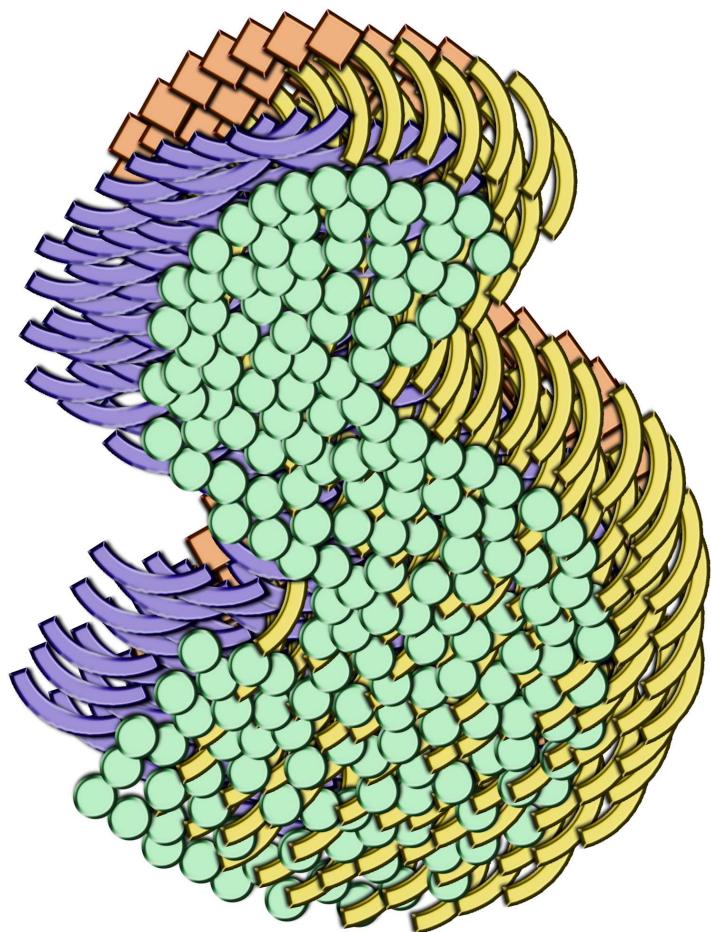




Reprogrammer : lettre S



Screenshot du programme CD-rom retravaillé sur paint. →

Květa Pacovská, plasticienne tchèque, est l'autrice d'éditions s'adressant à un public-lecteur « enfant » avec qui elle entretient un jeu mêlant formes géométriques, figures en papier découpé, reliefs et autres systèmes tangibles habillés par des palettes de couleurs variées, souvent dominées par un rouge vif.

Dans son œuvre « Alphabet », publié en 1996, Květa invite à la préhension de dessins de lettres alphabétiques. Un dessin dont elle est l'autrice, à la silhouette grasse et grossière, facilement reconnaissable car imposant dans la page, mais c'est un dessin qu'elle déforme, modifie en l'inscrivant dans différents modes de représentation.

C'est un système bien précis qui sera étudié, puis reconstruit ici, le long de cet article comme parcours d'une démarche archéologique, reconduite en démarche de réécriture contemporaine.

Ce système est présent dans le livre de Květa par des pages endossées de points éparpillés, presque tous équidistants les uns des autres, mais comme contenus dans une bordure invisible, faisant apparaître la silhouette de la lettre qu'ils forment en nuage.

Une deuxième édition de ce système existe, interactive cette fois-ci et accessible

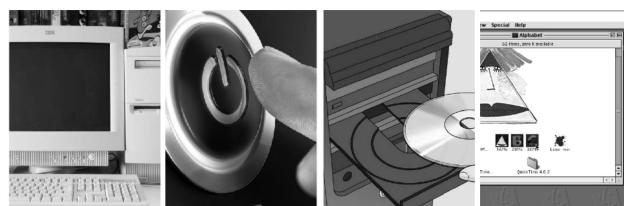
CD-Rom, développé par Dada Media, produit par NHK Educational Corporation, en 1999. La même année, quelques mois suivant la parution du « théâtre de minuit », un autre CD-rom interactif également adapté d'une œuvre de Květa Pacovska, développé et édité par le même éditeur, mais une équipe de développement différente.



1.Photo du livre « Alphabet » /2.Photo de Květa Pacovska.

Le livre de Květa existe encore, aussi longtemps que tous ces exemplaires papiers résisteront aux coups fatals du temps et des éléments environnants. Une force que la technologie du CD-rom peut envier à celle du papier et de l'imprimerie. En effet, dans ce monde (industriel) du multimédia, les technologies (d'écriture et de lecture) du CD-Rom sont presque définitivement mortes.

Prenons un jour de Janvier 2001 au hasard comme contexte de simulation. Deux ans après sa publication, pour lire le CD « Alphabet », il suffisait d'insérer le disque dans le compartiment adéquat à tout système informatique personnel de cette époque, le système d'exploitation de la machine (Windows 95 ou mac OS 9) reconnaissait naturellement l'objet une fois chargé et nous proposait des options de lancement, ou le lançait sans interaction demandée. C'était un jeu d'enfant.



Aujourd'hui, un jour quelconque de Janvier 2023, (à moins de disposer d'une machine de l'époque et du système encore fonctionnel) la démarche ne sera plus du tout la même.

Les systèmes d'exploitation ont évolués, le hardware également, et toutes ces modifications ont eu lieu petit à petit en acceptant l'idée, au nom du progrès technique, que pour l'industrie du multimédia, le maintien des technologies de disques compacts de données n'est plus pertinent. Ce maintien, comme la sauvegarde du savoir enfermé dans ces boîtes noires, laissé à la dérive, appartient à toute personne passionnée et capable, qui serait touchée par une certaine démarche bénévole d'émulation. Voici deux d'entre eux : Cat_7 et Ronald P. Regensburg.



Dans un contexte de workshop intitulé « Reprogrammer », l’œuvre numérique « Alphabet » fût évoquée comme objet d’étude. À cet instant, il est important de détailler quelques étapes du travail qui a eu lieu.

Le groupe dispose d’une machine d’époque fonctionnelle équipée de mac OS 9 et permettant de jouer tout CD-Rom/programme susceptible d’être analysé. Il est suggéré que chaque membre du groupe puisse faire fonctionner des versions numériques de ces CD-Rom/programmes sur leur machine personnelle afin de gagner en autonomie. Pour ce faire, il nous faut aborder quelques concepts.

fichier ISO / image disque

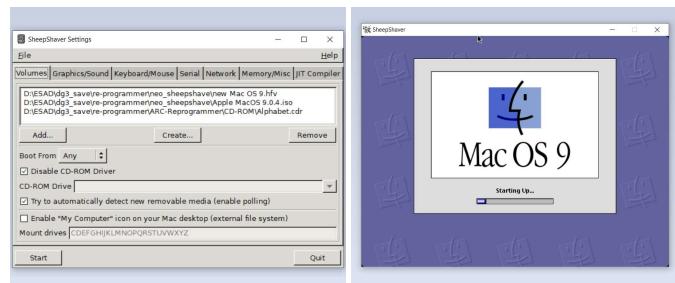
Un CD-Rom, s’il existe sous la forme d’une galette physique en polycarbonate, peut exister virtuellement sous la forme d’un fichier iso. Une image disque (ou image ISO) est un (voire plusieurs) fichier(s) archive proposant la copie conforme d’un disque optique ou magnétique (tel qu’il serait écrit sur celui-ci). Donc qu’il s’agisse du CD-Rom « Alphabet » ou du programme d’exploitation mac OS 9, il possible de les copier et conserver intégralement et sans altérations sous ces formats .iso .

L'émulation

En informatique, l'émulation fait référence au processus de création d'un environnement qui émule les propriétés d'un système, matériel ou logiciel, à l'intérieur d'un système complètement différent. Cela se fait via un émulateur, un programme informatique conçu pour imiter les propriétés d'un système « invité » à l'intérieur d'un système « hôte ».

Donc lancer « Alphabet » sur une machine moderne revient à suivre ces étapes : obtenir un émulateur (sheepshaver), obtenir un fichier iso de MAC OS 9 et d'« Alphabet », charger ces fichiers dans l'émulateur et le lancer. Ensuite tout se fait comme sur une machine d'époque.

| | | | |
|-----------------------|------------------|-------------|---------------|
| Mac OS 9.hfv | 10/03/2022 14:30 | Fichier HFV | 40 960 Ko |
| Mac OS ROM.rom | 10/03/2022 12:22 | Fichier ROM | 1 901 Ko |
| MACOS9.0.4 | 10/03/2022 12:03 | Fichier 4 | 40 960 Ko |
| new Mac OS 9.hfv | 11/11/2022 16:04 | Fichier HFV | 1 024 000 ... |
| SheepShaver.exe | 10/03/2022 11:28 | Application | 2 823 Ko |
| SheepShaver_nvram.dat | 11/11/2022 16:04 | Fichier DAT | 8 Ko |
| SheepShaver_prefs | 10/03/2022 14:42 | Fichier | 2 Ko |
| SheepShaverGUI.exe | 10/03/2022 11:28 | Application | 383 Ko |
| startup.wav | 10/03/2022 11:28 | Fichier WAV | 203 Ko |



Ce programme abrite un réseau de pièces musicales, animées, 2D, jouant systématiquement avec une lettre et dont l'entrée est associée à la pression d'une touche du clavier, entre a et z. Nous nous intéresserons à l'entrée de la touche « s ».

Comme expliqué précédemment, l'aspect des points en relief du livre a été retranscrit sur écran. Des particules à l'aspect métallique, caractérisées par une ombre propre et une ombre portée, qui s'oriente en fonction de la position du curseur. Une première séquence animée se déclenche, ces particules arrivent une-à-une à l'écran, comme des billes métalliques qui roule sur une surface blanche, pour se placer fixement, formant le dessin d'un « S ».



Les « billes » ont une vie propre (passive), elles se rapprochent ou s'éloignent simultanément du centre de l'écran par fréquence de 5 secondes comme une respiration très lente. Ces mouvements ont lieux avec ou sans interactions souris. Une propriété de magnétisme au curseur est activée lors de l'activité de ce dernier. Si l'inactivité se poursuit au-delà de 5 secondes, des points « gravitationnels » ou « magnétiques »

caractéristiques identiques au curseur de la souris, comme un fantôme immobile de ce dernier. Lors d'une pression sur une touche du clavier (touche de lettre uniquement), chacune des billes, à tour de rôle, s'empresse de changer de position (x et y) en se déplaçant sur l'écran (non pas en se téléportant) afin de former la nouvelle lettre désignée. Enfin, en guise de clôture, un click gauche de la souris permet de finir la séquence en laissant tomber chaque bille simultanément, et les fait rebondir sur un terrain hors-champs, situé juste à la limite basse de l'écran, le temps qu'un léger vent les fasse sortir de l'écran et nous emmène sur une autre séquence, déterminée par la dernière lettre formée.

Tout cela est possible à l'époque grâce au logiciel de Macromedia, nommé en 1999 « Macromedia Director 7 ». Un programme très populaire à l'époque pour la création de CD-Rom en tout genre et notamment des jeux. Une de ses grandes forces est l'intégration d'un langage de scriptage appelé Lingo, développé quelques années plus tôt pour les versions entre 2.2 et 4.0; et natif au logiciel. Director fonctionne selon une analogie cinématographique, on y crée des « movie », soit des séquences qui ont une durée et une vitesse de lecture, en sommes une « timeline » ou « ligne de temps » qu'on retrouve sur tout logiciel de montage vidéo ; sur laquelle on place des objets, statiques ou dynamiques. À partir de Mars 1994 (4.0), ces objets pouvaient être créés par script Lingo (POO¹), et partager en plug-in « Xtras ».² Associé à la POO, le concept de « Sprite » doit être également introduit, car c'est à l'aide d'objets Lingo contenant ces « Sprites » que certaines animations sont possibles, dont la modification des ombrages des particules.

Un Sprite est un élément graphique intégré dans une scène plus grande. D'une certaine manière, il

1. La programmation orientée objet (POO), ou programmation par objet, est un paradigme de programmation informatique. Elle consiste en la définition et l'interaction de briques logicielles appelées objets ; un objet représente un concept, une idée ou toute entité du monde physique, comme une voiture, une personne ou encore une page d'un livre.
2. Une technologie propre au langage Lingo permettant le

du reste. Si nous désirons animer le contenu graphique de cet objet, il faudrait utiliser plusieurs Sprites pour décomposer le mouvement en une série d'images, le tout intégré dans un objet qu'on appelerait : « Sprite sheet ».

Pour ce qui est relatif aux comportements des particules, on parle de moteur physique. C'est un programme qui décrit mathématiquement des lois de la physique comme la gravité, ou la restitution de forces et permet de les appliquer à des environnements habités par des particules sujettes à ces lois. Un coup d'oeil aux codes sources du programme CD-Rom nous a permis de comprendre que la physique présente était directement implémentée par le développeur sans passer par des librairies Xtras.

Matter.js est un moteur physique conçu pour le web. Cette librairie gère presque tout, de la création des objets graphiques, leurs propriétés interactives, jusqu'à l'affichage à l'écran. Elle est développée par une personne présentée sous le nom de « liabru », prénom « Liam », développeur freelance basé à Londres; et mise en ligne, ouverte à la collaboration, sur la plateforme github.

Regardons ce que la librairie offre comme possibilités, en répondant à une série de problématiques.

→ **Comment créer une particule qui serait restreinte dans son déplacement ?**

Les contraintes (= constrains), sont des objets de la librairie qui prennent en paramètre deux particules, une fixe, et une autre « libre » accrochée au bout d'un lien, partant de la première particule. Ce lien possède des propriétés d'élasticité, de friction et de longueur, 3 valeurs avec lesquelles jongler pour obtenir la sensation de détachement désiré.

→ **Comment attirer des particules au curseur ?**

Il existe un plugin à cette librairie permettant de désigner une particule comme « attractor », une qui attire toutes les autres. En lui attribuant des coordonnées x et y identiques à celle de notre curseur sur le plan 2D, nous obtenons très facilement ce phénomène d'attraction au curseur.



→ **Comment habiller nos particules d'une texture ?**

Lors de la création d'un objet, il est possible de lui passer en paramètres un objet : « options ». Ce nouvel objet contient lui-même un autre objet nommé « render », contenant lui-même un objet « sprite » qui, lui, possède une propriété « texture » à laquelle on attribut l'url d'un fichier image (.png ou .jpg).



→ **Comment ajuster cette texture en fonction de la position du curseur ?**

La méthode actuellement utilisée est de calculer un angle de rotation propre à chaque particule en prenant en compte les valeurs de position x et y à la fois des particules(x_1, y_1) et du curseur(x_2, y_2). Cela se traduit par la fonction suivant :

`atan2(y2 - y1, x2 - x1) * 180 / PI`

Le résultat est la rotation individuelle des particules, comme si elles se tournaient pour toujours faire face au pointeur. Une autre manière serait de manipuler une sprite sheet comme le faisait le programme original.

Maintenant que l'analyse du programme original est faite, et l'outil web permettant une reprogrammation fidèle a été déterminé, il reste une chose à faire, commencer la reprogrammation, mais avec quelles données ?

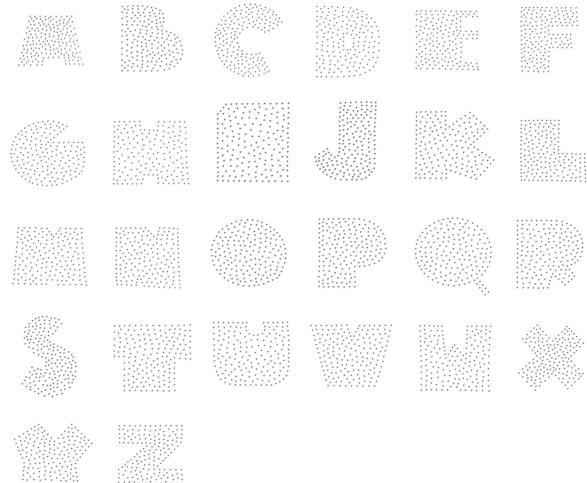
Il serait envisageable de prendre des captures d'écran de chacune des lettres, et de les passer comme calque dans un programme de dessin vectoriel comme illustrator ou inkscape, afin de dessiner par dessus un « polygone » ou une « polyline », deux attributs récupérables dans un fichier .svg et ainsi obtenir les coordonnées de tous les points composants ces dessins. Mais comme l'illustration en page (3) le démontre, la lettre « S » contient 180 points, nombre constant pour les 25 autres caractères ; ça fait long !

Posons le problème en une question : il y a-t-il un moyen de lire une série d'images, de reconnaître les zones d'ombres dans ces images, de les compter, et d'obtenir leur position x et y ?

OpenCV répond pleinement à ce problème. C'est une librairie libre, développée par Intel, spécialisée dans le traitement de données images. Donc en quelques lignes de code en langage python, nous pouvons lire tous les fichiers images d'un dossier cible, passer ces images en seuil pour simplifier le traitement, appeler une fonction de la librairie capable de déterminer des zones qui se détacheraient du fond, les compter, et stocker leurs valeurs de position x et y dans un objet. Un objet qui sera enfin écrit dans un nouveau fichier sous le format .json*, un formatage aussi pratique à lire pour un être humain que pour une machine. Les 26

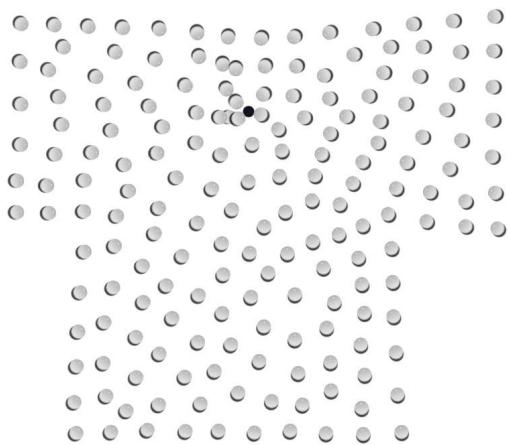
screenshots de l'application ROM sont toute fois nécessaire.

Voilà donc notre premier lot de données (= dataset). Les quelques pages suivantes expliquent brièvement comment un script écrit dans l'environnement de la librairie P5.js m'a permis de générer d'autres datasets, très rapidement et précisément, en partant de fichiers typographiques déjà existants.

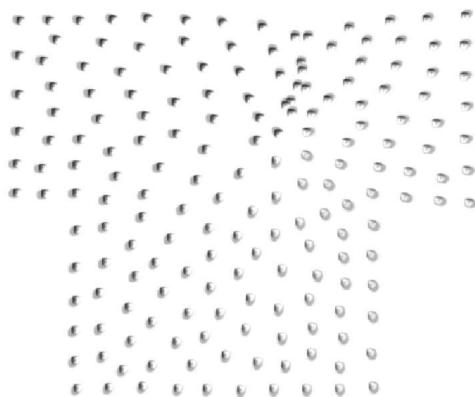


3

3. 26 screenshots de l'application CD-Rom, chargés sur sheepshaver.



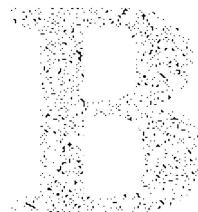
↑ Screenshot du nouveau programme
↓ Screenshot du programme originale



Suite avec P5.js

Comment générer des datasets de font en nuages de points !

P5.js est une librairie javascript conçu pour ce qu'on appelle le creative coding. Un peu comme director, la librairie utilise une métaphore pour son langage, celle du dessin car les fichiers javascript sont nommés « sketch », champ lexical hérité directement de Processing⁴.



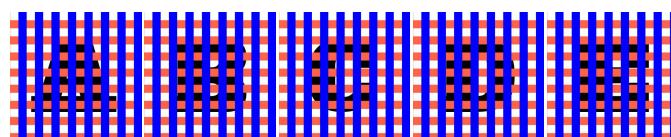
Une première idée est de passer du bruit d'une quelconque manière en noir et blanc, sans nuances de gris, sur une lettre comme le B ci-dessus. Mais le test présente des problèmes de densité dans certaines zones, peu optimal pour le rendu que nous souhaitons avoir.

Une seconde idée est de grillager des lettres noires, sur un fond blanc avec une grille paramétrable, et blanche. L'épaisseur des traits et

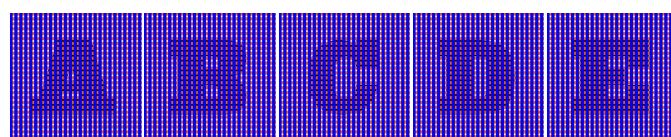
4. Processing est un environnement de développement intégré (IDE) et un langage de programmation créé en 2001 par Casey Reas et Ben Fry. Il est conçu pour simplifier la programmation des arts visuels, de la conception graphique, de l'animation et des interactions

leur espacement déterminent le nombre de points et leur taille.

Ci-dessous sont présents 3 séries d'images directement générées par le script détaillant des étapes de travail.



Grille aux traits larges et espacés, avec une couleur unique pour les lignes horizontales et verticales.



Grille aux traits fins et proches, mêmes couleurs qu'au-dessus.



Résultat avec une grille blanche et les mêmes épaisseurs que sur la seconde série.

Et une utilisation adaptée de fonctions P5.js et native à javascript permet d'obtenir des dossiers remplis d'images de ces fonts en pointillés.



Exemple du traitement appliqué à la font Pilowlava.

C'est cette piste qui conduira directement à penser une nouvelle application qui regroupe tous ce que nous venons de voir, nommé ColorKineType, votre prochaine lecture.

Webographie

Sprite (computer graphics), décembre 2022, depuis
wikipedia,
[https://en.wikipedia.org/wiki/Sprite_\(computer_graphics\)](https://en.wikipedia.org/wiki/Sprite_(computer_graphics))

Villar Toin, « **Virtualisation vs émulation : quelle est la différence ?** », *savoirdanslavie*, janvier 2022, [En ligne],
publié le 31 JANVIER 2022,
<https://www.savoirdanslavie.com/virtualization-vs-emulation/>

