

VisiCalc

Le tableur qui révolutionna l'informatique personnelle

Stéphane FOSSE

fosse.fr

17 février 2026

Copyright : cette œuvre est libre, vous pouvez la copier, la diffuser et la modifier
selon les termes de la [Licence Art Libre](#)

En 1978, deux diplômés du MIT créèrent VisiCalc sur un Apple II. Cette application bouleversa l'industrie informatique en transformant le micro-ordinateur en outil professionnel indispensable, ouvrant la voie à des décennies de domination du tableur dans les entreprises.

La genèse : d'une idée d'étudiant à un prototype fonctionnel

En 1978, Daniel Bricklin, étudiant à Harvard Business School, imagina une solution élégante à un problème récurrent. Lors de la préparation de ses études de cas, il devait recalculer manuellement d'interminables colonnes de chiffres chaque fois qu'une hypothèse changeait. Son expérience antérieure chez Digital Equipment Corporation, où il avait travaillé sur des systèmes de traitement de texte et de composition typographique, lui avait appris l'importance de la réactivité immédiate de l'interface utilisateur.

Bricklin conçut alors une « calculatrice visible » – un tableau interactif où les formules, les calculs et les résultats coexisteraient sur un même écran. Contrairement aux programmes FORTRAN ou Basic de l'époque qui séparaient strictement l'entrée des données, le traitement et la mise en forme de la sortie, VisiCalc fusionnait ces trois plans en une seule interface cohérente.

Au printemps 1978, Bricklin développa un premier prototype en Integer Basic sur le PDP-10 de Harvard. Cette version rudimentaire validait le concept fondamental : une grille où chaque cellule pouvait contenir soit une valeur, soit une formule référant d'autres cellules. Le nommage des cellules par coordonnées (colonnes alphabétiques, lignes numériques) résolvait élégamment le problème de référencement, bien que Bricklin ait initialement envisagé un placement libre des données à l'écran.

L'architecture technique : des choix contraints qui forgèrent l'excellence

Bob Frankston, camarade de Bricklin sur le projet Multics au MIT, rejoignit le développement fin 1978. Les deux hommes formaient un duo complémentaire : Bricklin définissait l'expérience utilisateur et rédigeait une documentation qui servait de spécification, tandis que Frankston maîtrisait les aspects systèmes et architecturaux les plus pointus.

L'environnement de développement illustre parfaitement les contraintes de l'époque. Incapable d'obtenir rapidement une machine DEC PDT, Bricklin se tourna vers l'Apple II récemment équipé de lecteurs de disquettes. Mais l'Apple II ne disposait d'aucun outil de développement sérieux. La solution ? Emprunter un mini-ordinateur Prime 350 via un prêt personnel garanti, puis développer sur Multics au MIT un assembleur croisé pour le processeur 6502 de l'Apple.

Cette approche, standard à l'époque mais souvent oubliée aujourd'hui, permettait d'exploiter la puissance d'un système partagé pour générer du code destiné au micro-ordinateur. Le code assembleur était écrit sur Multics à 120 caractères par seconde, compilé, puis téléchargé en hexadécimal vers l'Apple II via un programme terminal spécialement conçu qui convertissait le flux en binaire exécutable.

Les contraintes mémoire imposèrent des choix d'architecture ingénieux. VisiCalc devait tenir dans 16 kilo-octets de mémoire – objectif finalement raté puisque la version finale nécessitait 32 Ko. Chaque octet comptait. Le texte des invites à l'écran servait également de valeur de test lors de l'appui sur une touche, économisant ainsi de précieux octets. Le format de sauvegarde consistait simplement en la séquence de touches nécessaire pour recréer la feuille, permettant de facto l'implémentation future des macros sans code supplémentaire.

Frankston imposa une règle cardinale : si une fonctionnalité ne pouvait être expliquée clairement à l'utilisateur, il fallait revoir le programme. La documentation rédigée par Bricklin servait de test de simplicité. Cette discipline

força des choix comme l'étiquetage fixe des lignes et colonnes – initialement modifiables, cette fonctionnalité fut sacrifiée pour garantir un support téléphonique cohérent.

L'équipe opta pour l'arithmétique décimale plutôt que binaire, avec une précision de 12 chiffres. Cette décision visait à reproduire les erreurs d'arrondi qu'un comptable ferait manuellement, évitant les surprises du type « 0,999... » issues de l'arithmétique binaire. Les micro-ordinateurs de 1979 manipulaient naturellement des nombres décimaux via leurs calculatrices ; VisiCalc devait parler le même langage.

Le modèle éditeur-développeur : une alliance initiale fructueuse

En septembre 1978, Dan Fylstra, fondateur de Personal Software, vit le prototype de Bricklin. Fylstra, diplômé du MIT et de Harvard, éditait déjà MicroChess. Il comprit immédiatement le potentiel commercial et proposa de publier VisiCalc. Le 2 janvier 1979, Bricklin et Frankston fondèrent Software Arts Corporation. Le nom, délibérément choisi avec le mot « Arts », soulignait que la programmation n'était pas qu'une discipline technique mais également un art requérant esthétique et élégance.

Le contrat établissait une division nette : Personal Software gérait la commercialisation, la relation avec les constructeurs et la fabrication des disquettes, tandis que Software Arts conservait la propriété du code source, les droits d'auteur et la responsabilité du développement. Cette séparation fonctionnait initialement, Personal Software fournissant des avances sur royalties non remboursables qui permirent à Bricklin et Frankston d'acheter leur Prime 350.

Mitch Kapor rejoignit Personal Software comme chef de produit en mars 1980. Kapor avait déjà vendu VisiTrend/VisiPlot à Personal Software pour un million de dollars. Sa connaissance approfondie de VisiCalc – il en fut testeur bêta – et sa compréhension des besoins métier allaient s'avérer déterminantes, mais dans un sens que personne n'anticipait alors.

Le lancement et l'adoption : quand un logiciel devient *killer app*

VisiCalc fut présenté publiquement en juin 1979 à la National Computer Conference de New York, dans les sous-sols du Sheraton, espace réservé aux micro-ordinateurs que l'industrie des gros systèmes considérait encore comme le parent pauvre. La version 1.37, lancée en octobre 1979 au prix de 99 dollars, se vendit à 1 293 exemplaires en octobre et 4 258 en décembre.

Le logiciel transforma l'Apple II en machine professionnelle. Steve Wozniak, cofondateur d'Apple, attribua le succès de son entreprise à deux facteurs : le lecteur de disquettes et VisiCalc. Benjamin Rosen, analyste influent, prophétisa : « VisiCalc pourrait devenir la queue logicielle qui remue le chien informatique personnel ». Sa prédiction se réalisa : environ 90% des Apple II étaient vendus à de petites organisations, non pour un usage domestique comme prévu, mais pour exploiter VisiCalc.

L'adoption suivit un modèle d'évangélisation plutôt que de marketing classique. Les comptables et analystes financiers qui voyaient VisiCalc reproduire leur application manuscrite « se mettaient à trembler d'excitation ». Les informaticiens, au contraire, ne voyaient qu'un programme qu'ils pourraient écrire en Basic. Cette dichotomie révélait la vraie nature de VisiCalc : non pas une prouesse technique pour techniciens, mais un outil métier pour professionnels.

Le succès de VisiCalc reposait sur trois piliers identifiés par Richard Rumelt : il permettait aux non-programmeurs de bâtir des modèles quantitatifs sur ordinateur ; il offrait une rétroaction instantanée sur les modifications (le fameux *what-if*) ; c'était une application générique, rare à l'époque des programmes sur mesure.

La bataille des tableurs : de VisiCalc à Lotus 1-2-3

En janvier 1983, Mitch Kapor lança Lotus 1-2-3. Le logiciel, développé avec Jonathan Sachs, intégrait tableur, graphiques et fonctionnalités de base de données. Plus significativement, Kapor avait optimisé 1-2-3 exclusivement pour l'IBM PC, apparu en 1981. Tandis que Software Arts maintenait VisiCalc sur multiples plateformes (Apple II, Apple III, Atari, TRS-80, HP, IBM PC), Lotus concentrait toutes ses ressources sur l'écosystème IBM.

Kapor connaissait intimement VisiCalc : testeur bêta de VisiCalc Advanced Version, ancien chef de produit chez Personal Software, il identifia précisément les irritants à corriger. Lotus 1-2-3 offrait des largeurs de colonnes variables, des virgules dans les nombres, de longues étiquettes – autant d'améliorations de présentation qui permettaient d'utiliser directement la sortie dans un rapport. VisiCalc nécessitait souvent une recopie manuelle pour un rendu professionnel.

La rapidité d'exécution constituait un autre avantage décisif. Écrit entièrement en assembleur 8088 par Jonathan Sachs, 1-2-3 calculait à la vitesse de la touche de répétition, exactement comme l'original VisiCalc sur Apple II. Frankston confirme : « Mitch m'a dit : Dan, je me souviens comment tu nous as expliqué que Frankston

l'a rendu aussi rapide que la touche de répétition. Nous avons travaillé dur pour atteindre cette vitesse, c'est pourquoi c'est en assembleur. »

Les chiffres parlent d'eux-mêmes. En 1983, première année d'exploitation, Lotus Development réalisa 53 millions de dollars de revenus. L'année suivante : 156 millions. À l'automne 1983, les ventes mensuelles de 1-2-3 dépassèrent celles de VisiCalc. En 1985, Lotus Development rachetait Software Arts et abandonnait la marque VisiCalc.

Les leçons architecturales : pourquoi certains choix scellent le destin

Plusieurs décisions architecturales expliquent le déclin de VisiCalc face à 1-2-3. Software Arts avait opté pour une stratégie multi-plateformes, nécessitant de porter le code sur chaque nouveau système. Pour Atari 800, HP 125, TRS-80 Model I, II et III, IBM PC, chaque version réclamait un effort substantiel. Seth Steinberg développa une méthode ingénieuse de « traduction idiomatique » : le compilateur listait côte à côte le code 6502 original et sa traduction Z80, avec les commentaires synchronisés. Cette approche permit de maintenir la cohérence entre versions, mais mobilisait des ressources considérables.

Face à cette complexité, Software Arts créa « IL » (Implementation Language), un langage de plus haut niveau dérivé de Lisp. VisiCalc Advanced Version, lancé sur Apple III avec largeur de colonnes variable, macros et feuilles multiples, fut écrit en IL. Cette approche devait faciliter le portage, comme Microsoft le faisait avec Multiplan. Malheureusement, le code interprété d'IL s'avéra significativement plus lent que l'assembleur natif de 1-2-3.

Lotus, à l'inverse, paria sur une seule plateforme. Cette concentration des ressources permit à Sachs de peaufiner chaque détail de performance. Pendant des années, personne ne parvint à surpasser la qualité technique du code de 1-2-3 sur IBM PC avec 640 Ko de mémoire. Ce n'est qu'avec l'arrivée d'Excel sur Macintosh, puis Windows, qu'une alternative crédible émergea.

La relation entre Software Arts et Personal Software, rebaptisée VisiCorp en 1982, se détériora progressivement. Les discussions de fusion au printemps 1980 échouèrent. VisiCorp, sous la direction de Terry Opendyck, réduisit les royalties et investit massivement dans VisiOn, un environnement graphique concurrent de Windows. Les budgets marketing de VisiCalc s'effondrèrent au moment précis où Lotus dépensait des millions. En septembre 1983, VisiCorp assigna Software Arts en justice pour 50 millions de dollars, deux jours avant la vente prévue de Software Arts à CompuServe pour le même montant.

Le procès dura jusqu'en 1984. Software Arts contre-attaqua en invoquant la clause de « best efforts » (meilleurs efforts) du contrat de publication : VisiCorp ne commercialisait plus activement VisiCalc, concentré sur VisiOn. Le tribunal donna raison à Software Arts et leur restitua la marque VisiCalc. Mais la victoire judiciaire arriva trop tard. Lotus 1-2-3 dominait désormais le marché corporatif.

L'héritage : quand l'innovateur ne capture pas la valeur

VisiCalc illustre un paradoxe classique de l'innovation : les pionniers ouvrent la voie, les suiveurs récoltent. Bricklin et Frankston choisirent délibérément de ne pas breveter VisiCalc, par idéalisme et parce que les brevets logiciels n'existaient pratiquement pas en 1979. Un avocat consulté estima les chances d'obtenir un brevet « minces » et coûteuses. Cette décision permit à une industrie entière de se développer, mais priva les créateurs de leur rente.

Certains choix techniques de VisiCalc demeurent remarquablement durables. Le défilement synchronisé avec titres figés, les fenêtres multiples de tailles différentes, l'affichage immédiat des modifications – Excel, quarante ans plus tard, peine à égaler ces fonctionnalités. La séparation conceptuelle entre présentation et données, héritée de l'expérience Multics de Frankston, anticipe l'architecture moderne des applications.

Le modèle de la « feuille de calcul » (*spreadsheet*) s'imposa comme paradigme universel. Lotus 1-2-3 puis Excel ne firent qu'affiner l'intuition originale : fusionner les trois plans (données, calculs, présentation) dans une grille interactive où l'utilisateur pense directement en termes de résultats. Cette approche permit à des millions de personnes de devenir « programmeurs » sans écrire une ligne de code.

VisiCalc vendit approximativement 500 000 exemplaires durant son existence commerciale. Lotus 1-2-3 dépassa les 100 millions de dollars de ventes dès sa première année complète d'exploitation. Excel domine aujourd'hui avec plus d'un milliard d'utilisateurs selon Microsoft. Pourtant, tous descendent directement du prototype développé en Basic par Bricklin sur le PDP-10 de Harvard au printemps 1978.

En 1985, IBM racheta Lotus Development. En 1995, Microsoft Excel supplanta définitivement Lotus 1-2-3 comme tableur de référence. Mitch Kapor, ironie de l'histoire, sauva Software Arts de la faillite en 1985, achetant ses actifs en cinq jours pour éviter sa disparition. Bricklin fonda ensuite Software Garden, puis Trellix Corporation pour le développement web. Frankston rejoignit Lotus, puis Microsoft, où il contribua notamment aux réseaux domestiques.

Daniel Bricklin reçut le Grace Hopper Award de l'ACM en 1981, reconnaissant sa contribution exceptionnelle. L'IEEE et le MIT lui décernèrent également des distinctions. Ces honneurs, bien que mérités, ne reflètent qu'imparfaitement l'ampleur de sa contribution : VisiCalc ne fut pas simplement un logiciel à succès, mais la preuve de concept que le micro-ordinateur pouvait servir les entreprises et transformer le travail de millions de professionnels.

L'architecture de VisiCalc, forgée dans les contraintes extrêmes d'un Apple II à 32 Ko de RAM, offre des leçons toujours pertinentes. La rigueur dans la conception de l'interface utilisateur, l'obsession de la performance, la discipline de documentation comme spécification, le courage de supprimer les fonctionnalités trop complexes à expliquer – ces principes n'ont pas pris une ride. Bricklin et Frankston démontrèrent qu'avec des ressources limitées mais une vision claire, deux développeurs pouvaient révolutionner une industrie.

Références

- [1] Martin CAMPBELL-KELLY et Paul CERUZZI. [An Interview with Dan Bricklin and Bob Frankston](#). Charles Babbage Institute. Oral history collection, Charles Babbage Institute, University of Minnesota. Mai 2004.
- [2] Daniel J. POWER. [A History of Microcomputer Spreadsheets](#). In : *Communications of the Association for Information Systems* 4.1 (oct. 2000), p. 9.
- [3] Lazar RAKOVIĆ, Marton SAKAL et Veselin PAVLICEVIC. [Spreadsheets — How It Started](#). In : *International Scientific Journal of Management Information Systems* 9.4 (déc. 2014).
- [4] Jim RUE. [Dan Bricklin — The Father of Spreadsheets](#). In : *Technical Support Magazine* (nov. 2005).