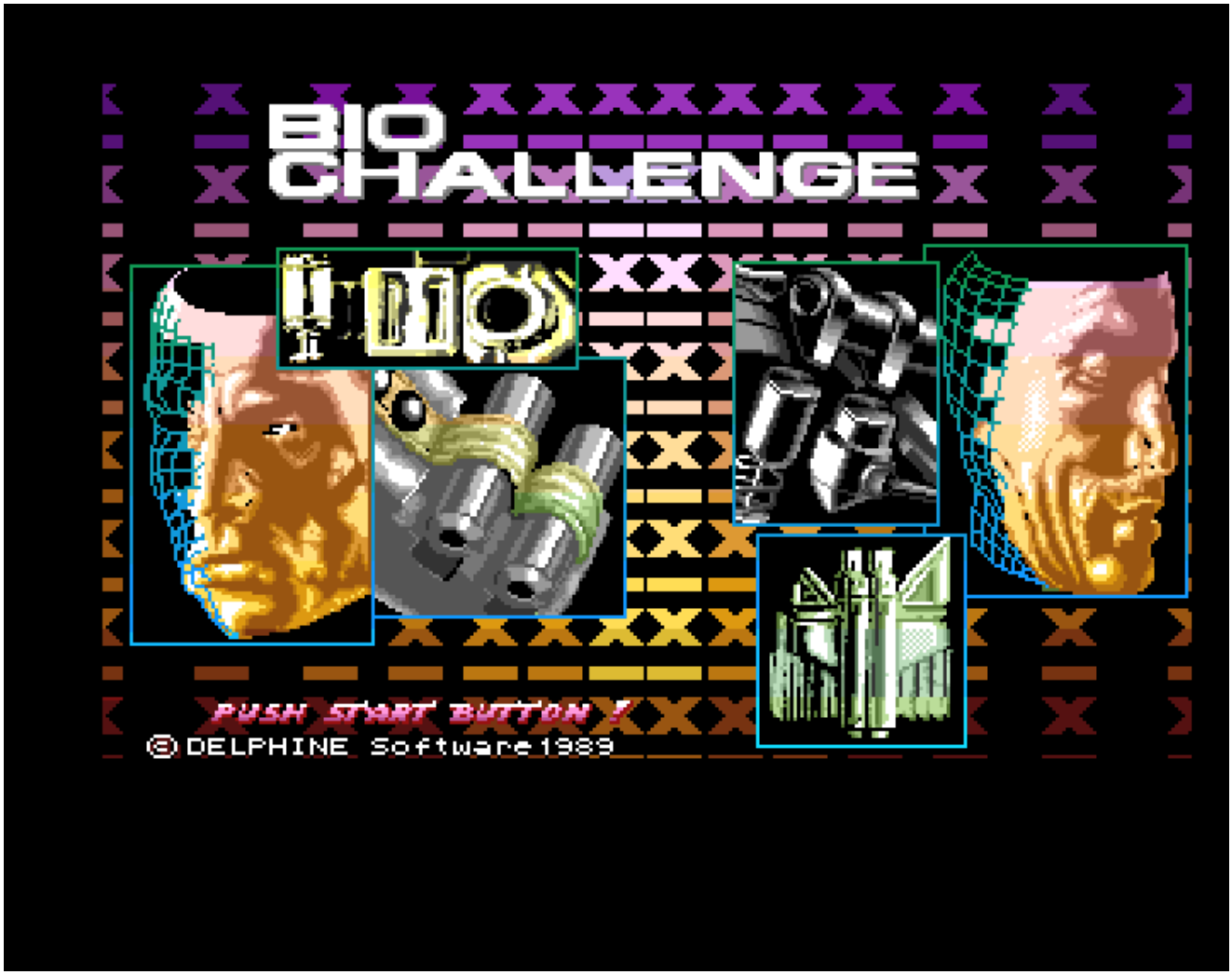


Tutoriel	**AmigaCracking : Bio Challenge**
Protection	**Single Track**
Auteur Original	**Rob**
Soumit par (sur flashtro.com)	**Rob [2004-07-25]**
Version	**14/03/2018 Gi@nts**
Vérification/Correction	**V2, Testé et fonctionnel de A à Z**

BIO CHALLENGE
* CRACK TUTORIEL *

Table des matières

Matériels nécessaire

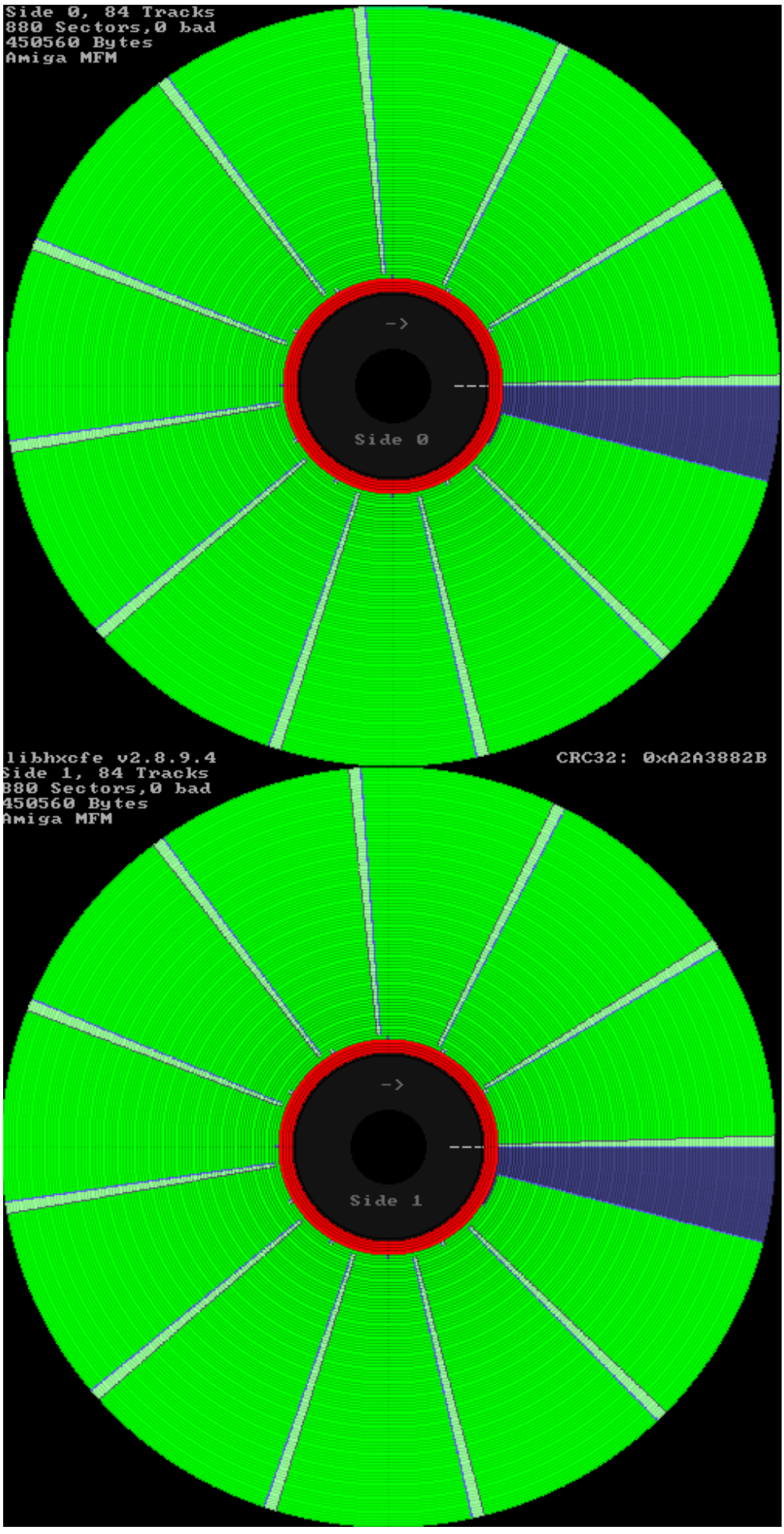
1) Un AMiGA avec 512K ou l'émulateur WINUAE
2) Une Carte ACTION REPLAY MKIII (ou ça ROM Image)
3) Une version de Xcopy Pro (image ou disquette)
4) Le jeu Original BIO CHALLENGE ou son image CAPS (SPS 1734)

General Info

Ce tutoriel Français est basé sur le tutorial original de Rob.
Ce document n'est pas une traduction mot par mot de celui-ci mais plus une nouvelle version.
Suivit pas à pas avec des nouvelles informations.

Bon tuto.

Gi@nts

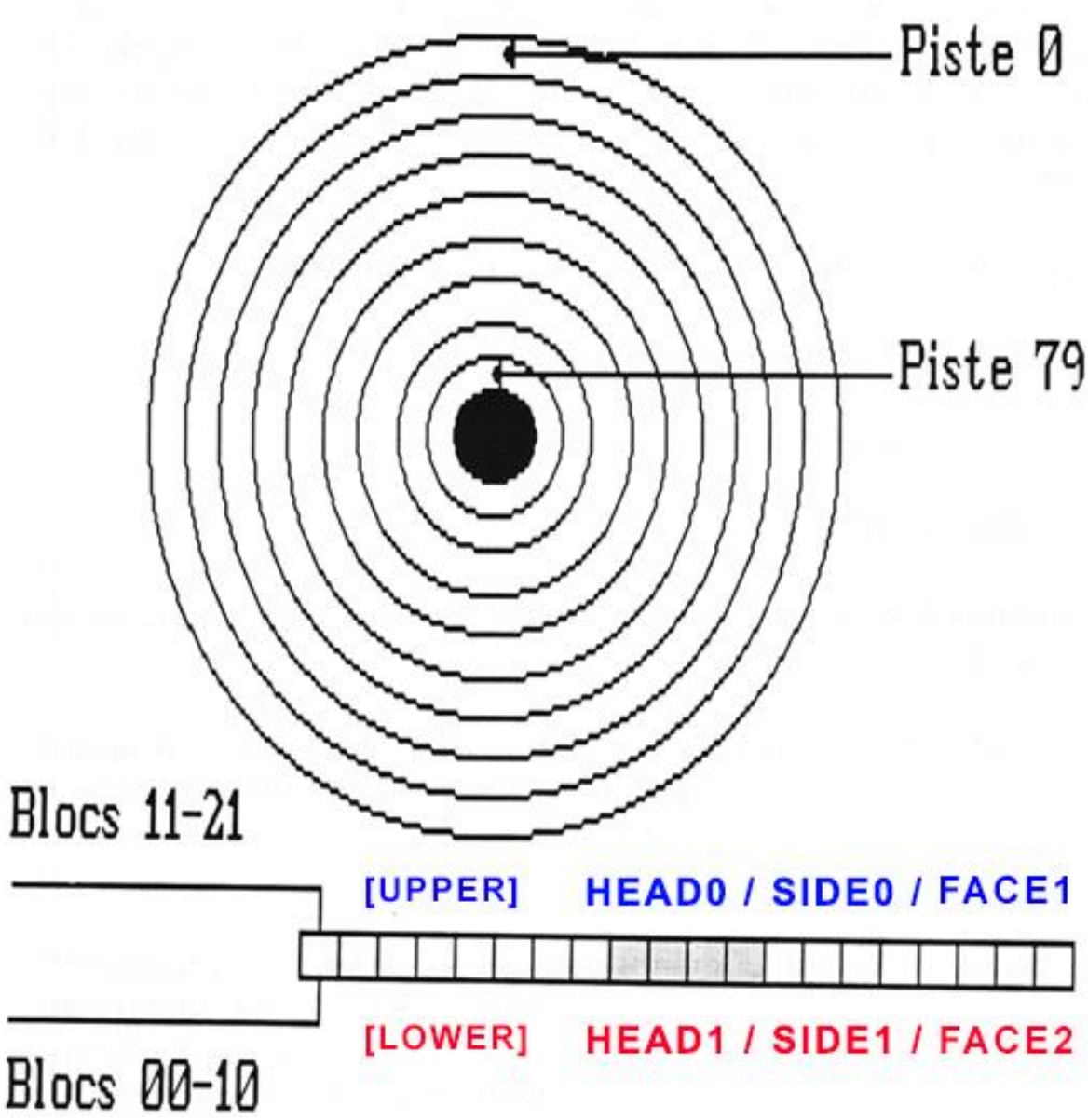

Side 0, 84 Tracks
880 Sectors,0 bad
450560 Bytes
Amiga MFM
->
Side 0
libhxcfe v2.8.9.4
Side 1, 84 Tracks
880 Sectors,0 bad
450560 Bytes
Amiga MFM
CRC32: 0xA2A3882B
->
Side 1

Agencement des disquettes Amiga v1.1

En France :

On utilise des termes comme : *piste, bloc, secteur, face*…
Piste : 0 à 79 Certaine disquette pousse jusqu'à 81 voire 82 pistes mais le standard reste quand même 80 pistes (de 0 à 79)
Face : 0/1 ou **1/2** ou **A/B**, Dessus ou dessous tout simplement. Sur Amiga nous avons deux faces utilisées sur 99% des jeux.

Chaque piste, pour un format standard 'AmigaDOS' est composée de plusieurs *bloc ou secteur*, en général 11 **par face**.
Le terme piste peut désigner l'ensemble d'une piste (les deux 'side' du disque), ou uniquement une 'side' d'une piste.
Une piste standard *amigados* est découpée en plusieurs partie appelé **bloc**, **secteur**, **sector**.



Dans d'autre pays :

On utilisera d'autre terme, comme **sector, keys, tracks, cylindre**, **head**…

Le terme **track** par exemple que l'on aurait vite fait de traduire 'piste' ne colle pas forcément à notre description française.
En général, le terme **tracks** désigne toujours une position sur la disquette mais **elle va de 0 à 159** (soit 160 **tracks**)
Le maximum étant 160 et non 80 car on a deux faces bien sûres, en fait, elle correspond à une piste sur une face.

Il peut néanmoins arriver que l'on utilise dans des tuto anglo-saxon le terme *tracks* dans le sens 'piste' en français (donc de 0 à 79 et non de 0 à 159). Mais en règle générale, il a plutôt une plage de 0 à 160.

C'est le terme **cylindre** qui 'colle' plus à notre définition française de **piste.**
En effet, il est courant d'utiliser le terme **cylindre** pour désigner une position sur la disquette de 0 à 79.

Le terme **sector** ou **key** quant à lui corresponds au terme français **bloc** ou **secteur.**

Sur une disquette au format ***Amigados***, nous avons 880ko et nous avons 11 secteurs par face, par piste.

La taille d'une piste ayant une valeur physiquement maximum.
Le nombre maximum de **sector** sur une piste dépends assez logiquement de la taille de ses **sectors.**

Pref…beaucoup de terme qui ne sont pas forcément utilisé dans leur sens propre, le mieux est de lire un tuto et de comprendre quel sens l'auteur a voulu leurs donner.

Il existe aussi un autre type d'appellation utilisé par exemple par le logiciel **MFM-Warp** de Ferox*
**C'est un programme qui scan le disque en bas niveau et essaye d'en réaliser une copie.*

Track	Calcul	Résultat	Format utilisé sous MFMWarp
0	0/2	0 et pair	0.0
1	1/2	0 et impair	0.1
2	2/2	1 et pair	1.0
3	3/2	1 et impair	1.1
156	156/2	78 et pair	78.0
157	157/2	78 et impair	78.1
158	158/2	79 et pair	79.0
159	159/2	79 et impair	79.1

On notera que :

Le premier secteur (secteur 0) appelé aussi *bootbloc* commence sur la *lowerSide* en piste 00
et se fini en piste 79 sur le *upperside*

En *tracks* c'est le même système sauf que l'on terminera en Track 179 et non 79.

La piste Zero est celle situé le plus à l'extérieure du disque.

Le 1er secteur logique, donc le premier bloc sur la disquette, se trouve **piste 0 secteur 0**
Les *bloc* se suivent physiquement mais ne sont pas forcément ordonnée, on parle aussi d'entrelaceme
nt.

Le bloc 11 (si on part de 0 bien sur) n'est pas le 1er secteur de la seconde piste mais le 1er secteur *de la face suivante.*
(voir image ci-dessus)

En format **Amigados**, **la taille d'un secteur est de 512 octets**
Ce qui nous donnes comme taille disponible : 512*11 secteurs*80 pistes*2 faces = 901 120 octets soit 880Ko
Une 'track' AmigaDos a une taille de 512 * 11 = **5632** en décimal soit ***$1600*** *octets*

Mise en application sous l'AR :

Il existe deux commandes sous l'AR qui permettent de charger sauver des pistes, à savoir : **RT** et **WT**
Elles fonctionnent pareil.
L'une permet la lecture, l'autre l'écriture.

#**RT** alias Read Track. Permet le chargement de donnée située sur la disquette vers la mémoire.
#la première valeur sera la track de **départ** [0 à 159] à indiquer en hexa. **/!\ ne pas confondre avec piste**

#La seconde valeur sera le nbr de demi track à copié à partir de là.

#**WT** alias Write Track. Permet la sauvegarde de donnée située en mémoire vers la disquette.

Exemples :

```
RT 20 1 50000
```
Start Track = $20 et taille à lire = 1
On copiera donc la piste !16 (en décimal) side 0 en mémoire $50000

Oui car **20** est donné en hexa, ce qui nous donnes !32 en décimal **mais** il indique une track (de 0 à 159) **PAS en piste.**
Pour avoir l'équivalent en piste on divisera donc par 2 (car deux faces).

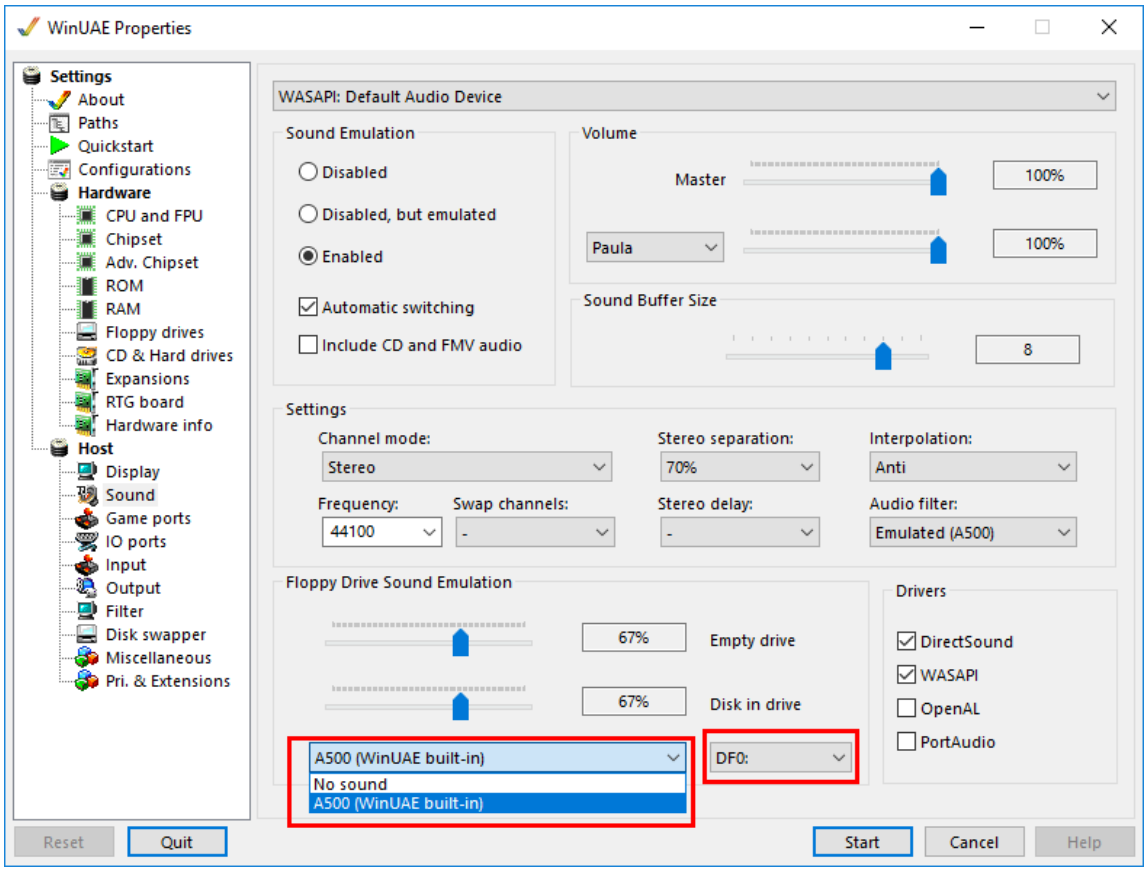
$20/2=$10 = !16 (en décimal donc) et comme il n'y a pas de retenu on est sur la face0.

```
RT 21 2
```
Start Track = $21 et taille à lire = 2
On copiera la piste !16 side 1 et la piste !17 side 0 en mémoire 50000

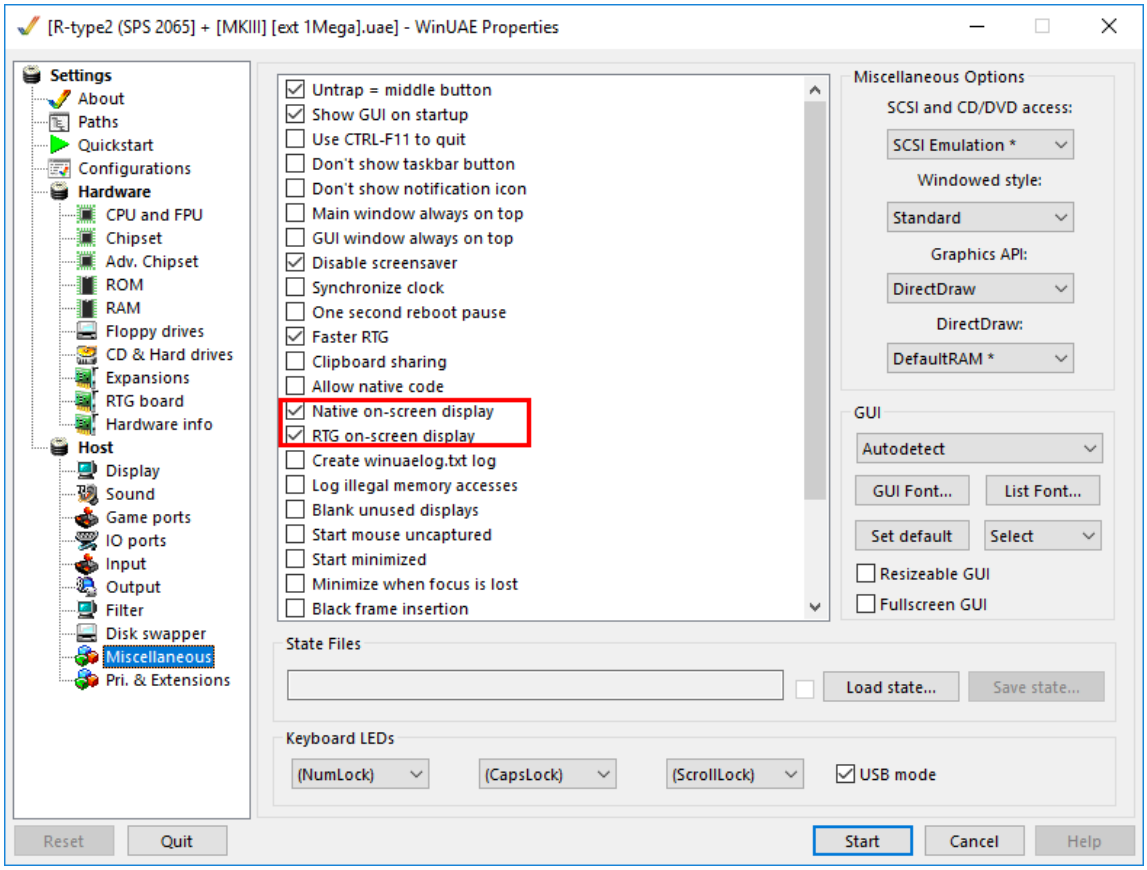
21 est donné en hexa **donc $21 = !33** en décimal.
33/2 = 16.5 ,donc piste 16 side 1 et comme on continue à lire/copier les donnes (**taille à lire =2**), on continue la copie.
On change donc de track car on est déjà sur la face 1 (il existe que 2 faces sur une disquette)
On arrive donc sur la prochaine track à savoir, piste 17 en side 0 puisque que c'est la première face au niveau structure la side 0.

WinUAE

Pour ceux qui utilisent ***winUAE*** pour ces tutoriels (j'imagine, la plupart des personnes), Je vous conseille fortement d'activer le son des lecteurs de disquette histoire d'entendre ce que le lecteur effectue comme accès.
HOST -> SOUND -> FLOPPY DRIVE SOUND EMULATION - > DF0 Built-In

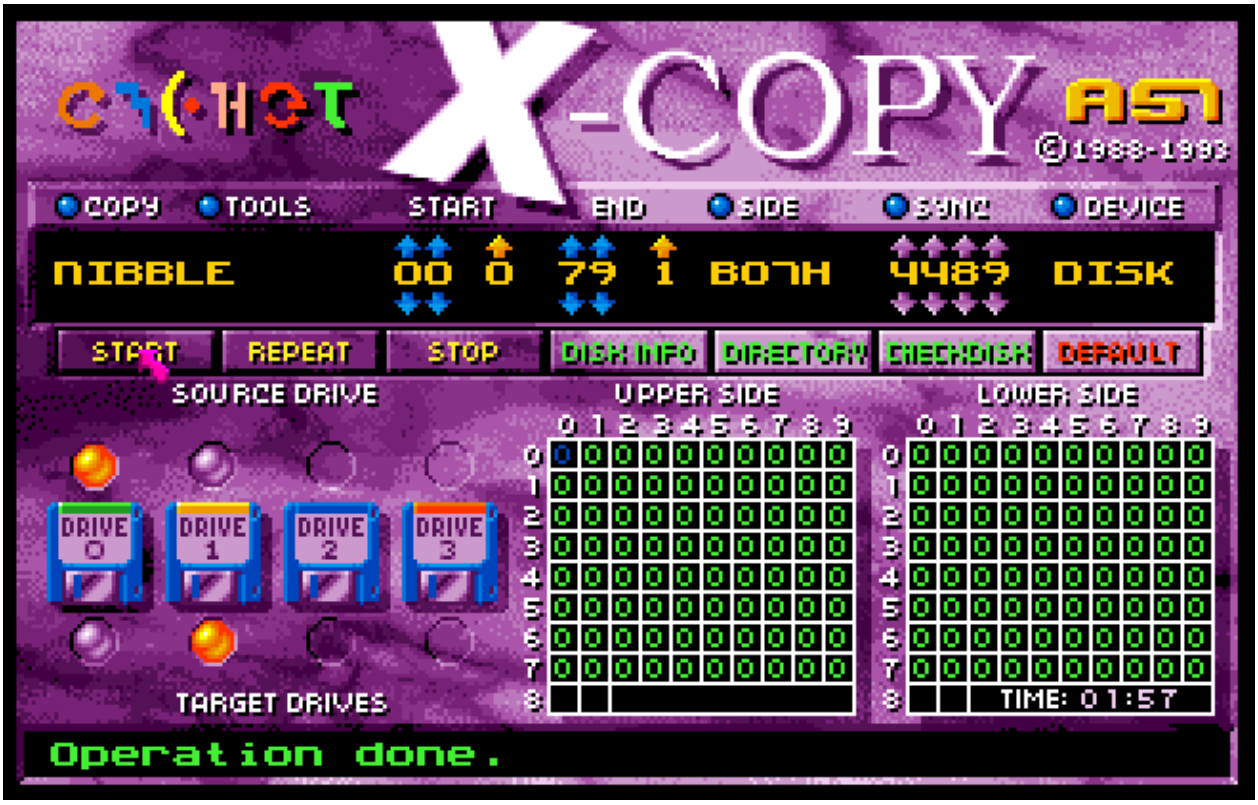


Voir même, pour plus d'information. Par exemple afficher sur qu'elle face l'on se trouve, d'activer :
Host -> Miscellaneous -> Native on-screen display AND RTG on-screen display



Part 1 X-Copy

Comme dans Tous bon hack qui se respecte, on va commencer par essayer de copier le disk original.
À l'aide de X-Copy Pro, effectuer une copie de la disquette originale en mode **NIBBLE**



Hummmm, quelque chose d'étrange en T00.1 à part ça, rien de spécial semble t-il.

Rebooter votre amiga avec cette copie insérée dans le lecteur DF0 et booter dessus.
Assez rapidement, l'Amiga Crash.
Mais gardons quand même cette copie sous la main.

Part 2 Analyse de l'image IPF

FILENAME	1734_BioChallenge_AMIGA.ipf
TYPE	Floppy_Disk
ENCODER	CAPS(V1)
FILE	1734(V1)
DISK	0
TRACK	00-83
SIDE	0-1
PLATFORM	Amiga
REVOLUTION	4
PROTECTION	COPYLOCK [T00.1]

Mon script d'analyse nous indique la présence d'une protection **Copylock**
D'ailleurs si on regarde la piste 00.1 de plus prêt on peut voir le nombre de block détecté à 15

TrackNumber	Size Record (bytes)	Crc	Status	Track Size	Detail Tr. Size	Start Byte	Bit	DataKey	Block	Density	Signal	Encoder	Flag
T00.0	80	6511DB5D	Good	12542	100330 bits = Data=95744 + Gap=4586	7448	59587	001	11	Auto	cell_2us	0	None
T00.1	80	93EE94EB	Good	12546	100347 bits = Data=97824 + Gap=2523	15	124	002	15	Auto	cell_2us	0	None
T01.0	80	7281F090	Good	12543	100340 bits = Data=95744 + Gap=4596	819	6559	003	11	Auto	cell_2us	0	None
T01.1	80	B444649C	Good	12543	100342 bits = Data=95744 + Gap=4598	8258	66070	004	11	Auto	cell_2us	0	None
T02.0	80	217B06CC	Good	12543	100340 bits = Data=95744 + Gap=4596	8859	70876	005	11	Auto	cell_2us	0	None
T02.1	80	DA7E4406	Good	12543	100342 bits = Data=95744 + Gap=4598	3308	26467	006	11	Auto	cell_2us	0	None
T03.0	80	80DB77B4	Good	12543	100340 bits = Data=95744 + Gap=4596	4234	33872	007	11	Auto	cell_2us	0	None
T03.1	80	B7F717DB	Good	12543	100343 bits = Data=95744 + Gap=4599	11702	93621	008	11	Auto	cell_2us	0	None
T04.0	80	27BDE3E0	Good	12542	100336 bits = Data=95744 + Gap=4592	354	2833	009	11	Auto	cell_2us	0	None
T04.1	80	74F42DFE	Good	12543	100341 bits = Data=95744 + Gap=4597	7265	58126	010	11	Auto	cell_2us	0	None
T05.0	80	69E1768D	Good	12543	100340 bits = Data=95744 + Gap=4596	7896	63172	011	11	Auto	cell_2us	0	None
T05.1	80	FC61B211	Good	12543	100342 bits = Data=95744 + Gap=4598	2769	22152	012	11	Auto	cell_2us	0	None
T06.0	80	C36EEBE7	Good	12543	100342 bits = Data=95744 + Gap=4598	4017	32142	013	11	Auto	cell_2us	0	None
T06.1	80	5A3A3C42	Good	12543	100342 bits = Data=95744 + Gap=4598	11425	91403	014	11	Auto	cell_2us	0	None
T07.0	80	F765A20A	Good	12543	100340 bits = Data=95744 + Gap=4596	12058	96468	015	11	Auto	cell_2us	0	None
T07.1	80	6167F1DB	Good	12543	100342 bits = Data=95744 + Gap=4598	7231	57851	016	11	Auto	cell_2us	0	None
T08.0	80	E4CCB408	Good	12543	100339 bits = Data=95744 + Gap=4595	8180	65442	017	11	Auto	cell_2us	0	None
T08.1	80	7BF004DE	Good	12543	100340 bits = Data=95744 + Gap=4596	3028	24229	018	11	Auto	cell_2us	0	None
T09.0	80	B1E46CE1	Good	12543	100342 bits = Data=95744 + Gap=4598	3177	25421	019	11	Auto	cell_2us	0	None
T09.1	80	232C10EE	Good	12543	100343 bits = Data=95744 + Gap=4599	10442	83539	020	11	Auto	cell_2us	0	None

TRACK		Data Length (bytes)		Data (bits)				CRC32 of the complete Extra Data Block			Adress
Data block Description	Sector ID	Data		bytes/sector	GAP		Codage	GapDef	DataOff		Adresse
		MFM bits	bytes		MFM bits	bytes			MFM bits	bytes	
[T00.0]		6446		51568				71DBE8EA			13576-20021
#0		8704	545		0	1	MFM	0352	0352	15	13576-13607
#1	3	8704	545	512	0	1	MFM	0906	0906	57	13608-13639
#2	4	8704	545	512	0	1	MFM	1460	1460	92	13640-13671
#3	5	8704	545	512	0	1	MFM	2014	2014	126	13672-13703
#4	6	8704	545	512	0	1	MFM	2568	2568	161	13704-13735
#5	7	8704	545	512	0	1	MFM	3122	3122	196	13736-13767
#6	8	8704	545	512	0	1	MFM	3676	3676	230	13768-13799
#7	9	8704	545	512	0	1	MFM	4230	4230	265	13800-13831
#8	10	8704	545	512	0	1	MFM	4784	4784	300	13832-13863
#9	0	8704	545	512	0	1	MFM	5338	5338	334	13864-13895
#10	1	8704	545	512	4586	287	MFM	5892	5892	369	13896-13927
[T00.1]		6780		54240				A6C61D47			20050-26829
#0		8704	545		0	1	MFM	0480	0480	31	20050-20081
#1		8704	545		0	1	MFM	1034	1034	65	20082-20113
#2		8704	545		0	1	MFM	1588	1588	100	20114-20145
#3		8704	545		0	1	MFM	2142	2142	134	20146-20177
#4	0	8704	545	512	0	1	MFM	2696	2696	169	20178-20209
#5	1	8704	545	512	0	1	MFM	3250	3250	204	20210-20241
#6	2	8704	545	512	0	1	MFM	3804	3804	238	20242-20273
#7	3	8704	545	512	0	1	MFM	4358	4358	273	20274-20305
#8	4	8704	545	512	0	1	MFM	4912	4912	308	20306-20337
#9	5	8704	545	512	0	1	MFM	5466	5466	342	20338-20369
#10	6	8704	545	512	0	1	MFM	6020	6020	377	20370-20401
#11	7	352	23	512	387	25	MFM	6574	6574	411	20402-20433
#12	8	576	37	512	387	25	MFM	6615	6615	414	20434-20465
#13	9	576	37	512	385	25	MFM	6670	6670	417	20466-20497
#14	10	576	37	512	1364	86	MFM	6725	6725	421	20498-20529

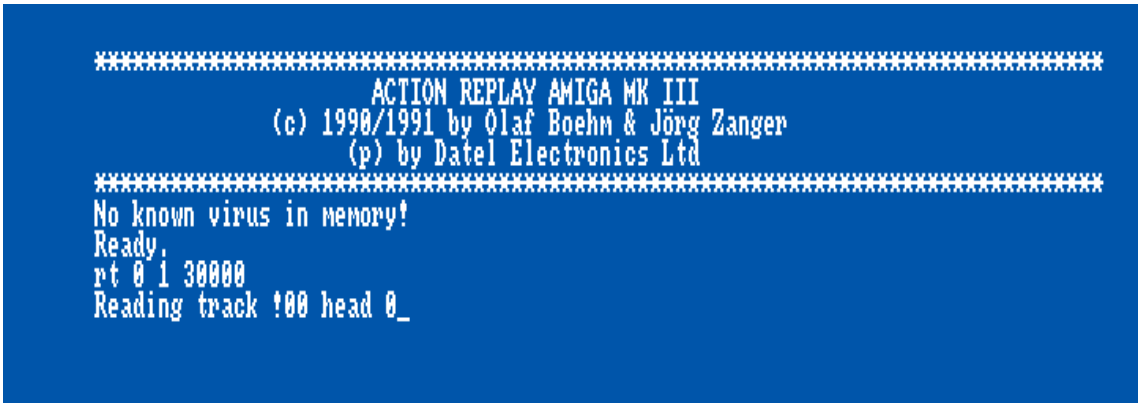
On peut voir que 15 'block' sont détectées en T00.1 alors que normalement on a 11 block sur un disk AmigaDOS.
Que la taille de la piste est plus importante que les autres à savoir **6780 bytes** (logique vue que l'on est sur une piste plus longue), alors que l'on est sur **6446 bytes** sur toutes les autres pistes.

Part 3 Let's do it

Insérer la disquette originale du jeu dans le lecteur de l'Amiga
et chargeons la première piste dans la mémoire pour voir ce qui se passe.

#RT alias Read Track, permet le chargement de la track 0 à 1 (1ère piste de la face 0)
#M, alias Visualisation mémoire HEXA/ASCII

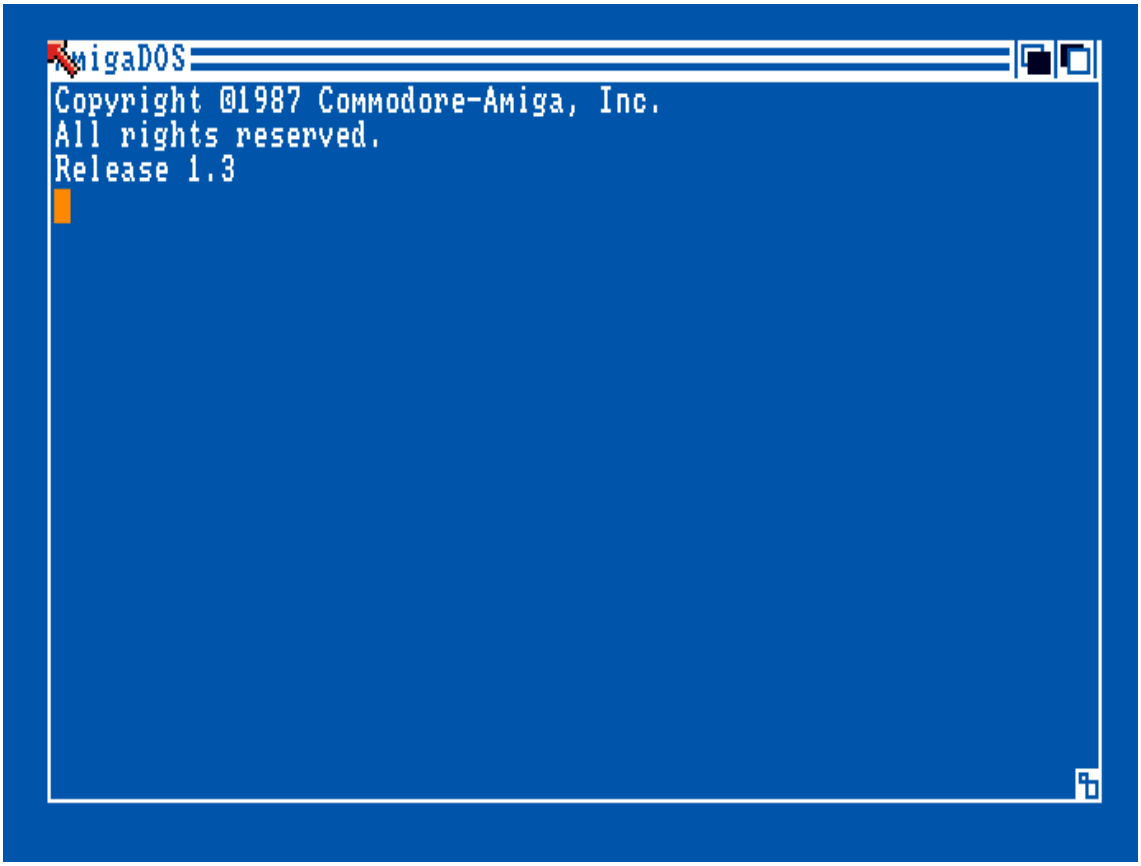
Entrer dans votre AR et taper le texte suivant : `rt 0 1 30000`



Taper : `m 30000`

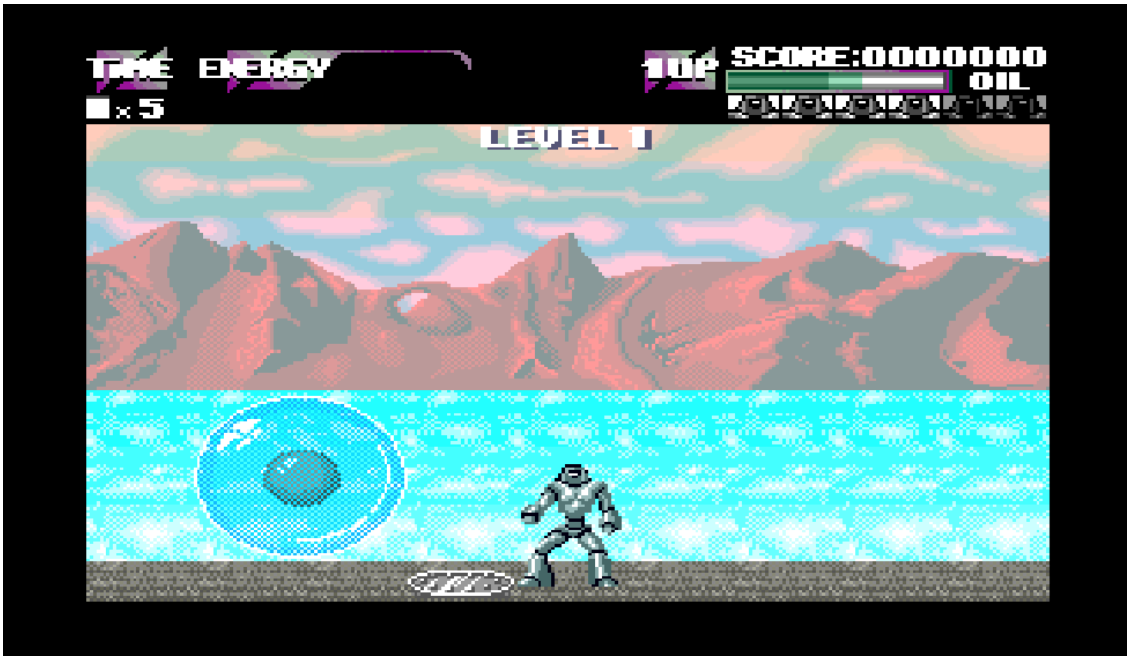
Comme le montre clairement l'image ci-dessus, la 1er piste est standard avec un *header* '**DOS**'
On peut aussi voire assez clairement des informations ascii qui nous laissent penser qu'il s'agit d'une protection **RNC copylock**

Réinsérer la disquette **originale** du **jeu** et **booter** dessus.
Après quelques instants on peut voir un **CLI** ainsi que le workbench apparaître suivi du chargement du jeu.
Nous allons nous servir de cette '*petite faille*' *'boot cli'* pour passer la protection très simplement.



Réinsérer votre **copie** préalablement effectuée dans le lecteur de l'amiga, **entrer** dans votre **AR**
#INSTALL, alias BOOTBLOCK INSTALL, permet d'installer un secteur de boot sur l'unité indiquée, 0=DF0, 1=DF1
et **Taper :** `INSTALL 0`

Rebooter votre Amiga, **tester** le jeu.


TIME
ENERGY
x 5
TOP
SCORE:0000000
OIL
LEVEL 1