

# Les fondements de la mémoire informatique

## Memory Basics

Carl Therrien John Aycock  
Cindy Poremba

Éditorialisation/content curation  
Maxime Deslongchamps

Traduction/translation  
Hélène Buzelin

**Référence bibliographique/bibliographic reference**  
Therrien, Carl, John Aycock et Cindy Poremba. *La remédiation des images de cinéma dans le jeu vidéo / Remediation of Cinema Images in Videogames*. Montréal : CinéMédias, 2023, collection « Encyclopédie raisonnée des techniques du cinéma », sous la direction d'André Gaudreault, Laurent Le Forestier et Gilles Mouëllic.

**Dépôt légal/legal deposit**  
Bibliothèque et Archives nationales du Québec,  
Bibliothèque et Archives Canada/Library and Archives Canada, 2023  
ISBN 978-2-925376-06-4 (PDF)

**Appui financier du CRSH/SSHRC support**  
Ce projet s'appuie sur des recherches financées par le  
Conseil de recherches en sciences humaines du Canada.

This project draws on research supported by the  
Social Sciences and Humanities Research Council of Canada.

**Mention de droits pour les textes/copyright for texts**  
© CinéMédias, 2023. Certains droits réservés/some rights reserved.  
Creative Commons Attribution-ShareAlike 4.0 International



**Image d'accroche/header image**  
Vue de la console PC Engine Duo-R avec le jeu *Snatcher*  
(Konami, 1988). [Voir la fiche](#).

PC Engine Duo-R video game console with *Snatcher*  
(Konami, 1988). [See database entry](#).

**Base de données TECHNÈS/TECHNÈS database**  
Une base de données documentaire recensant tous les contenus  
de l'*Encyclopédie* est en [libre accès](#). Des renvois vers la base sont  
également indiqués pour chaque image intégrée à ce livre.  
A documentary database listing all the contents of the *Encyclopedia*  
is in [open access](#). References to the database are also provided for  
each image included in this book.

**Version web/web version**  
Cet ouvrage a été initialement publié en 2020 sous la forme  
d'un [parcours thématique](#) de l'*Encyclopédie raisonnée des  
techniques du cinéma*.

This work was initially published in 2020 as a [thematic parcours](#)  
of the *Encyclopedia of Film Techniques and Technologies*.

# Les fondements de la mémoire informatique

par Carl Therrien, John Aycock et Cindy Poremba

Traduction : Hélène Buzelin

Par commodité, on a enfoui la dimension matérielle des technologies numériques dans des boîtes noires qui, au fil du temps, sont devenues toujours plus petites. Si certains aspects de ces technologies sont donc intangibles pour l'utilisateur final, il n'en demeure pas moins que l'information, dans un système numérique, repose sur des supports matériels dont les propriétés déterminent dans une large mesure la capacité de ce système à restituer des images photographiques et cinématographiques. Cette section rappelle quelques notions essentielles pour comprendre le formidable défi que constitue l'usage d'un courant électrique pour permettre à un système informatique de représenter, de stocker et de traiter des informations complexes comme des images cinématographiques.

Tous les appareils numériques obéissent à un principe binaire : dans certaines conditions, le courant passe – ou non. Les composants électroniques ont toujours été pensés et remodelés de façon à exploiter le potentiel qu'offre cette binarité. La plupart des transistors fonctionnent, ainsi, comme des interrupteurs. Si les ingénieurs des années 1940 pouvaient manipuler chaque tube électronique à la main, aujourd'hui les fabricants sont en mesure d'imprimer des milliards de transistors de taille nanométrique sur de minuscules circuits de silicium afin de produire toutes sortes de processeurs qui exécutent des millions de tâches par seconde. Les données traitées par ces processeurs sont elles aussi représentées et stockées en mode binaire. Certaines applications comme la conception assistée par ordinateur requièrent énormément de puissance de calcul (on dit alors qu'elles sont « *process intensive* »), tandis que d'autres exigent du système qu'il stocke, recherche et traite un volume considérable de données (on dit de ces applications qu'elles sont « *data intensive* »)<sup>[1]</sup>. Si la création de jeux vidéo inspirés du cinéma demande à la fois une puissance de calcul conséquente et de grandes capacités de stockage, le premier enjeu a toujours été de réussir à emmagasiner des images photographiques dans les mémoires des appareils électroniques destinés aux particuliers.

Le codage binaire d'informations complexes se fait par juxtaposition de bits. Très tôt dans l'histoire de l'informatique, les ordinateurs ont dû intégrer des codes alphanumériques pour que l'écriture sur clavier puisse servir d'unité d'entrée. Par exemple, quatre bits d'information peuvent générer 16 valeurs différentes ( $2^4$ ), ce qui est inférieur au nombre de caractères de l'alphabet latin. En revanche, avec une chaîne de huit bits (ce qu'on appelle un octet, l'unité de base de la mémoire informatique), on peut représenter 256 valeurs ( $2^8$ ), ce qui est suffisant pour encoder tous les symboles d'un clavier alphanumérique ordinaire.

Le graphisme des jeux vidéo permet de visualiser les possibilités et les limites de cette architecture. La situation est assez simple dans le cas d'une image en noir et blanc : les unités phosphorescentes et luminescentes à l'écran, les fameux « pixels », peuvent être contrôlées individuellement selon une logique binaire d'interrupteur (allumé/éteint) semblable à celle décrite ci-dessus. Imaginons qu'un développeur souhaite représenter un *space invader* en utilisant huit pixels pour la hauteur et autant pour la largeur. Comme chaque octet intègre huit bits, il aurait besoin de huit octets pour générer la figure ci-dessous.

Le paysage technologique devient infiniment plus complexe si on ajoute la couleur. Dans un scénario hypothétique où le dessin utiliserait une palette de 256 couleurs, il faudrait définir un code précis de huit bits pour désigner chacune d'entre elles. Dans cet exemple un peu naïf (et beaucoup plus simple que celui qui a présidé à l'intégration de la couleur dans les jeux vidéo), chacun des 64 pixels de la figure ci-contre correspondrait à un octet. Comprenons bien que, même dans un jeu relativement simple comme *Space Invaders* (Taito, 1978), la création de scènes interactives exige la saisie et la manipulation d'une multitude d'éléments stockés dans le programme informatique : des images matricielles (communément appelées *bitmap* en anglais) décrivant une panoplie de personnages et d'états, des sons et le programme lui-même. La capacité de la mémoire varie beaucoup d'une plateforme à l'autre et elle a sans cesse évolué dans le temps, rythmant l'histoire du jeu vidéo. En pratique, notre *space invader* de 64 octets n'aurait pas pu être intégré, de manière fonctionnelle, à la console Atari VCS et à ses 128 octets de mémoire vive. En revanche, aujourd'hui, il entrerait sans problème dans n'importe quel téléphone intelligent. Le coût des composants de la mémoire électronique dans les années 1970 est en grande partie responsable des restrictions chromatiques typiques des jeux de cette époque. Les pixels verts de la première version pour arcade du jeu de Taito étaient créés par superposition d'une bande de gel collée sur le moniteur pour donner de la couleur à certains éléments précis.

0	0	1	1	1	1	0	0
0	1	1	1	1	1	1	0
0	1	0	1	1	0	1	0
1	1	1	1	1	1	1	1
1	0	1	0	0	1	0	1
0	1	0	1	1	0	1	0
0	1	0	1	1	0	1	0
0	1	0	0	0	0	1	0

Affichage monochrome d'un extraterrestre de *Space Invaders*. [Voir la fiche](#).



Un extrait vidéo est accessible [en ligne](#).

Afin d'ajouter des variations de couleur aux affichages monochromes, certaines consoles et machines d'arcade recouvraient l'écran avec des filtres de plastiques colorés transparents. [Voir la fiche](#).

À partir de cette explication schématique, on peut imaginer les capacités de stockage et de mémoire nécessaires pour transposer, en codage binaire, la norme de 24 images par seconde dans une définition équivalente à des millions de pixels, où chaque pixel serait associé à des millions de couleurs. Par exemple, selon les normes DCI (Digital Cinema Initiatives), les films en 4K ont une définition de 4096 × 2160 pixels à 24 images par seconde, chaque image pouvant occuper jusqu'à 1,3 million d'octets<sup>[2]</sup>. Pour un film entier, cela demeure un défi en matière de transmission et de stockage des données. En plus des limites de la mémoire vive (RAM), le stockage de tous les clips vidéo d'un film dans les puces mémoires (ROM) reste également complexe, si l'on considère que le système vidéoludique doit représenter différents états en fonction des choix et des actions du joueur. Dans ce contexte, il est étonnant que les développeurs aient ne serait-ce qu'essayé d'intégrer, dans leurs jeux, des images d'apparence photographique ou cinématographique. Il est plus fascinant encore qu'ils aient cherché à le faire alors que ce nouveau média n'en était qu'à ses balbutiements.

---

[1] Cette distinction utile est tirée de Noah Wardrip-Fruin, *Expressive Processing: Digital Fictions, Computer Games, and Software Studies* (Cambridge, Massachusetts: MIT Press, 2009).

[2] Digital Cinema Initiatives, « DCI Specification, Version 1.4 », juillet 2020, [https://dcimovies.com/archives/spec\\_v1\\_4/DCI\\_DCSS\\_v1-4\\_20-July-2020.pdf](https://dcimovies.com/archives/spec_v1_4/DCI_DCSS_v1-4_20-July-2020.pdf).

# Memory Basics

by Carl Therrien, John Aycock and Cindy Poremba

The material aspect of digital technologies has been buried into convenient and increasingly smaller black boxes. While some aspects of media have been made intangible for the end user, information in a digital system still relies on very specific material boundaries that have significantly impacted the ability of computer systems to remediate photographic and cinematic images. This section introduces some elementary electronic notions which underline the formidable challenge of harnessing electrical current in order to represent, store and manipulate complex information such as cinema images in computer systems.

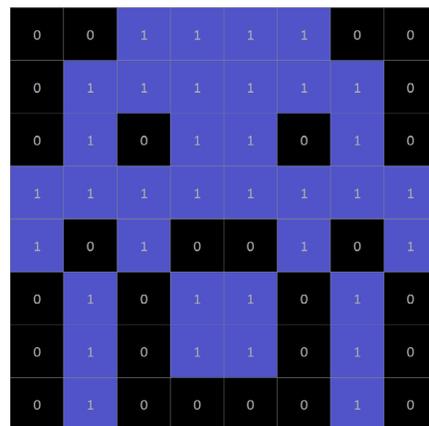
All digital tools rely on a binary reality: under certain conditions, electrical current either flows or it doesn't. Electronic components were created and constantly refined in order to exploit the potential of the binary condition. Many types of transistors have integrated this fundamental switch ability; while engineers in the 1940s could manipulate each tube transistor by hand, contemporary manufacturers can print billions of transistors at the nanometer level on a tiny silicon-based circuit in order to produce versatile processors, running through millions of instructions per second. The data manipulated by these computer processors also relies on binary representation and storage. Some computer applications such as computer-assisted design necessitate tremendous computational power (they are "process intensive") while others rely on the system's ability to store, fetch and manipulate incredible quantities of information (they are "data intensive").<sup>[1]</sup> While the creation of cinema-inspired videogames involved both aspects, transposing photographic images in a way that could fit memory structures in consumer-grade equipment certainly represented the biggest challenge.

The accumulation of bits in memory components is essential to encode complex realities in binary representations. An early necessity in computer design was the integration of alphanumeric codes in order to integrate typing as an input device. For instance, 4 bits of information led to a maximum of 16 values ( $2^4$ ), which is insufficient to point towards all the elements in the Roman alphabet. With a chain of 8 bits (which is known as a "byte" or "half word", one of the most fundamental units of memory in computer systems), one can represent 256 values ( $2^8$ ), which is sufficient to encode all the symbols in a typical alphanumeric keyboard.

The graphical component of videogames can help us visualize the affordances and limitations of this architecture. The situation is relatively simple for a black and white display: phosphorescent or luminescent units on a screen (the famous "pixels") can be controlled individually with a switching principle (on/off) similar to the one described above. Imagine a game creator wants to represent a space invader on the screen, using 8 pixels for both height and width.

Considering one byte integrates 8 individual bits, one would require 8 bytes in order to bring the next figure to the display.

This technological portrait is made infinitely more complex with the addition of color. In a hypothetical scenario where our space invader would be drawn from a 256-color palette, a specific 8-bit code is required to point to each of the colors in this palette. In this naive example (which is more simple than the actual early implementations of color in videogames), each of the 64 pixels in the bitmap represented in the next figure would require one byte. Now, it is essential to understand that even a relatively simple game such as, for instance, *Space Invaders* (Taito, 1978), require multiple elements to be fetched and manipulated in working memory in order to create interactive scenes, such as bitmaps that depict a variety of characters and states, sounds, and the program itself. Available memory varies greatly from one platform to the next, and has been evolving constantly in videogame history; while our 64-byte space invader cannot be integrated in any functional manner on the Atari VCS and its 128 bytes of RAM, it is now trivial to implement on any smartphone. The cost of memory components in the 1970s explains, among many other design aspects, the chromatic limitations seen in videogames; the green pixels seen in the first version of Taito's game in the arcade are actually created through an overlay on specific parts of a monochrome display.



Monochrome display of an alien from *Space Invaders*.

[See database entry.](#)

Through this simplified explanation of technological affordances, one can begin to imagine the amount of storage and working memory required to transpose in a binary format the cinematic standard of 24 images per second, with a screen resolution equivalent to millions of pixels, where each of these pixels can be defined through millions of different hues. For instance, 4K movies as defined by the DCI digital cinema specification have a resolution of 4096 × 2160 pixels at 24 images (frames) per second, and each frame can be comprised of up to 1.3 million bytes.<sup>[2]</sup> For a full movie, this presents challenges for data transmission and storage even today. Beyond RAM



A video clip is available [online](#).

To add color variations to monochrome displays, some consoles and arcade machines had their screen covered by a transparent colored overlay.

[See database entry.](#)

limitations, storing all the movie clips in memory chips (or ROMs) can represent a challenge up to this day, considering that a ludic system needs to depict state variations in order to reflect user selections or performance. In this context, it is fascinating that game developers tried to integrate photographic or movie-like images in videogames at a very early stage in the medium's history.

.....  
[1] Noah Wardrip-Fruin, *Expressive Processing: Digital Fictions, Computer Games, and Software Studies* (Cambridge, Massachusetts: MIT Press, 2009).

[2] Digital Cinema Initiatives, "DCI Specification, Version 1.4," July 2020, [https://dcimovies.com/archives/spec\\_v1\\_4/DCI\\_DCSS\\_v1-4\\_20-July-2020.pdf](https://dcimovies.com/archives/spec_v1_4/DCI_DCSS_v1-4_20-July-2020.pdf).