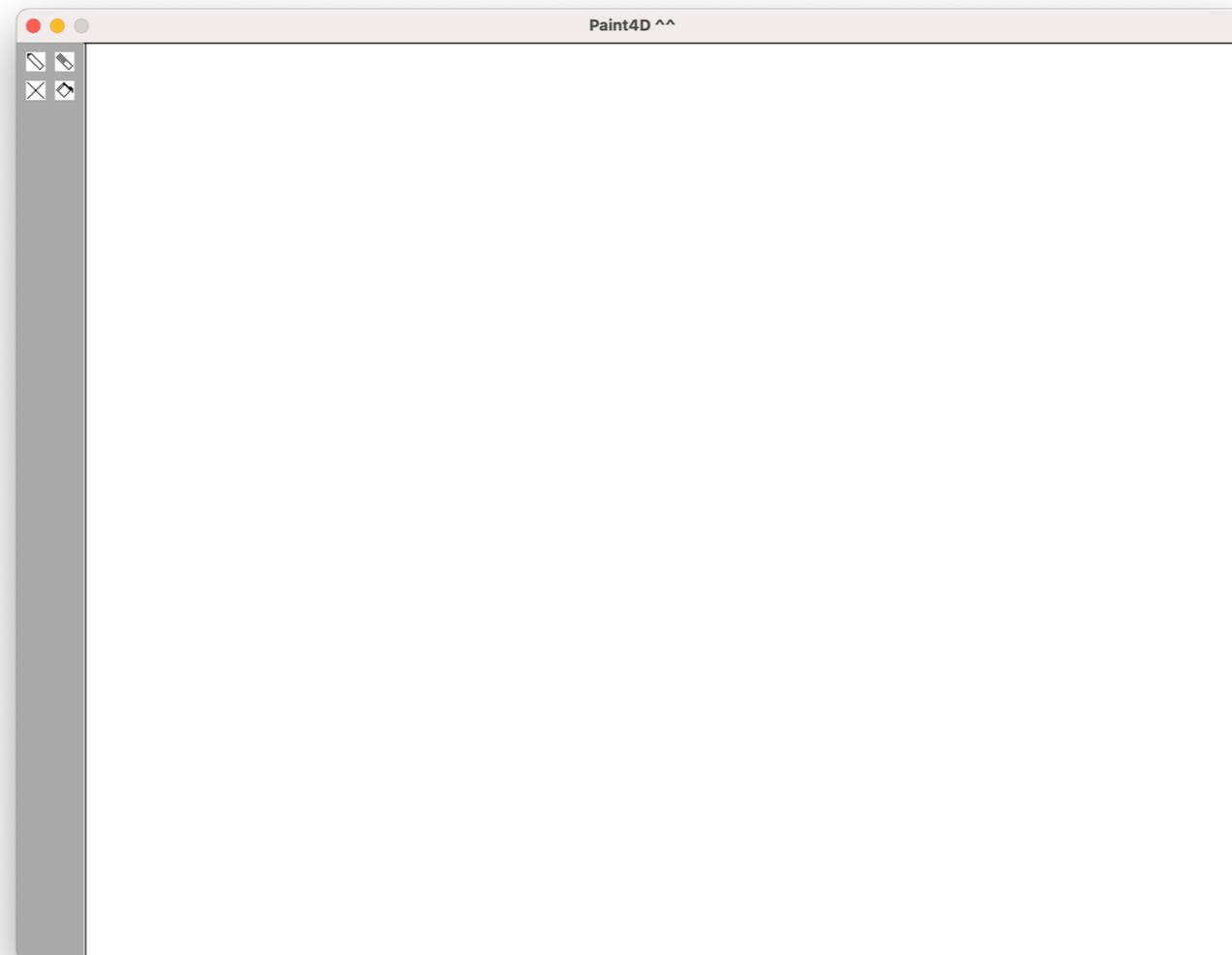
GIT du cours : <https://code.up8.edu/amsi/capg/>

Objectifs (on ajoutera des choses)

- Lire et comprendre le code de départ : `01_GL4D_beginners/sc_00_03_paint-0.1.tgz`
 - `tiny4D.h`, `toolbar.h` puis `window.c`
 - Vous pouvez consulter la documentation générée (*Doxygen* nécessaire) par : `make doc`
- Commencer par **dessiner un point au clic** (click c'est + 😊)
- **Compléter l'interaction** pour pouvoir sélectionner chaque outil
 - A la souris
 - Avec des raccourcis clavier
- Pour **la croix** (icône `_clear`), effacer le Canvas (la toile) au clic/déclik
- Pour **le stylo** (icône `_pen`), faire en sorte que les points émis par la souris, en motion, soient reliés
 - Reprenez et adaptez la fonction `drawLine` réalisée lors de la précédente séance
- **La gomme** est une version plus « grossière » du stylo, dessinant toujours en blanc
- **Le pot de peinture** propage la couleur au clic/déclik « que sur la couleur cliquée »
- ...

Code de départ

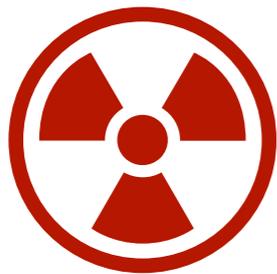
- Décompressez : `01_GL4D_beginners/sc_00_03_paint-0.1.tgz`
 - Tester en allant dedans : `make && ./paint`
 - Lire (il y a des commentaires) : `tiny4D.h`, `toolbar.h` puis `window.c`
 - Sinon vous pouvez consulter la documentation générée (*Doxygen* nécessaire) par : `make doc`
- Posez des questions !



Exécution du code de départ

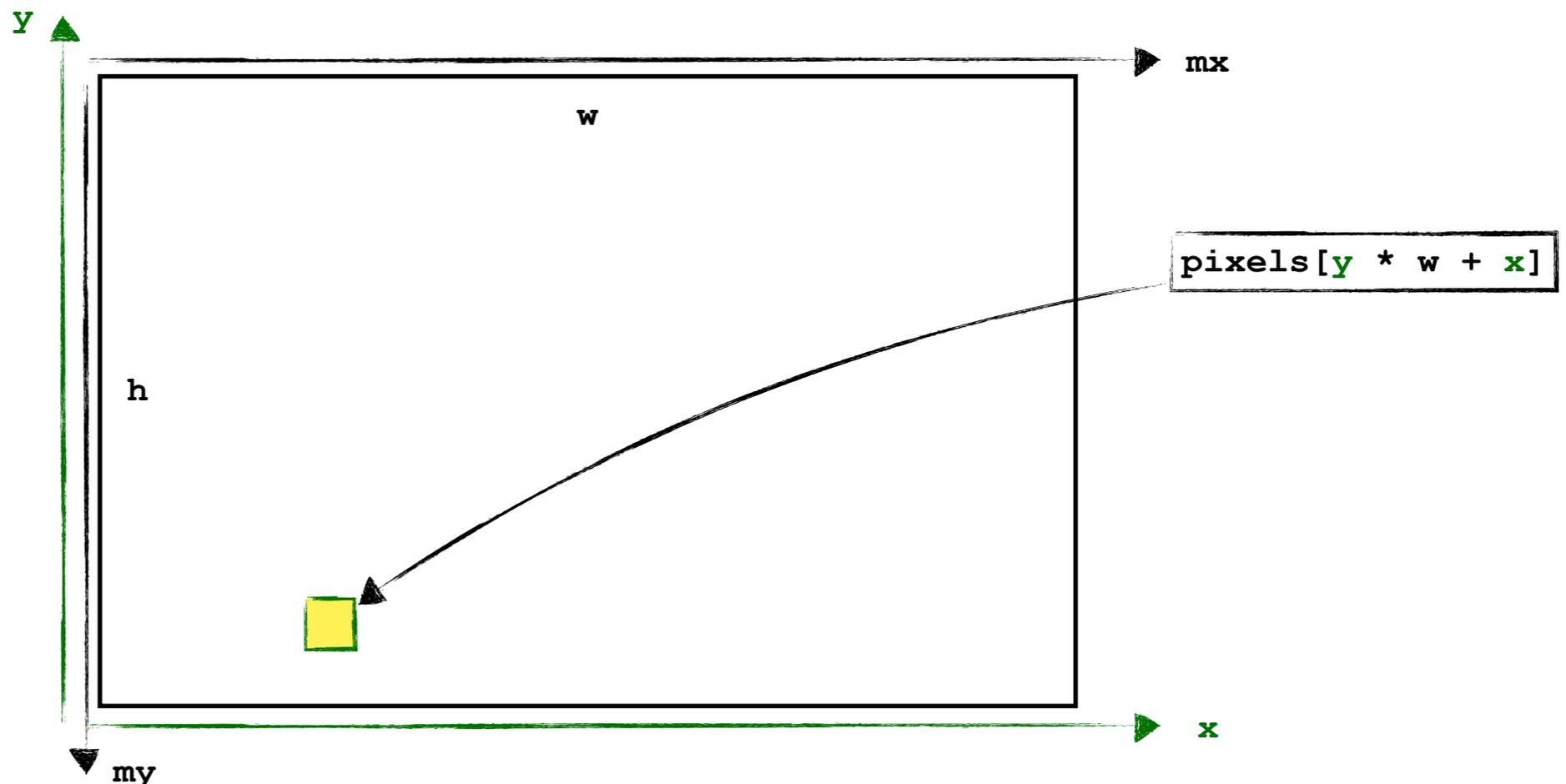
Dessiner un point au clic

- Avant l'appel à la `mainLoop`, mettre en place deux *callback functions* : au clic (et déclic), au *motion* (mouvement de souris avec au moins un bouton déjà cliqué). Voir :
 - `void gl4duwMouseFunc(void(*) (int button, int state, int x, int y) func);` c'est à dire qu'elle attend une fonction qui a la forme :
 - `void foo(int button, int state, int mx, int my) { ... }`
 - `void gl4duwMotionFunc(void(*) (int x, int y) func);` c'est à dire qu'elle attend une fonction qui a la forme :
 - `void bar(int mx, int my) { ... }`
 - La première vous donne le premier et le dernier point (*state* à 1 et à 0) du tracé, la seconde vous donne tous les autres points. Le bouton (*button*) ne nous intéresse pas ici. *mx* pour *mouse-X* et *my* pour *mouse-Y*.

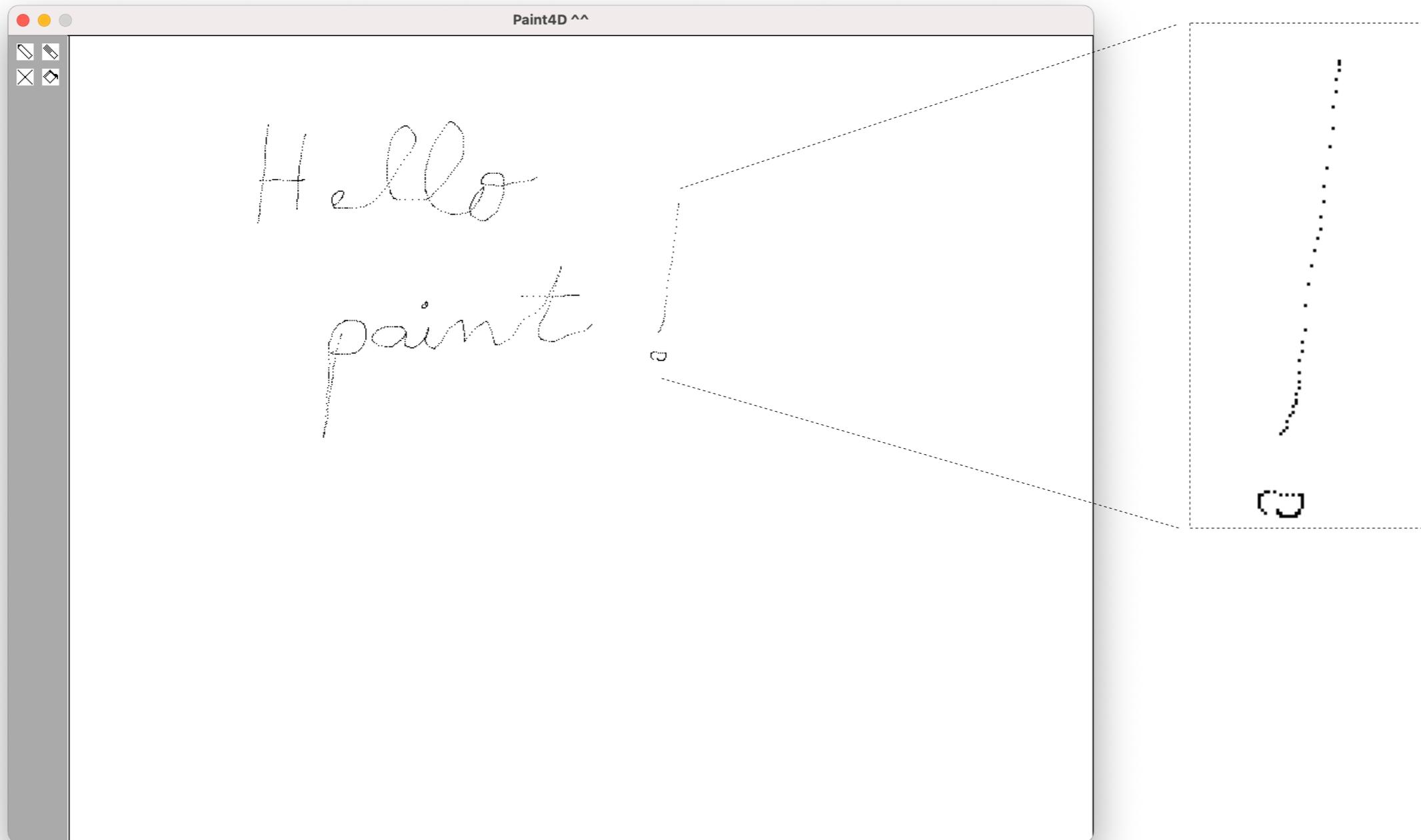


Et ne pas oublier de signaler un changement

`gl4dpScreenHasChanged();`



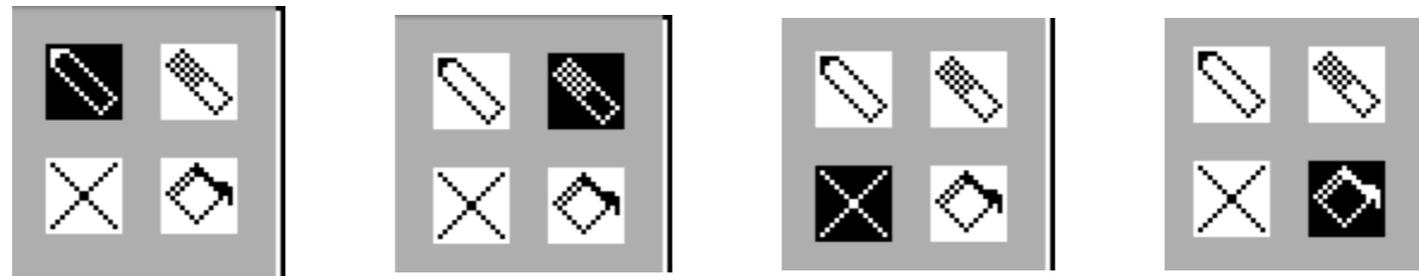
Dessiner un point au clic (suite)



On peut obtenir un résultat équivalent à celui-là.

Compléter l'interaction (+ tout effacer)

- Pour pouvoir sélectionner chaque outil :
 - Utilisez la variable globale (statique) `_selected_icon` : 0 \Rightarrow aucune icône sélectionnée, 1 \Rightarrow la première est sélectionnée, 2 \Rightarrow la deuxième ...
 - Au clavier (`gl4duwKeydownFunc`) et à la souris (`gl4duwMouseFunc`)
- Avec des raccourcis clavier : 'p' pour le stylo (**pen**), 'e' pour la gomme (**eraser**), 'c' pour tout effacer (**clear**) et 'f' pour le remplissage par pot de peinture (**fill**).
- A la souris : accepter définitivement la sélection après clic et déclic sur la même icône
 - Astuce : voir la fonction `draw_icon`, j'ai caché l'identifiant de l'icône dessinée dans la composante alpha du pixel.



Résultat attendu : pouvoir passer de l'un à l'autre

Implémenter le « tout effacer »

Attention, la croix (**clear**) peut avoir un comportement différent. Deux propositions :

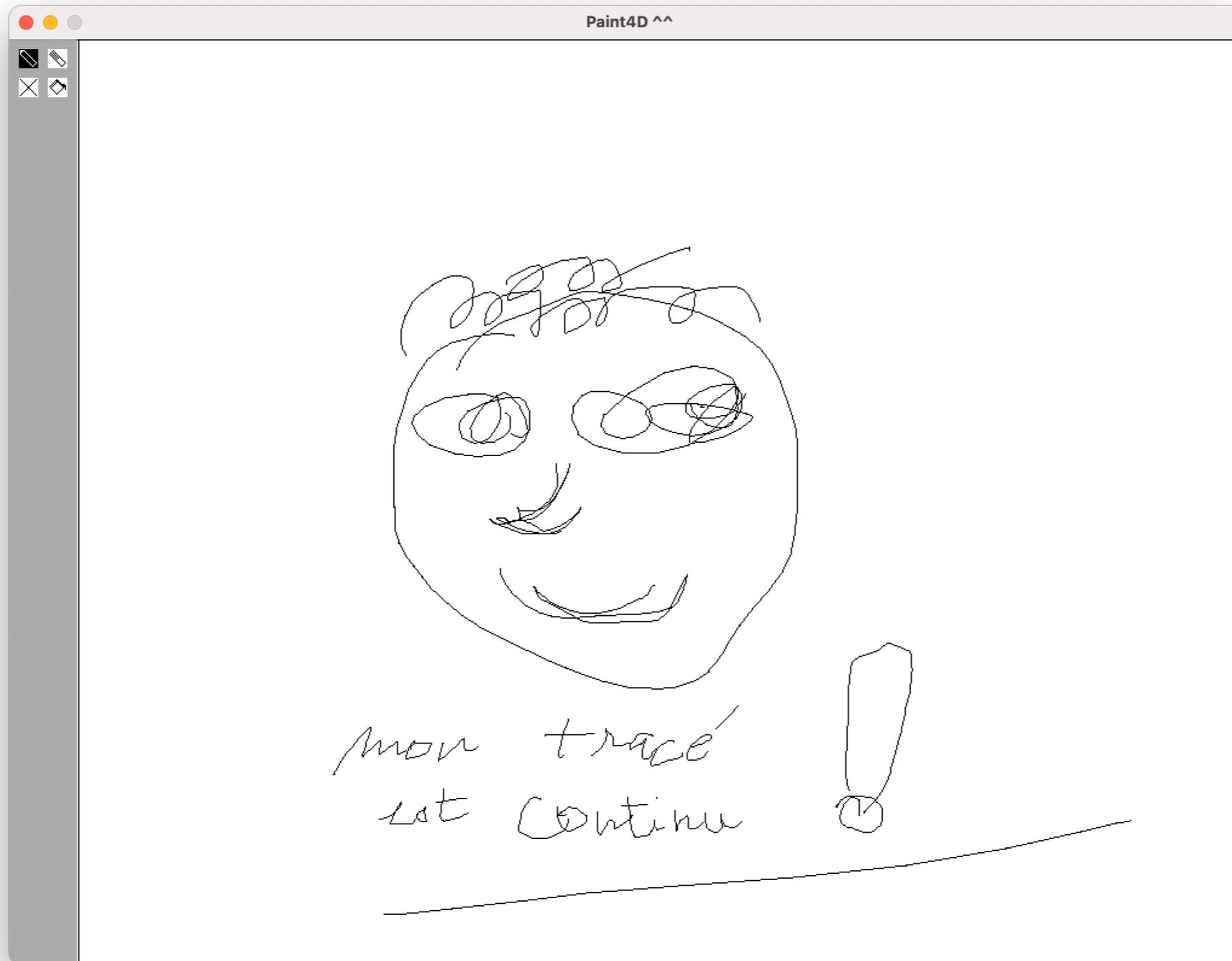
- Une fois le clic validé, l'effacement doit être déclencher, puis soit on revient vers aucune icône sélectionnée, soit on revient à la précédente icône sélectionnée.
- Une fois le clic validé (icône croix sélectionnée), on doit cliquer dans la zone de dessin pour déclencher l'effacement avec la couleur `_cur_bgcolor`.

Compléter le tracé de traits

- Le **stylo** (icône `_pen`) doit permettre de réaliser un tracé en reliant deux à deux les points émis par la souris.
 - 1er point, émis par `g14duwMouse` (où `state == 1`) ; 2e point émis par `g14duwMotion` ; 3e point aussi ; 4e ... dernier point émis par `g14duwMouse` (où `state == 0`)
 - Il faut donc stocker le point précédent (ou premier) et à partir du 2e relier avec le précédent puis le mettre à jour avec l'actuel
- Pour tracer un segment entre deux points, vous pouvez reprendre et adapter la fonction `drawLine` réalisée lors de la précédente séance

```
static void drawLine(unsigned char * image, int w, int h, int x0, int y0, int x1, int y1, unsigned char color) {
    int u = x1 - x0, v = y1 - y0;
    if(abs(u) > abs(v)) { /* octan 1, 4, 5, 8 */
        float y = 0.0f, pente;
        int x;
        int pas = u < 0 ? -1 : 1;
        pente = v / (float)abs(u);
        for (x = 0; x != u + pas; x += pas) {
            if(in_screen(x + x0, (int)y + y0, w, h))
                image[(y0 + ((int)y)) * w + x + x0] = color;
            y += pente; /* quand x augmente de 1 y augmente de pente qui est <= 1 */
        }
    } else { /* octan 2, 3, 6, 7 */
        float x = 0.0f, ipente;
        int y;
        int pas = v < 0 ? -1 : 1;
        ipente = u / (float)abs(v);
        for (y = 0; y != v + pas; y += pas) {
            if(in_screen((int)x + x0, y + y0, w, h))
                image[(y0 + y) * w + (int)x + x0] = color;
            x += ipente; /* quand y augmente de 1 x augmente de l'inverse de la pente qui est <= 1 */
        }
    }
}
```

Compléter le tracé de traits (suite)

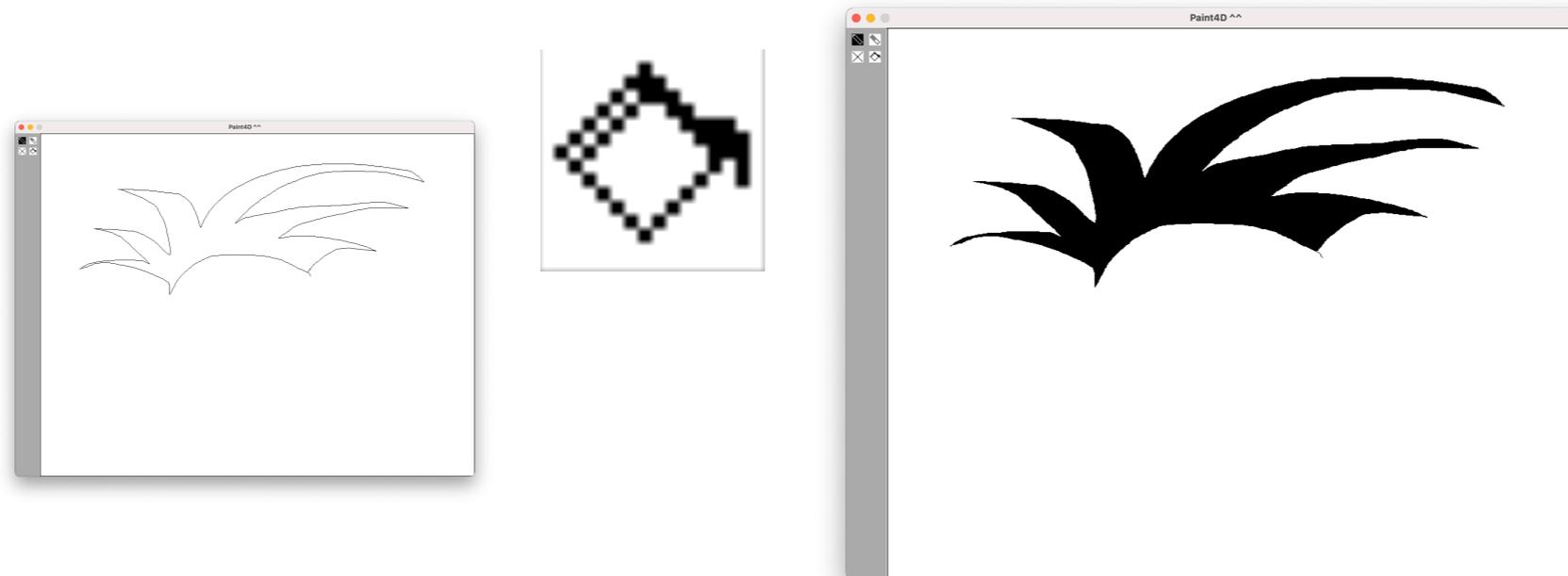


La gomme

- Faire quasiment la même chose que le stylo. Deux différences :
 - Utiliser `_cur_bgcolor` à la place de `_cur_fgcolor`
 - Au lieu de colorier qu'un seul pixel, colorier un carré de 3x3 ou de 5x5 autour du pixel

Le pot de peinture 🤔

- A vous de jouer
- Piste :
 - Récupérer la couleur du pixel cliqué
 - Propager sur le voisinage (4-connexe ou 8-connexe) tant que la couleur du voisin est « exactement » la même que celle du premier pixel cliqué (point précédent)



Exemple de ce qui arrive quand on fait « pot de peinture » dans une forme fermée.
Si la forme n'est pas fermée, la peinture se propage partout !

BONUS (compliqué) :
faire en sorte de remplir avec un motif (en dur)



Aller plus loin 🚀🚀🚀

- On aimerait pouvoir sélectionner le couleur de l'outil (`_cur_fgcolor`), voire aussi la couleur du fond (`_cur_bgcolor`).
- PROPOSITION :
 - Ajouter une icône qui permet de faire apparaître/disparaître une palette dans la toolbar
 - On peut aller chercher une palette en dur comme la palette Web
 - Faire en sorte que la couleur cliquée, bouton gauche, soit la `_cur_fgcolor`, et bouton droit, la `_cur_bgcolor`

Exemple

