

AIRWOLF : LES TONTONS CODEURS

Ou comment des branquignols ont flingué un hélicoptère

AU MENU

AVANT-PROPOS : UN MOT AVANT DE TRINQUER	5
PREMIERE PARTIE : LE CONTEXTE (OU COMMENT ON EN EST ARRIVÉS LÀ)	6
Chapitre 1 : Le pitch	7
Chapitre 2 : L'enquête	8
DEUXIEME PARTIE : L'ARCHITECTURE (OU COMMENT C'EST FOUTU LÀ-DEDANS)	10
Chapitre 3 : La carte du territoire	11
Chapitre 4 : Les couleurs de l'arnaque	13
Chapitre 5 : La mémoire vidéo (un vrai sac de nœuds)	14
TROISIEME PARTIE : L'HÉLICOPTÈRE (LE HÉROS MALGRÉ LUI)	16
Chapitre 6 : Portrait du bestiau	17
Chapitre 7 : La gravité (cette salope)	18
Chapitre 8 : Le tir (ta seule défense)	19
QUATRIEME PARTIE : LE SCROLLING (ÇA DÉFILE, MAIS PAS BIEN)	20
Chapitre 9 : Le choix de la facilité	21
CINQUIEME PARTIE : LES COLLISIONS (L'ARNAQUE ULTIME)	23
Chapitre 10 : Le système malin	24
Chapitre 11 : Le cyan, c'est la combine	25
Chapitre 12 : Le coup du Stack Pointer	26
SIXIEME PARTIE : LES ENNEMIS (LES FAIRE-VALOIR)	28
Chapitre 13 : La fiche signalétique	29
SEPTIEME PARTIE : LE SON (LA MUSIQUE DES EMMERDEMENTS)	31

Chapitre 14 : Le PSG	32
Chapitre 15 : L'explosion	33
HUITIÈME PARTIE : LA VÉRITÉ (LE POT AUX ROSES)	34
Chapitre 16 : Le niveau 2, parlons-en	35
Chapitre 17 : Le sabotage	35
Chapitre 18 : Les indices	37
Chapitre 19 : Plusieurs coupables	37
NEUVIÈME PARTIE : LA RÉSURRECTION	39
Chapitre 20 : Airwolf Reloaded	40
DIXIÈME PARTIE : LA PHILOSOPHIE DU BIDOUILLEUR	41
Chapitre 21 : Ce que cette histoire nous apprend	42
Chapitre 22 : Les leçons techniques	42
ANNEXES	43
Annexe A : La carte mémoire	43
Annexe B : Les codes de triche	43
Annexe C : Le best-of des routines	43
ÉPILOGUE : UN DERNIER VERRE ?	44
À propos de l'auteur	45



Quand on mettra les cons sur orbite, t'as pas fini de tourner. - Figure 1

AVANT-PROPOS : UN MOT AVANT DE TRINQUER

Y'a des questions qu'on ferait mieux de pas se poser. Genre : "Pourquoi le percepteur frappe toujours à ma porte ?" ou "C'est quoi le sens de la vie ?" Des questions comme ça, ça vous pourrit une existence.

Moi, une nuit où j'arrivais pas à pioncer, j'me suis posé celle-là : "Y'a quoi au niveau 2 d'Airwolf sur Amstrad CPC ?"

Une question conne, vous allez me dire. Et vous auriez raison. Sauf que les questions connes, c'est souvent celles qui vous mènent aux réponses les plus intéressantes. C'est comme les femmes compliquées : on sait qu'on va morfler, mais on peut pas s'empêcher.

Alors j'ai fouillé. J'ai gratté. J'ai désossé ce jeu comme un poulet du dimanche. Et ce que j'ai trouvé, mes amis, ça vaut le détour. C'est l'histoire d'un braquage qui a mal tourné - sauf que le braqué, c'est le joueur, et les braqueurs, c'est les développeurs.

Mais avant de commencer, un conseil : servez-vous un verre. Cette histoire, elle se lit mieux avec un petit jaune.

PREMIERE PARTIE : LE CONTEXTE (OU COMMENT ON EN EST ARRIVÉS LÀ)

Chapitre 1 : Le pitch

La série télé

Airwolf, à la base, c'est une série américaine des années 80. Un hélicoptère noir piloté par un type qui s'appelle Stringfellow Hawke - un nom pareil, faut le faire. Le gars, il bosse pour une agence secrète, il défouraille du méchant, et il a une gueule de premier de la classe qui aurait mal tourné.

La série, elle cartonnait. Alors forcément, les marchands du temple se sont dit : "Tiens, et si on en faisait un jeu vidéo ? Y'a du pognon à se faire."

L'adaptation

Elite Systems récupère les droits. Ils pondent des versions pour tous les ordis de l'époque : le Spectrum pour les fauchés, le Commodore pour les frimeurs, et l'Amstrad pour... bah, pour les gens normaux, quoi.



Le premier écran. Après, c'est plus la même limonade. - Figure 2

Le jeu sort en 1985. Le joueur, il pilote son hélico dans des grottes, il doit sauver des otages et tirer sur des trucs. Simple comme bonjour, sur le papier.

La réputation

Sauf que rapidement, le bouche-à-oreille fait son œuvre. Et c'est pas glorieux.

Airwolf, c'est pas un jeu difficile. C'est un jeu impossible. Le genre de truc où tu meurs quinze fois avant d'avoir compris les commandes. Les gamins dans les cours de récré, ils se passaient le mot : "Le niveau 2 d'Airwolf, personne l'a jamais vu."

C'était devenu une légende. Un mythe. Le Graal des jeux vidéo.

Quarante ans plus tard, j'allais découvrir pourquoi personne l'avait jamais vu. Et c'est pas joli-joli.

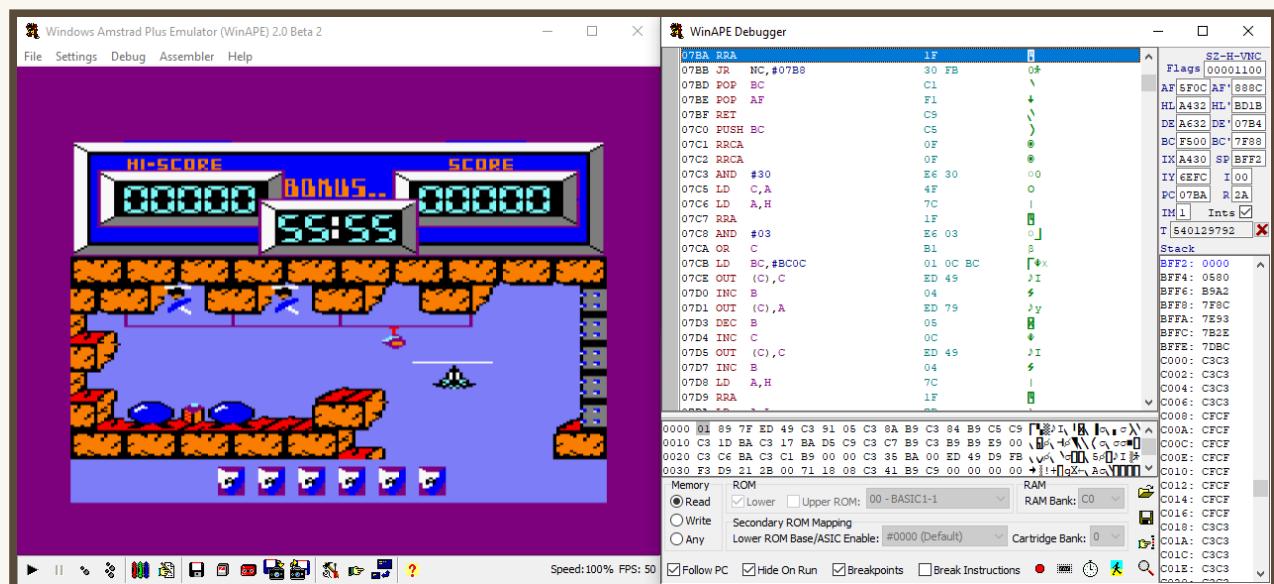
Chapitre 2 : L'enquête

Une nuit comme les autres

Faut que je vous explique le contexte. Cette enquête, elle a pas commencé dans un bureau avec des types en costard. Elle a commencé dans mon pieu, à trois heures du mat', avec une insomnie carabinée.

Quand tu dors pas, faut occuper le cerveau. Sinon il te fait des misères. Alors plutôt que de compter les moutons - activité de cave s'il en est - j'ai décidé de compter les octets.

Les outils du métier



Mon établi. C'est pas la Rolls, mais ça fait le taf. - Figure 3

Pour démonter un jeu, faut les bons outils. Moi, j'ai WinAPE. Un émulateur avec tout ce qu'il faut pour voir ce qui se passe sous le capot : - Un désassembleur (ça traduit le charabia machine en truc lisible) - Un visualiseur de mémoire (pour voir où ils planquent leurs magouilles) - Des points d'arrêt (pour stopper le film au bon moment)

Avec ça, même le code le plus retors, il finit par cracher le morceau.

Le premier os

Et le premier os, je l'ai trouvé assez vite. Le fameux "bug" de la zone finale ? C'est pas un bug. Regardez-moi ça :

```
CP #08          ; Zone 8, mon gars ?
.piege:
    JP Z, .piege      ; Alors tourne en rond jusqu'à la fin des temps
```

Quelqu'un - un être humain, avec des mains et un cerveau - a écrit ces trois lignes pour bloquer le jeu EXPRÈS.

C'est pas une erreur. C'est un assassinat prémedité.

DEUXIEME PARTIE :
L'ARCHITECTURE (OU COMMENT
C'EST FOUTU LÀ-DEDANS)

Chapitre 3 : La carte du territoire

64 Ko de bazar

L'Amstrad, il a 64 Ko de mémoire. C'est pas Byzance, mais c'est ce qu'on avait à l'époque. Airwolf, il utilise tout ça selon un plan bien précis :

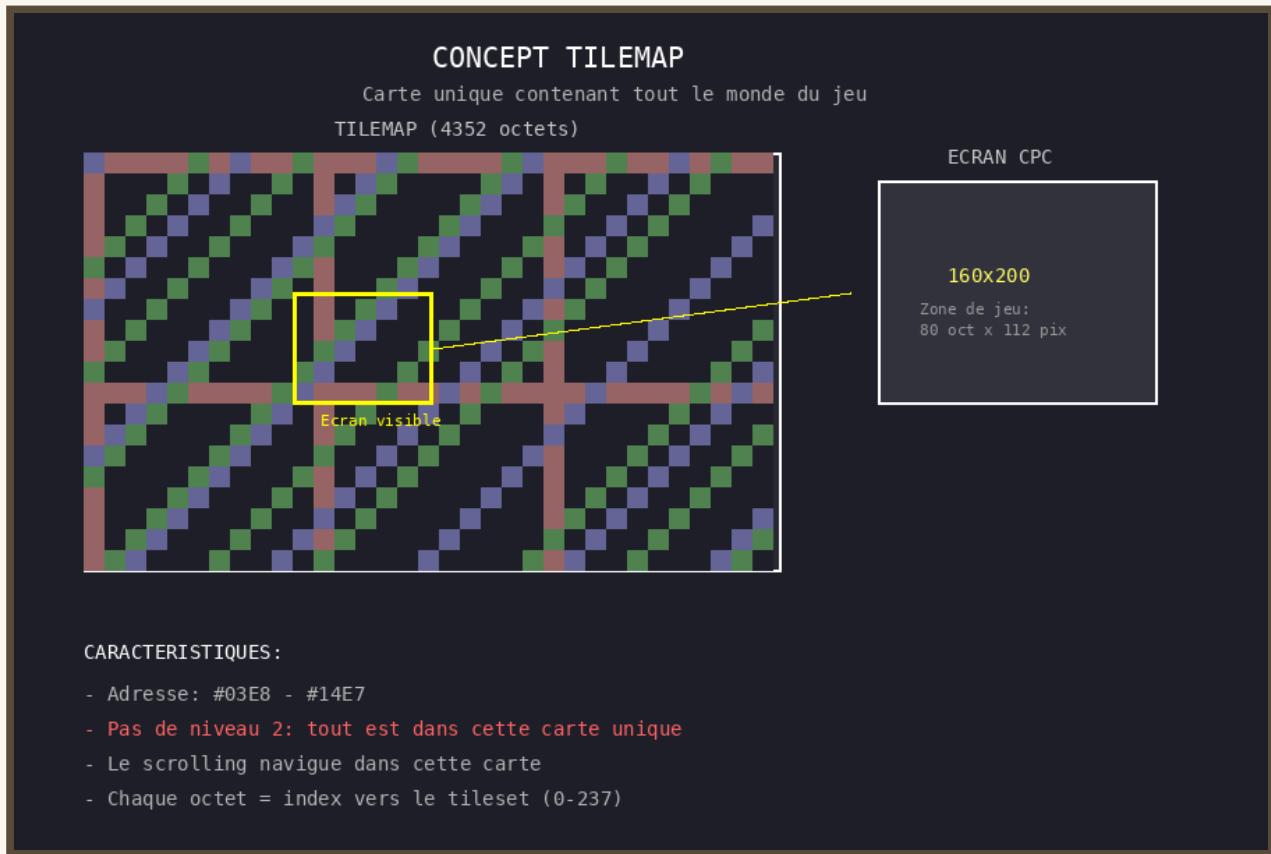


La carte au trésor. Sauf que le trésor, c'est de la camelote. - Figure 4

Adresse	Contenu	Commentaire du chef
#0000-#03E7	Variables système	Touche pas à ça, petit
#03E8-#14E7	LA CARTE DU MONDE	Tout le schmilblick est là
#14E8-#4F67	Les décors	238 morceaux de puzzle
#4F68-#4FFF	Les pales	Pour faire joli
#5000-#5FFF	Les explosions	Parce que tu vas mourir souvent
#6000-#64E7	Zone de travail	Le bac à sable
#64E8-#68A7	La musique	Pour accompagner ton malheur
#68A8-#80FF	LE CODE	C'est là que ça se passe
#C000-#FFFF	L'écran	Ce que tu vois

Le coup de la tilemap unique

Première surprise : tout le jeu tient dans une seule carte de 4352 octets. Y'a pas de chargement depuis la cassette, pas de niveau caché quelque part. Tout est là, d'entrée de jeu.

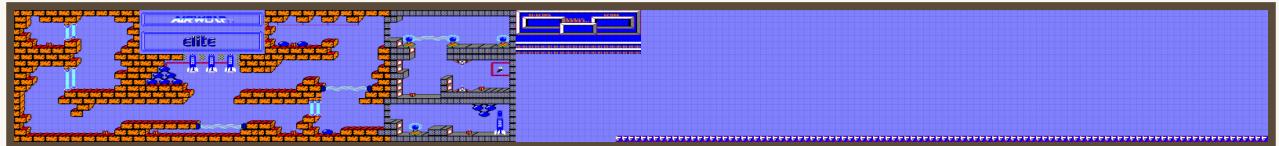


Une carte, et c'est marre. - Figure 5

Tu vois le problème ? Si tout le monde est là dès le départ, où il est, le niveau 2 ?

Patience. On y vient.

Vue d'ensemble



La totale. Tout ce que t'as à explorer. Enjoy. - Figure 6

La carte, elle se décompose comme ça : - **À gauche** : L'entrée, pour les petits joueurs - **Au milieu** : Le labyrinthe, pour les courageux - **À droite** : La zone 8, pour ceux qui y arrivent jamais

Cette zone 8. On y reviendra. C'est là que tout se joue.

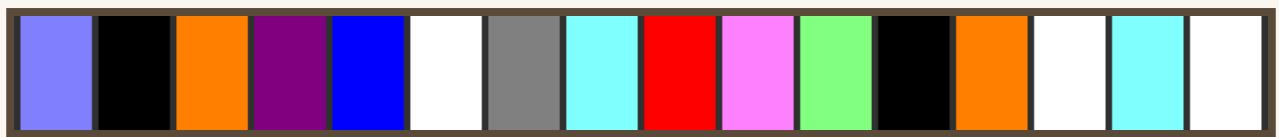
Chapitre 4 : Les couleurs de l'arnaque

Le Mode 0

L'Amstrad en Mode 0, c'est 16 couleurs et une résolution de 160x200 pixels. Pas de quoi pavoiser, mais suffisant pour les embrouilles.



27 couleurs au catalogue. Ils en prennent 16. - Figure 7



Celles qu'ils ont choisies. Y'a une entourloupe dedans. - Figure 8

Chaque couleur a son rôle

Dans ce jeu, les couleurs, c'est pas que pour faire joli. Chacune a une fonction :

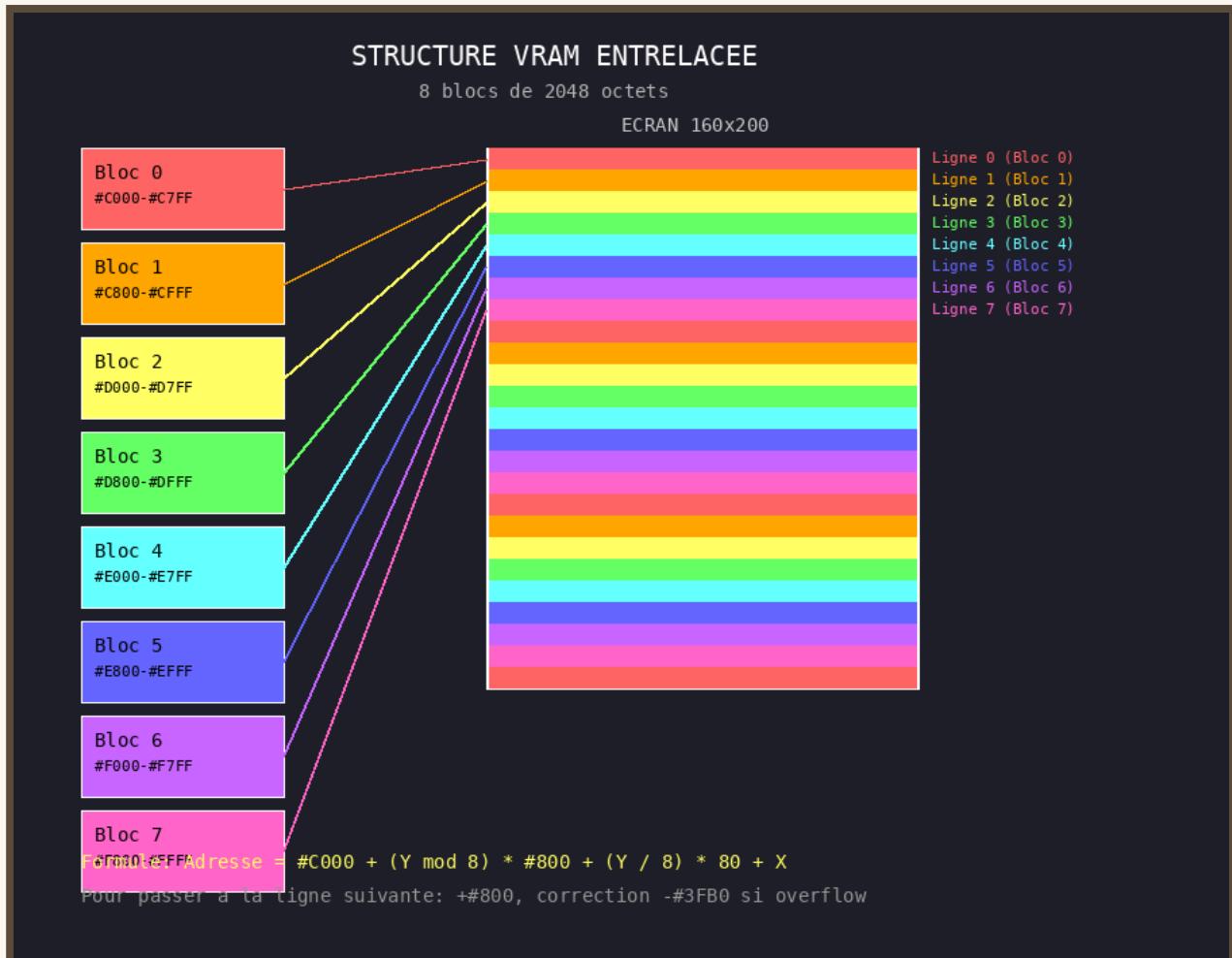
- **Bleu** (index 0) : Le vide. Là où tu peux voler peinard.
- **Noir** : L'ambiance. Le décor de fond.
- **Orange** : Les murs. Tu touches, tu crèves.
- **Cyan** : Les combines. Les boutons, les passages secrets.
- **Rouge** : Les emmerdeurs. Les ennemis.

Et y'a un truc qui cloche : deux oranges presque pareils. Pourquoi ? On en reparle au chapitre sur les arnaques.

Chapitre 5 : La mémoire vidéo (un vrai sac de noeuds)

L'entrelacement

La mémoire vidéo de l'Amstrad, c'est pas rangé comme dans un livre. C'est entrelacé comme un pull de mamie :



Un bordel organisé. Faut être tordu pour inventer ça. - Figure 9

```

Bloc 0 : lignes 0, 8, 16, 24...
Bloc 1 : lignes 1, 9, 17, 25...
Et ainsi de suite jusqu'au bloc 7

```

Pour afficher quoi que ce soit, faut jongler avec ce système. Un vrai casse-tête chinois.

**TROISIÈME PARTIE :
L'HÉLICOPTÈRE (LE HÉROS
MALGRÉ LUI)**

Chapitre 6 : Portrait du bestiau

SYSTEME HELICOPTERE AIRWOLF

7 FRAMES DE ROTATION:

Frame 0 DROITE Frame 1 Frame 2 Frame 3 HAUT Frame 4 Frame 5 Frame 6 GAUCHE

6 FRAMES PALES DU ROTOR:

F0 F1 F2 F3 F4 F5

DIMENSIONS:

- Corps: 16 octets x 20 lignes = 320 octets/frame
- Pales: 16 octets x 3 lignes = 48 octets/frame
- Total visible: 32 pixels x 23 pixels

VARIABLES PRINCIPALES:

<code>HELICO_X</code>	Position X (0-80)
<code>HELICO_Y</code>	Position Y (0-112)
<code>HELICO_DIR</code>	Direction (0=droite, 1=haut, 2=gauche)
<code>HELICO_ROTATION_PHASE</code>	Frame rotation actuelle (0-6)
<code>HELICO_ROTOR_FRAME</code>	Frame pales (0-5)
<code>GRAVITY_DOWN_TEMPO</code>	Tempo gravite (acceleration)
<code>HELICO_COLLISION_DETECTED</code>	Flag collision

GRAVITE: Tempo decrémente (10->4), acceleration progressive
Reset à 10 dès qu'un input vertical est détecté

Ton taxi pour l'enfer. - Figure 10

Les variables qui le gouvernent

L'hélico, il est défini par un paquet de variables entre #79C0 et #79F0 :

<code>HELICO_X:</code>	Où il est (horizontal)
<code>HELICO_Y:</code>	Où il est (vertical)
<code>HELICO_DIR:</code>	Où il regarde (0=droite, 1=haut, 2=gauche)
<code>GRAVITY_DOWN_TEMPO:</code>	La vitesse à laquelle il tombe

Le pauvre, il est soumis à la gravité, il peut mourir d'un rien, et en plus il doit sauver des otages. Une vie de chien.

La rotation



Sept positions. De profil droit à profil gauche. Classe. - Figure 11

L'hélico peut tourner dans sept positions différentes. Et le code qui gère ça, il est propre :

```

HELICO_PROCESS_ROTATION:
    LD A, (HELICO_DIR)      ; Où tu veux aller ?
    ADD A,A                 ; x2
    ADD A, (HELICO_DIR)     ; x3 -> Position cible

```

Ça, c'est du travail de pro. Pas le genre de code qu'un branquignol écrit.

Chapitre 7 : La gravité (cette salope)

Le système

Quand tu touches à rien, l'hélico, il tombe. Et plus tu attends, plus il tombe vite. C'est la physique, mon gars.

```

APPLY_GRAVITY:
LD A,(GRAVITY_DOWN_CPT)
DEC A           ; Compte à rebours
LD (GRAVITY_DOWN_CPT),A
JR NZ,.pas_encore

LD A,(GRAVITY_DOWN_TEMPO)
DEC A           ; Accélère la chute
LD (GRAVITY_DOWN_TEMPO),A

```

C'est vicieux. Tu lâches le manche deux secondes, et paf, tu t'écrases comme une crêpe.

L'observation du philosophe

Ce système de gravité, il est bien foutu. Trop bien foutu pour un jeu censément bâclé. Quelqu'un a passé du temps là-dessus. Du temps et du talent.

Alors pourquoi le reste est-il aussi bancal ? Mystère.

Chapitre 8 : Le tir (ta seule défense)

Comment ça marche

Tu peux tirer dans trois directions. La routine `FIRE_NEW_SHOOT` (#7898) gère tout ça :

```

FIRE_NEW_SHOOT:
LD A,(NB_SHOOT_FIRED)
CP #01
RET Z           ; Un seul tir à la fois, mon coco

LD A,#32         ; 50 images de vie pour le projectile
LD (SHOOT_LIFE_CPT),A

```

Un seul tir à la fois. Déjà que le jeu est dur, ils en rajoutent une couche.

QUATRIEME PARTIE : LE SCROLLING (ÇA DÉFILE, MAIS PAS BIEN)

Chapitre 9 : Le choix de la facilité

SYSTEME DE SCROLLING
Defilement par tiles (16 pixels)

BUFFERS:

SCROLLING_BUFFER (#6000)	1256 octets - Adresses source/destination
SCROLL_NEXT_SCR_BUF (#8100)	8960 octets - Ecran pre-calcule

PROCESSUS DE SCROLLING:

1. Pre-calculer le prochain ecran dans SCROLL_NEXT_SCR_BUF
2. Remplir SCROLLING_BUFFER avec les adresses VRAM
3. Pour chaque ligne (112 fois):
 - Charger adresses source/destination
 - Executer 76 LDI (copie 76 octets)
4. Copier la nouvelle colonne depuis le buffer pre-calcule

INSTRUCTION LDI (Load and Increment):

- Copie (HL) vers (DE) SCROLL_LINE_LEFT: 76 x LDI déroulées
- Incrémente HL et DE, décrémente BC SCROLL_LINE_RIGHT: 76 x LDD (sens inverse)

Le scrolling par blocs. Pas fluide pour un sou. - Figure 12

Par tiles, pas par pixels

Airwolf, il scrolle pas pixel par pixel comme les jeux de pros. Il scrolle par blocs de 16 pixels. Ça fait des à-coups, c'est moche, mais c'est plus simple à coder.

Le code fantôme

Sauf que - et c'est là que ça devient intéressant - y'a du code mort à l'adresse #7382. Du code de scrolling pixel par pixel. Jamais utilisé.

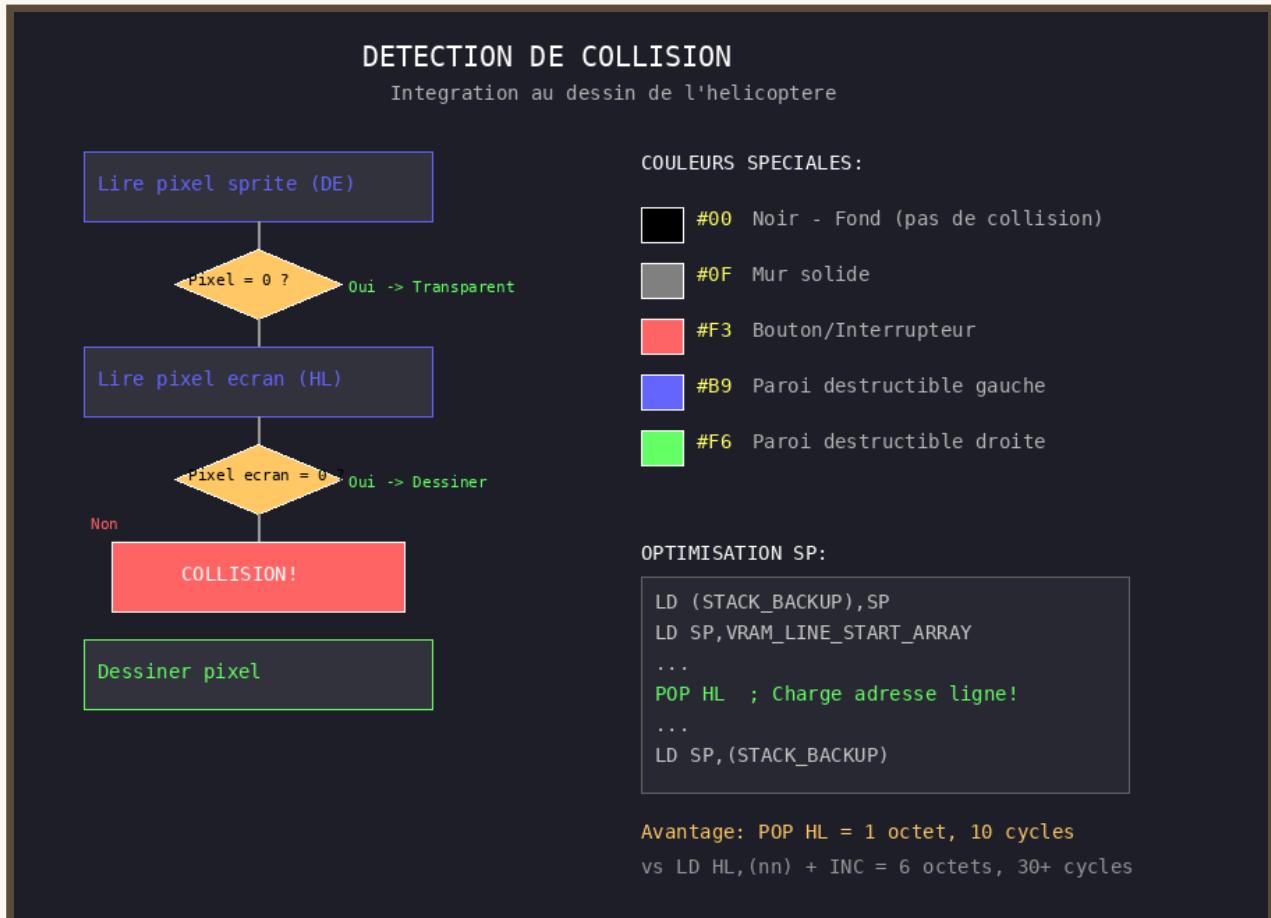
```
UNUSED_FULL_SCROLL_LEFT:    ; Code fantôme !
; Scrolling fluide
; Abandonné en cours de route
```

Quelqu'un avait prévu mieux. Et puis quelqu'un d'autre a dit "non, on a plus le temps".

L'histoire d'un compromis. L'histoire de toutes les magouilles.

CINQUIEME PARTIE : LES COLLISIONS (L'ARNAQUE ULTIME)

Chapitre 10 : Le système malin



Collision intégrée au dessin. Malin, le loustic. - Figure 13

DéTECTER en dessinant

Le coup de génie d'Airwolf, c'est de détecter les collisions PENDANT qu'il dessine l'hélico :

```

DRAW_HELICO:
LD A,(DE)           ; Pixel de l'hélico
OR A
JR Z,.transparent

LD A,(HL)           ; Pixel de l'écran
OR A
JR Z,.dessiner      ; Rien ? On dessine

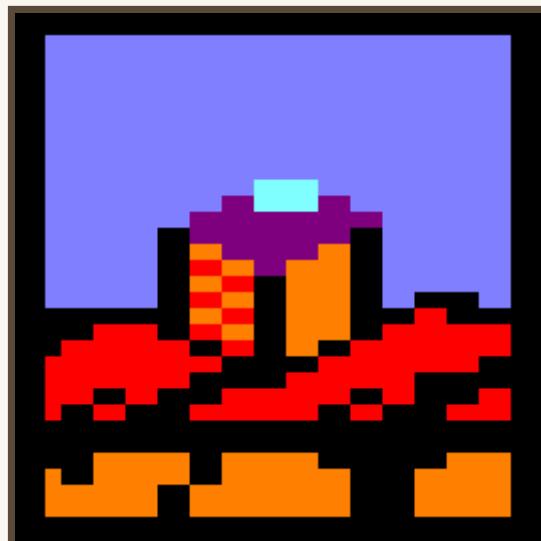
; Y'a quelque chose ? COLLISION !
LD A,#01
LD (HELICO_COLLISION_DETECTED),A

```

Deux opérations en une. C'est beau. C'est élégant. C'est l'œuvre de quelqu'un qui sait ce qu'il fait.

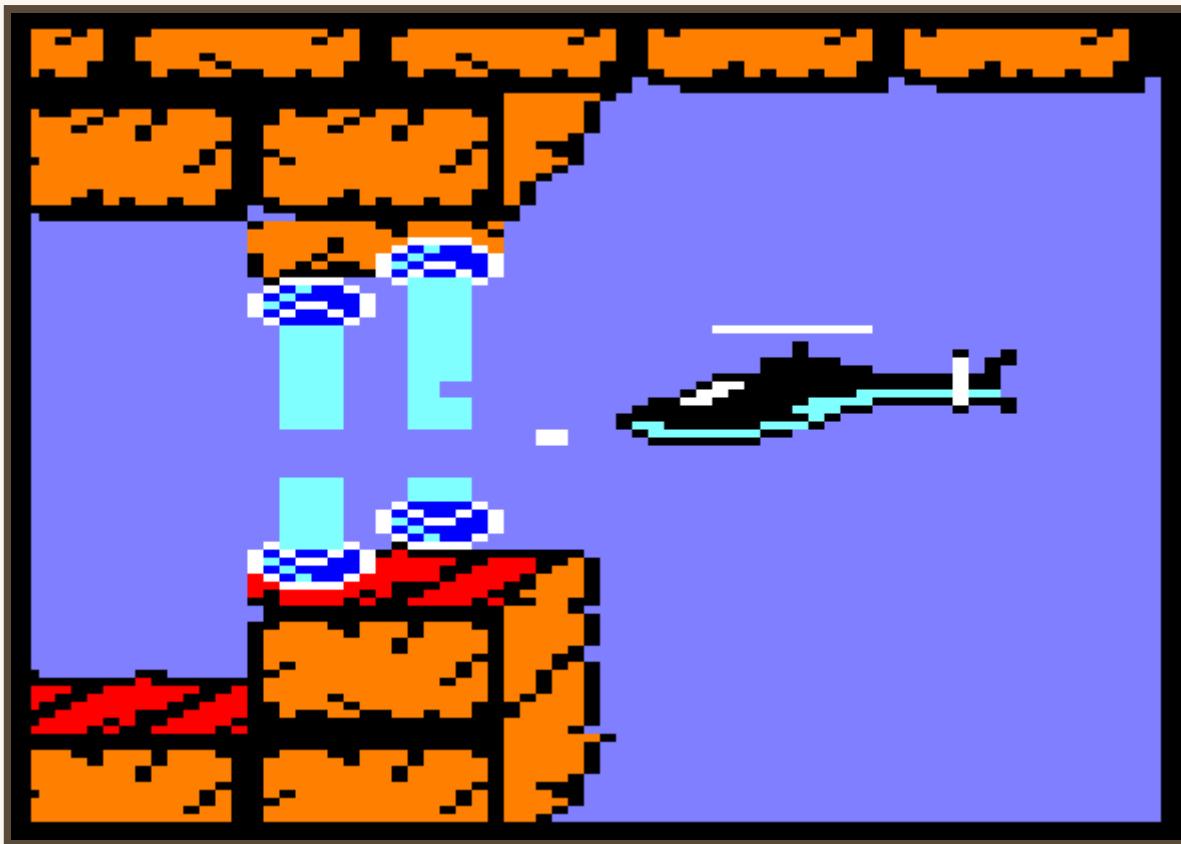
Chapitre 11 : Le cyan, c'est la combine

La couleur qui fait des trucs



Le cyan, c'est l'interrupteur. - Figure 14

Dans ce jeu, le cyan, c'est la couleur magique. Tu tires dessus, il se passe des choses. Un mur tombe, un passage s'ouvre.



Tir sur le cyan, le mur tombe. Simple mais efficace. - Figure 15

L'orange en double

Et ces deux oranges identiques ? J'ai fini par piger. Y'en a un pour les murs, un pour les ennemis. Comme si quelqu'un avait voulu qu'on puisse dégommer les ennemis aussi.

Sauf que ça a jamais été codé. Encore une fonctionnalité partie à la trappe.

| Chapitre 12 : Le coup du Stack Pointer

L'astuce de sioux

Dans `DRAW_HELICO` (#7F1C), y'a une technique qui en dit long sur le niveau du codeur :

```
DRAW_HELICO:  
    LD (STACK_BACKUP),SP      ; On sauvegarde la pile  
    LD SP,VRAM_LINE_START_ARRAY ; On la détourne  
  
    LD B,#14  
.boucle:  
    POP HL                   ; Hop, l'adresse tombe toute seule !  
    ; ... dessin ...  
    DJNZ .boucle  
  
    LD SP,(STACK_BACKUP)     ; On remet tout en place
```

Utiliser le Stack Pointer comme pointeur de données, c'est dangereux mais diablement efficace. Le genre de truc qu'un virtuose fait, pas un amateur.

Alors pourquoi, POURQUOI, ce même jeu contient une boucle infinie à la con ?

Réponse : c'est pas le même gars qui a codé les deux.

SIXIÈME PARTIE : LES ENNEMIS (LES FAIRE-VALOIR)

Chapitre 13 : La fiche signalétique

STRUCTURE DES ENNEMIS			
24 octets par entité			
Offset	Description	Taille	PATTERNS:
#00-01	Adresse VRAM initiale	2	ANIM_RL_45 D-G, 45 frames
#02-03	Pointeur pattern animation	2	ANIM_RULD Carré
#04-05	Adresse sprite de base	2	ANIM_STATIC Immobile
#06	Largeur (octets)	1	
#07	Hauteur (pixels)	1	
#08	Tempo mouvement initial	1	
#09	Compteur tempo mouvement	1	
#0A-0B	Adresse sprite courante	2	
#0C	Nombre frames total	1	
#0D	Frame courante	1	
#0E	Tempo animation	1	
#0F	Compteur tempo animation	1	
#10	Flag actif (0=inactif)	1	
#11-12	Adresse VRAM courante	2	
#13	Duree mouvement initial	1	
#14	Compteur duree mouvement	1	
#15-16	Pointeur pattern courant	2	
#17	Flag reset position	1	

Format pattern:

```
db DIRECTION, DUREE
db DIRECTION, DUREE
db #FF ; Fin/loop
```

DOUBLE ANIMATION:

Animation MOUVEMENT (pattern)	Animation GRAPHIQUE (sprites)
-------------------------------	-------------------------------

Les deux animations sont indépendantes et fonctionnent en parallèle

24 octets par ennemi. Bien rangé. - Figure 16

24 octets chacun

Chaque ennemi, il a sa petite fiche de 24 octets avec tout ce qu'il faut :

Offset	C'est quoi	Remarque
#00-#01	Sa position	Où il crèche
#02-#03	Son pattern	Comment il bouge
#06-#07	Sa taille	Gros ou petit
#10	Actif ou pas	0 = il dort

C'est propre. C'est organisé. Le travail d'un type qui aime son métier.

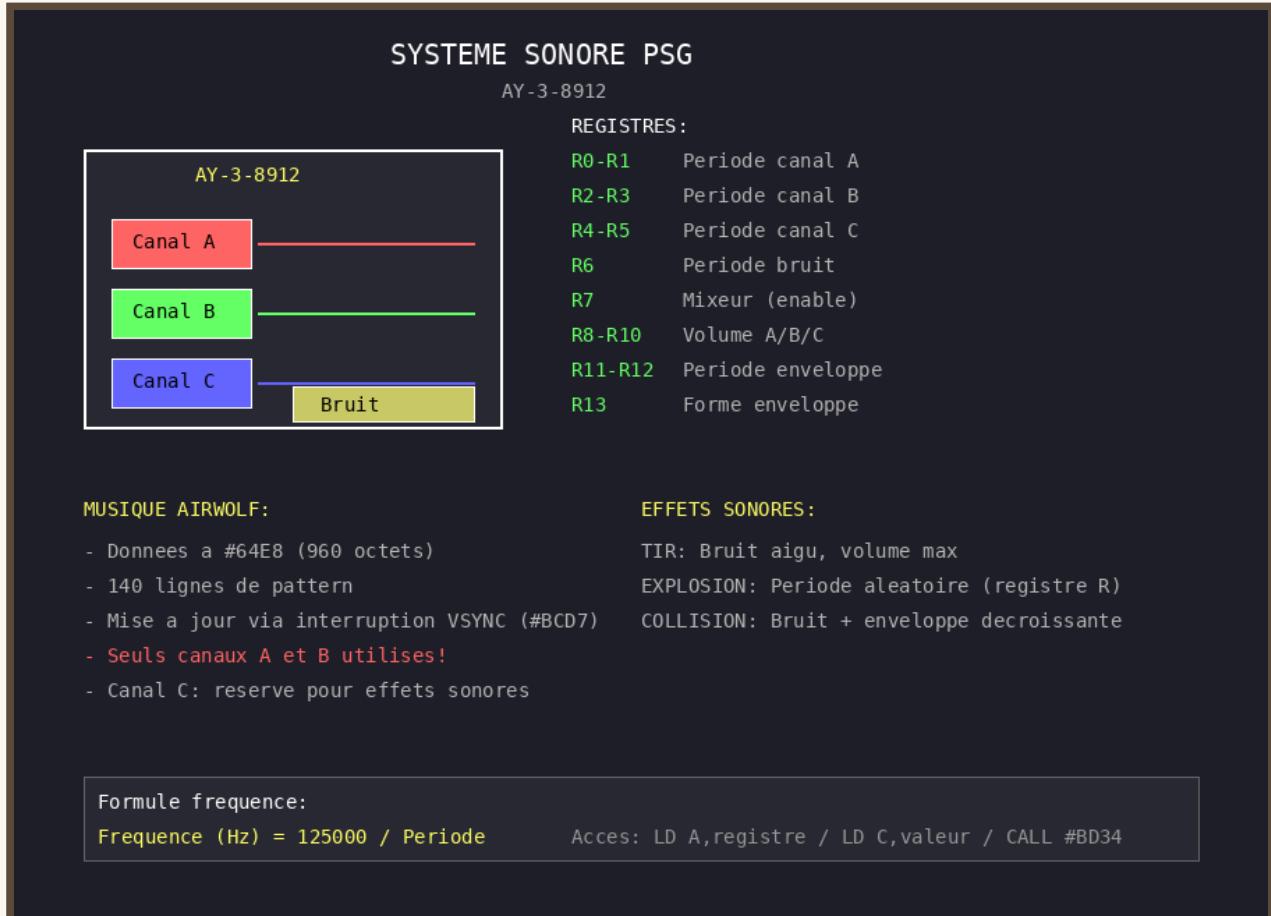
Les patterns de mouvement

```
ANIM_DROITE_GAUCHE:  
    db DROITE, 45      ; 45 images à droite  
    db GAUCHE, 45      ; 45 images à gauche  
    db #FF              ; Et on recommence
```

Simple, efficace, malin. Pas de l'artisanat de zonard.

SEPTIEME PARTIE : LE SON (LA MUSIQUE DES EMMERDEMENTS)

Chapitre 14 : Le PSG



Le AY-3-8912. Trois canaux de bonheur. - Figure 17

Trois canaux, deux utilisés

L'Amstrad a une puce sonore avec trois voix. Dans Airwolf, y'en a que deux qui servent.

```
UPDATE_SOUND_REGISTERS:
; Canal A - check
; Canal B - check
; Canal C - que dalle
```

Encore une fonctionnalité à moitié faite. Ça devient une habitude.

Chapitre 15 : L'explosion



Quatre étapes pour crever. - Figure 18

Quand tu meurs - et tu vas mourir souvent - t'as droit à une jolie animation d'explosion. Au moins, ils ont soigné la mort.

HUITIÈME PARTIE : LA VÉRITÉ (LE POT AUX ROSES)

Chapitre 16 : Le niveau 2, parlons-en

La grande question

Depuis 1985, tout le monde se demande : "C'est quoi le niveau 2 ?"

J'ai retourné le code dans tous les sens. J'ai cherché partout. Et voici la réponse :

Y'a pas de niveau 2.

Pas de routine de chargement. Pas de deuxième carte. Nada. Que dalle. Peau de balle.

La carte de 4352 octets à l'adresse #03E8, c'est tout ce qu'y'a. Point final.

La morale de l'histoire

Le niveau 2, c'était un mirage. Un rêve de gosse. Les joueurs, ils se disaient "si j'étais meilleur, je le verrais". Ben non. Y'avait rien à voir.

C'est pas leur faute. C'est la faute aux développeurs qui ont vendu du vent.

Chapitre 17 : Le sabotage

La zone 8

L'écran final - Zone 8

La zone 8. Le bout du chemin. Le mur. - Figure 19

Voilà la zone 8. L'écran que personne atteint. Et pour cause.

LE MYSTERE DU CRASH

La boucle infinie a l'adresse #698A

```

EXECUTE_BUTTON_ACTION:    ; #6971
    LD A,(VIEW_INDEX)   ; #6971
    CP #01              ; Zone 1?
    JP Z,BUTTON_VIEW_1 ; #6976
    ...
    CP #08              ; #6988 Zone 8?
.infinite:
    JP Z,.infinite     ; #698A BLOQUE!
    ...
    RET

```

CE QUE CA SIGNIFIE:

1. VIEW_INDEX = position du joueur dans le monde
2. Zone 8 = zone finale
3. Quand VIEW_INDEX == 8
4. JP Z saute a lui-même
5. Le CPU boucle a #698A

=> CE N'EST PAS UN BUG
=> C'EST INTENTIONNEL

DUMP MEMOIRE ORIGINAL:

#6988: FE 08 CA 8A 69 FE 0C ...

CP #08 / JP Z,#698A / CP #0C

DUMP MEMOIRE PATCHÉ:

#6988: FE 08 00 00 00 FE 0C ...

CP #08 / NOP NOP NOP / CP #0C

CONCLUSION: Les développeurs ont délibérément bloqué le jeu en zone 8.
3 octets (CA 8A 69) remplacent par 3 NOPs (00 00 00) dans Airwolf Reloaded.

L'arnaque du siècle. - Figure 20

Le code qui tue

```

EXECUTE_BUTTON_ACTION:          ; #6971
    LD A,(VIEW_INDEX)        ; T'es où ?
    CP #01                  ; Zone 1 ?
    JP Z,BUTTON_VIEW_1      ; OK
    ...
    CP #03                  ; Zone 3 ?
    JP Z,BUTTON_VIEW_3      ; OK
    ...
    CP #08                  ; Zone 8 ?
.arnaque:
    JP Z,.arnaque           ; TOURNE EN ROND ÉTERNELLEMENT

```

Trois octets. **CA 8A 69**. C'est tout ce qu'il a fallu pour flinguer le jeu.

L'aveu

C'est pas un bug. C'est un meurtre avec pré-méditation.

Les types, ils savaient que la zone 8 marchait pas. Ils avaient plus le temps de la finir. Alors plutôt que de livrer un truc qui plante ouvertement, ils ont mis une boucle infinie en se disant : "De toute façon, qui c'est qui va arriver jusque-là ?"

Des malins. Des escrocs, mais des malins.

Chapitre 18 : Les indices

Le code mort

Le scrolling pixel (#7382) ? Jamais utilisé. Quelqu'un l'avait codé, quelqu'un d'autre l'a abandonné.

Les ennemis immortels

Les deux oranges ? Prévus pour rendre les ennemis destructibles. Jamais implémenté.

Le troisième canal

La musique sur deux canaux au lieu de trois ? Pas eu le temps de finir.

Le verdict

Ce jeu, c'est un chantier abandonné. Un immeuble à moitié construit. Les fondations sont solides, y'a des pièces magnifiques, mais le toit, il manque.

Chapitre 19 : Plusieurs coupables

Les styles de code

En analysant le code, on voit bien que c'est pas le même gars qui a tout fait : - Y'a des routines de virtuose (le Stack Pointer) - Y'a des trucs de branquignol (la boucle infinie) - Les conventions changent d'une section à l'autre

Le scénario probable

Un ou deux bons développeurs ont commencé le projet. Puis les délais ont serré. On a rajouté du monde. La qualité a baissé. Et à la fin, quelqu'un a dit "on livre comme ça, on verra bien".

C'est pas un crime individuel. C'est un crime collectif. Le crime de l'industrie du jeu vidéo des années 80.

NEUVIÈME PARTIE : LA RÉSURRECTION

Chapitre 20 : Airwolf Reloaded

Quarante ans après, avec l'artiste Titan, on a réparé les dégâts.

Les trois octets de la mort (`CA 8A 69`) ont été remplacés par trois NOP (`00 00 00`).
La boucle infinie a été brisée.

Pour la première fois de l'histoire, Airwolf peut être terminé.

C'est pas de la nostalgie. C'est de la justice.

DIXIÈME PARTIE : LA PHILOSOPHIE DU BIDOUILLEUR

Chapitre 21 : Ce que cette histoire nous apprend

Sur les développeurs

Derrière chaque ligne de code, y'a un être humain. Avec ses talents, ses limites, et ses patrons qui lui mettent la pression.

Les gars qui ont fait Airwolf, c'était pas des incompétents. C'était des artisans pris dans un système qui valorise la quantité plutôt que la qualité.

Sur la curiosité

Cette enquête, elle est partie d'une insomnie. D'une question conne posée à trois heures du mat'.

Et elle m'a occupé des semaines. Elle m'a fait découvrir des trucs fascinants. Elle m'a connecté avec des inconnus d'il y a quarante ans.

La curiosité gratuite, celle qui sert à rien, c'est peut-être la plus précieuse.

Chapitre 22 : Les leçons techniques

Malgré le désastre, y'a des pépites dans ce code : - Le détournement du Stack Pointer (astucieux) - La collision intégrée au dessin (élégant) - Le système de patterns pour les ennemis (propre)

Si vous codez des jeux rétro, y'a des trucs à piquer là-dedans.

ANNEXES

Annexe A : La carte mémoire

```
#0000-#003F : Variables système
#03E8-#14E7 : TILEMAP (tout le jeu est là)
#14E8-#4F67 : TILESET (les graphismes)
#6971       : EXECUTE_BUTTON_ACTION (scène du crime)
#698A       : LA BOUCLE INFINIE (l'arme du délit)
#7F1C       : DRAW_HELIICO (le chef-d'œuvre)
#C000-#FFFF : Mémoire vidéo
```

Annexe B : Les codes de triche

Pour ceux qui veulent voir la zone 8 sans se faire mal :

Touche	Effet
T	Arrête le chrono
G	Annule la gravité
C	Passe à travers les murs

Annexe C : Le best-of des routines

Adresse	Nom	Verdict
#6971	EXECUTE_BUTTON_ACTION	COUPABLE
#698A	La boucle	ARME DU CRIME
#7382	Scrolling pixel	VICTIME (jamais utilisé)
#7F1C	DRAW_HELIICO	CHEF-D'ŒUVRE

ÉPILOGUE : UN DERNIER VERRE ?

Je repose le clavier. L'écran de l'émulateur brille encore, mais l'enquête est finie.

Airwolf, c'est l'histoire d'un rêve qui a mal tourné. Des types qui voulaient faire un bon jeu, et un système qui les en a empêchés. Des promesses non tenues, des fonctionnalités abandonnées, et à la fin, un sabotage pour masquer la misère.

Mais y'a pas que du mauvais dans cette histoire. Y'a du talent gâché, des astuces brillantes, et surtout, y'a une leçon : le code, il ment pas. Quarante ans après, il raconte encore l'histoire de ceux qui l'ont écrit.

Et maintenant, grâce à Airwolf Reloaded, ces types-là peuvent enfin être fiers. Leur jeu peut être terminé. Le niveau 2 existe pas, mais la fin, elle, est maintenant accessible.

C'est pas grand-chose, mais c'est déjà ça.

"Les cons, ça ose tout. C'est même à ça qu'on les reconnaît."

Sauf que là, les cons, c'était pas les développeurs. C'était ceux qui leur ont pas donné le temps de finir.

FIN

Un texte écrit avec l'assistance de Claude, dans le style de Michel Audiard. Parce que le reverse-engineering, c'est comme les bons mots : faut savoir où chercher.

Décembre 2025

À propos de l'auteur

Alain Le Guirec est un type qui aime bien démonter les trucs pour voir comment c'est foutu dedans. C'est lors d'une nuit d'insomnie qu'il s'est attaqué à Airwolf, et il a pas été déçu du voyage.

De cette aventure sont nés : - **Airwolf Pico** : une version qui marche, sur PICO-8 (https://www.lexaloffle.com/bbs/?pid=airwolf_pico-0) - **Airwolf Reloaded by Titian** : la version originale, enfin réparée

"La culture, c'est comme la confiture : moins on en a, plus on l'étale."

Sauf que là, on en a. Et on l'étale quand même.