

2017-09-06

L'idée d'un pinceau est de produire un tracé ayant certaines caractéristiques. Le pinceau idéal produirait des effets en temps réel, chose que j'ai presque réussi pour de simples tracés.

Tester une application graphique comme si on l'utilisait pose de sacrés problèmes. Passons outre les détails parce que c'est long et ennuyeux et très loin du sujet de production d'images de ce blogue, mais je mentionne ce fait pour expliquer pourquoi je crée une application en parallèle destinée à tester le code spécifique de création des traits au pinceau : le programme va tracer un trait entre deux points prédéterminés ce qui me permettra d'observer les valeurs des variables pendant la progression du tracé et trouver des bogues ou, mieux, des moyens d'optimiser l'ensemble. C'est aussi un moyen d'isoler le code qui dessine de celui qui gère l'attribution spatiale des points et les moments où le tracé commence et se termine et, éventuellement, la pression du stylet. Plus les séparations sont claires entre chaque groupe fonctionnel de l'application, plus il sera facile de faire progresser l'ensemble.

Windows est en charge de donner à l'application les indications de l'emplacement de la souris ou du stylet. Ces indications surviennent plusieurs fois par seconde et à chaque fois mon petit programme est censé pouvoir accomplir tout son travail d'ici la prochaine indication de Windows. Si mon programme traîne de la patte, Windows est patient et laissera tomber, au besoin, des événements de souris. Ces manques se traduisaient au début par des lignes incomplètes à l'écran.

L'essentiel du travail de dessin consiste à avoir un point de départ et de fin et, entre les deux, un tracé. Le point de départ correspond au moment où on appuie sur le bouton de la souris ou lorsqu'on pose le stylet sur la surface de dessin. Le point de fin c'est, évidemment, le relâchement du bouton de souris ou la relève du stylet.

Au pinceau, le début et la fin du tracé ne sont pas identiques : quantité de facteurs physiques entrent en jeu : type et quantité de matière (huile, acrylique, aquarelle, etc. ; pinceau presque sec ou surchargé...), angle du pinceau, longueur et résistance de poils et ainsi de suite. Tous ces facteurs sont simulés par différents procédés dans les applications de dessin. Ce que je cherche à faire n'est pas de reproduire fidèlement la matière dans le monde des pixels mais de découvrir des choses nouvelles (pour moi en tout cas), un langage visuel qui me satisfasse davantage.

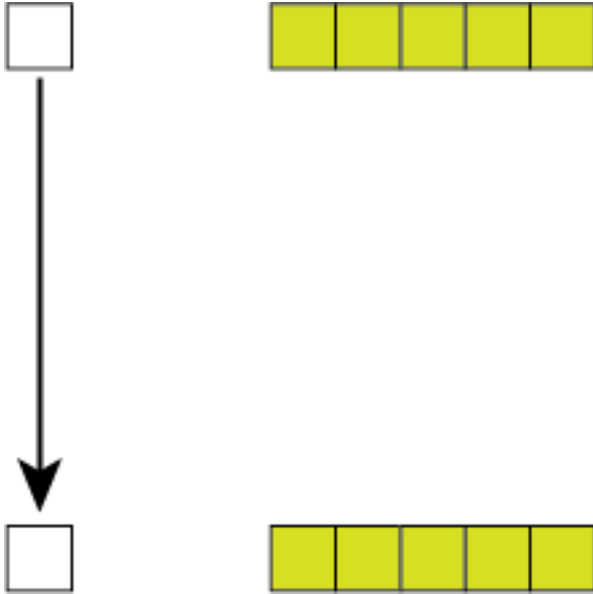
Je pressentais un peu que le travail de construction de ce programme me pousserait à observer avec plus de finesse les détails des images qui existent et celles que j'imagine. Il est maintenant question d'un langage plus profond, riche et intime. De fait, je pense que mes observations peuvent être partagées ici.

À sa plus simple expression, un pinceau numérique est un ensemble virtuel de pixels qui forment une image :

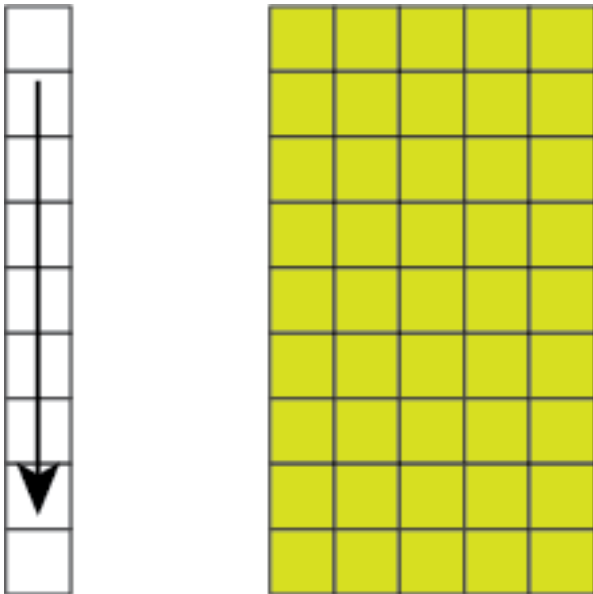


Cette image sera « plaquée » sur la surface de dessin. Répétée plusieurs fois, elle formera un trait.

Le tracé du trait est déterminé par au moins deux pixels :



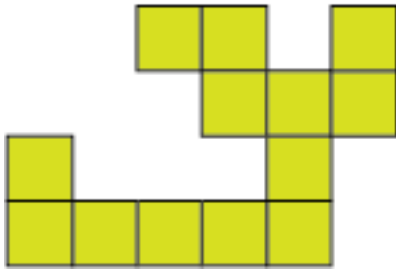
Si la distance entre les deux est trop grande, on peut procéder par calculs pour déterminer tous les points où on doit apposer l'image du pinceau :



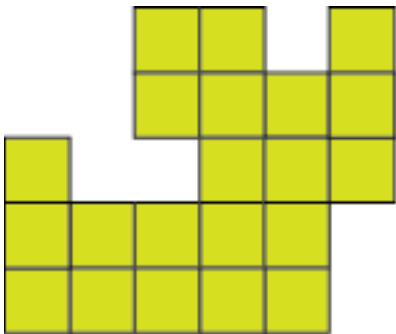
C'est très joli, mais très basic et déjà, ça demande beaucoup de travail de code car, à un ordi, il faut tout lui dire et dans les moindres détails.

Mais déjà, à partir de ce travail de base, on peut imaginer des variations. On peut songer à déplacer des pixels du pinceau et à faire varier la couleur. Ici, à nouveau, peut entrer en jeu les règles des automates cellulaires qui dicteraient des comportements selon les variables de l'environnement (les pixels environnants, couleurs de la surface à repeindre, distance, vitesse, etc.).

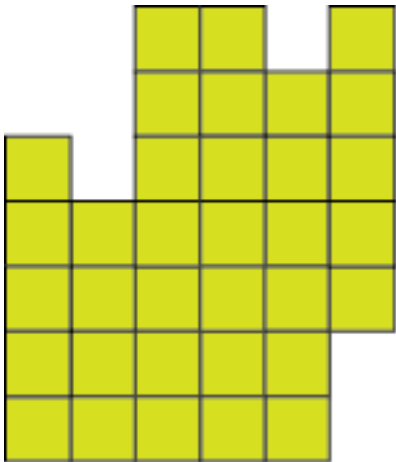
Dans les cas d'images pour pinceaux plus complexes, la complexité disparaît au fur et à mesure de placages consécutifs :



2 placages :

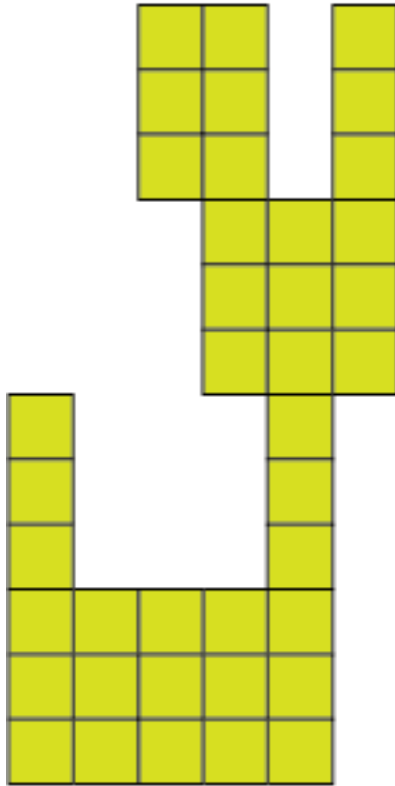


5 placages :



On peut aussi préférer étirer l'image du pinceau. L'étirement se fait progressivement ce qui veut dire qu'à chaque fois que le pinceau avance d'un pixel à l'écran il faut tout redessiner à partir du début du trait, ce qui ne fait que surcharger le travail du processeur. L'effet est cependant un peu plus réaliste.

Image étirée sur 3 pixels :



Dans les cas présentés ici, ça ne donne que quelque chose s'apparentant à un motif bâclé, sans trop d'âme ni de vitalité. Je cherche quelque chose de plus complexe, disons organique, sans pour autant me cantonner dans la reproduction de la matière.

Pour cela il me faut tester quantité de variables, de variations, de valeurs en évitant d'être encombré par le reste de l'application. Les composés les plus réussis pourront être intégrés comme fonctionnalités à Paintual.

En parallèle, je construis un système de collecte, stockage et classification d'images. Je tiens à ce que mon programme puisse tout seul trouver des images similaires à ce que je suis en train de dessiner et les compléter automatiquement.

Donc du code, du code, du code et pas beaucoup de dessins à montrer pour le moment.