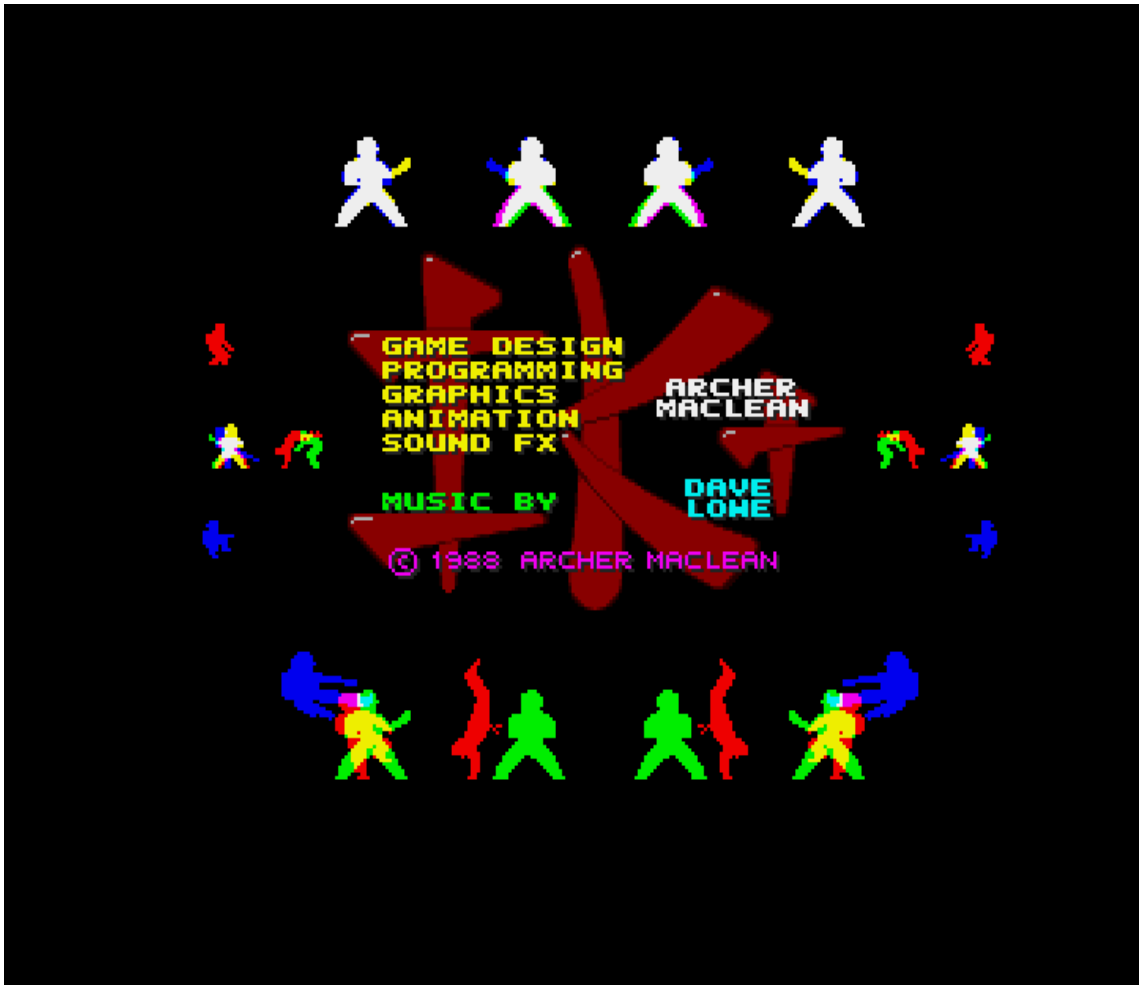


Source :
flashtro.com
Traducteur : Gi@nts

Tutorials	**AmigaCracking : MFM : IK +**
Protection	**MFM (multi file)**
Original Author	**Rob**
Submitted by (on flashtro.com)	**musashi9 Date: 2015-09-08 15:25**
Version FR	**16/02/2015 Gi@nts v1.1**
Re-lecture	**mikedafunk**

* IK+ CRACK TUTORIAL *

Matériels nécessaire.

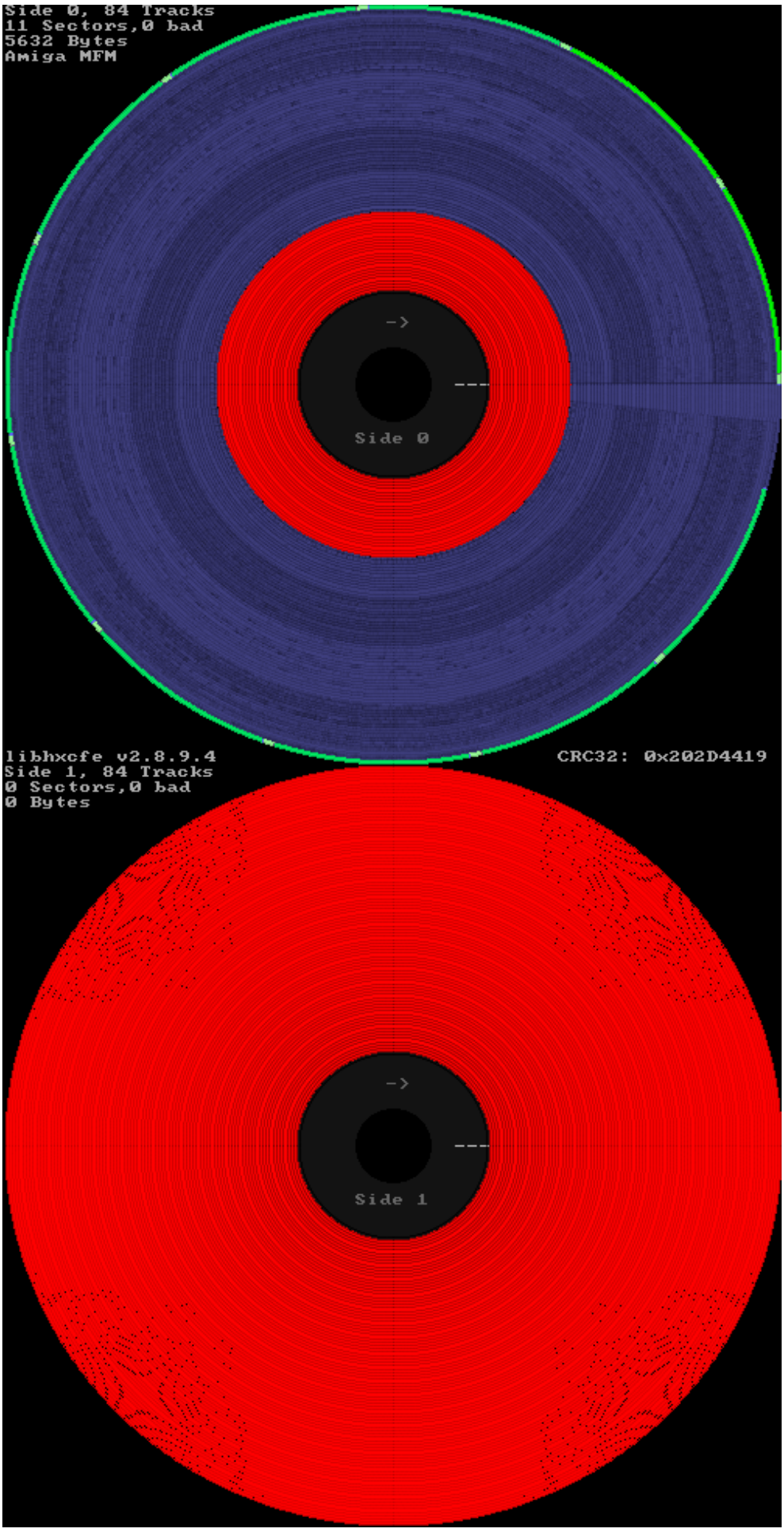
1) Un AMiGA avec 512K (ou plus) ou l'émulateur WINUAE
2) Une Carte ACTION REPLAY MKIII (ou sa ROM Image)
3) Le jeu Original IK+ ou son image CAPS (SPS 0339)
4) Le logiciel de compression 'Double Action' de VINCE of TRISTAR (exemple compilation : Anarchy-ScratchPack29)

Général Info :

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

Bon tuto.

Gi@nts

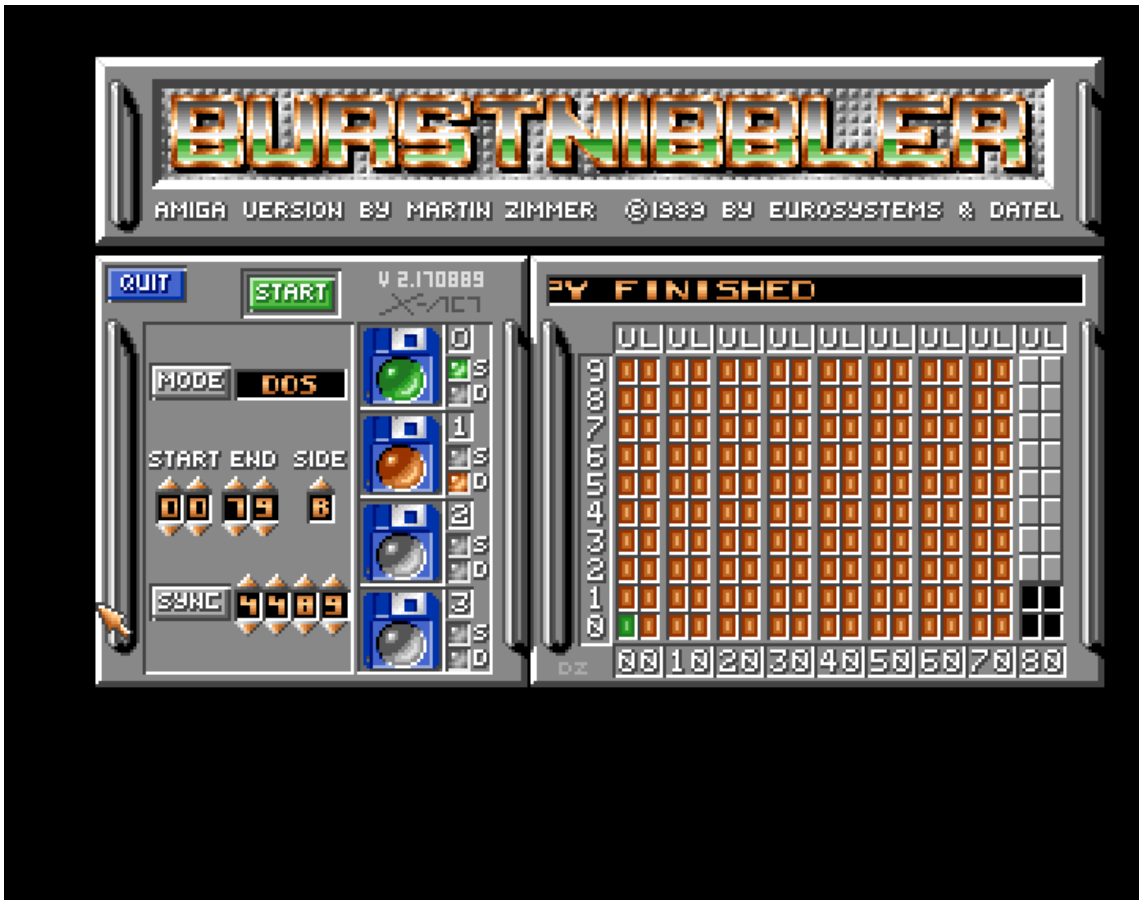

Side 0, 84 Tracks
11 Sectors,0 bad
5632 Bytes
Amiga MFM
->

Side 0
libhxcfe v2.8.9.4
Side 1, 84 Tracks
0 Sectors,0 bad
0 Bytes
CRC32: 0x202D4419
->

Side 1

Comme dans Tous bon hack qui se respecte, on va commencer par essayer de copier le disk original.
Entrer dans votre **AR**
et **Taper : burst**

Choisissez une unité de destination contenant une disquette vierge et lancer la copie.



Devinez quoi... Ce n'est pas vraiment copiable en l'état.

Commencer par charger la premier piste (et seulement **dos-lisible**) dans la mémoire pour voir ce qui se passe.

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

Entrez dans votre AR et tapez le texte suivant: **rt 0 1 70000** puis **m 70000**

```
**********************************************************************
                ACTION REPLAY AMIGA MK III
          (c) 1990/1991 by Olaf Boehm & Jörg Zanger
                (p) by Datel Electronics Ltd
**********************************************************************
No known virus in memory!
Ready.
rt 0 1 70000
Disk ok
m 70000
:070000 44 4F 53 00 EF 88 02 84 00 00 03 70 41 FA 00 18 DOS........pA...

d 7000c
~07000C LEA     70026(PC),A0
~070010 LEA     0007FA00,A1
~070016 MOVE.W  #FF,D1
~07001A MOVE.L  (A0)+,(A1)+
~07001C DBF     D1,0007001A
~070020 JMP     0007FA00
```

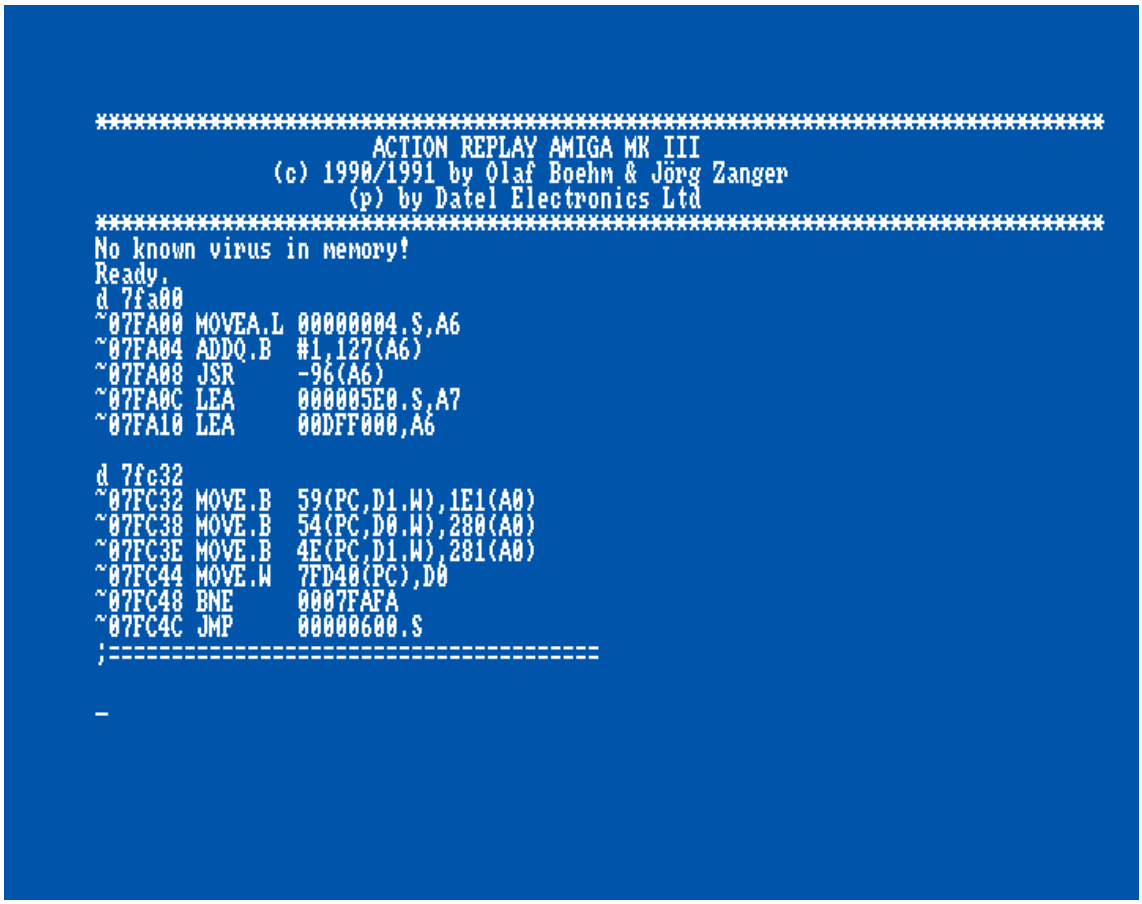
Comme le montre clairement l'image ci dessus, la 1er piste est standard avec un *header* '**DOS**'
Aller donc désassembler ce bootcode, juste après ce header :
#Désassemblage à partir de l'adresse mémoire 7000C
Taper d 7000c
Nous avons ici une petite routine de copie.
Elle copie le reste du bloc en **$7FA00** et l'exécute.

Effectuer un reboot de votre Amiga et laisser le jeu se charger, il semblerait qu'il charge toute les données d'une seul coup et démarre le jeu.

Ce qui serait intéressant pour nous, serait de trouver l'adresse du début du jeu en mémoire et de la sauver sur disquette.

Nous savons que le *bootcode* est déplacé en **$7FA00.**
Allons donc désassembler cette zone mémoire et chercher le code qui, après chargement, démarrer le jeu.

Rebooter encore une fois votre Amiga, laissez le jeu commencer à charger les données et **entrer** dans votre **AR**
#D alias désassemblage à partir de l'adresse 7fa00
Taper d 7fa00



Après quelques sous-routines, il semblerait qu'il se branche à l'adresse **$600**
Effectuez encore un reset de votre Amiga, laissez le jeu se charger un petit peu comme précédemment.
Nous allons insérez un *'breakpoint'* à l'adresse **$600**

Entrer dans votre **AR**
#BS permet, des que l'adresse mémoire $600 est atteinte, d'effectuer un arrêt du code.
#X permet le retour au code Amiga en court.
Taper bs 600 puis **X**

Une fois le breakpoint inséré et le retour au code normale, le jeu continue sont chargement.
Comme prévus, des l'adresse **$600** atteinte, notre **AR** prends la main.

Profitons en pour sauver tout ça sur disquette, insérez une disquette vierge dans df0
#SM, alias SaveMemory permet donc de sauver la zone mémoire de 600 à 7D000
Tapez : **SM a,6000 7D000**

Puis, rechargez le tout à l'adresse **$600** et executons ce code
#LM, alias LoqdMemory permet donc de charger un fichier à l'adresse mémoire $600
#G comme GO, démarre le code en $600 comme demandé.
Tapez : LM a,6000 puis **G 600**

```
lm a, 600
Loading from 000600 to 07D000
Disk ok
g 600_
```

Le jeu se lance.
Commencez une partie afin de tester si tous se passe bien.
Bien sur, cela aurait été trop beau, très rapidement le message **BUM COPY** apparaît à l'écran :

Suivi d'un crash complet du jeu.

Au bout de quelques essaies, il semblerait que ce message apparaisse tout le temps à la 26eme seconde du temps.

Il serait sympa de trouver ce compteur et voir si on peu faire des choses sympathiques avec.
Effectuez encore une fois un redémarrage de votre Amiga, entrez dans votre **AR**, recharger le jeu et exécuter le comme précédemment.

```
lm a, 600
Loading from 000600 to 07D000
Disk ok
g 600_
```

Comme précédemment, commencez une partie.
Quand le compteur **TIME** en haut à droite de l'écran, affiche **29**, **entrez** dans votre **AR**
#TS, alias Training Search, permet de commencer une recherche de 'trainer', pratique pour les compteurs.
Tapez : TS 29 puis **X**


```
Ready.
ts 29
First trainpass!
Change the countvalue next time!
Searched up to: C80000
Trainmode active!
Ready.
X_
```

Quand le compteur **TIME** en haut à droite de l'écran, affiche **28**, **entrez** dans votre **AR**
#T, alias Training, permet de continuer une recherche de 'trainer'.
Tapez : T 28 puis **X**

```
Ready.
t 28
Possible addresses:
0007DF
Searched up to: C80000
Trainmode active!
Ready.
```

Comme le montre l'image ci dessus, une adresse mémoire est rapidement trouvée, à savoir : **$7DF**

Allons chercher quelle adresse d'opcode fait appel à cette zone mémoire.
#FA, alias Find addressing opcode, permet de chercher une adresse de code modifiant une zone mémoire.
Tapez : FA 7DF

```
FA 7df
Search from: 000000 to: C80000
001172 CMPI.B   #26,000007DF.S
0011E4 MOVE.B   000007DF.S,D0
0011F2 MOVE.B   D0,000007DF.S
006012 MOVE.B   000007DF.S,D0
00606A MOVE.B   000007DF.S,D0
006078 MOVE.B   D0,000007DF.S
006200 MOVE.B   000007DF.S,D0
006310 MOVE.B   #30,000007DF.S
006BCA CMPI.B   #6,000007DF.S
006BEC MOVE.B   000007DF.S,D0
006BFC MOVE.B   000007DF.S,D0
_
```

La premier ligne à tendance à nous sauter aux yeux :
001172 CMPI.B #26,000007DF.S

Allons désassembler à partir de cette adresse mémoire.
Tapez : D 1172

```
d 1172
~001172 CMPI.B  #26,000007DF.S
~001178 BNE     00001194
~00117C LEA     FFFFFE4A.S,A0
~001180 TST.W   7D0(A0)
~001184 BEQ     00001194
~001188 MOVE.W  #C8,00000952.S
~00118E MOVE.W  #18,00000954.S
~001194 NOP
~001196 MOVE.B  00000608.S,D1
~00119A ANDI.W  #3,D1
~00119E BNE     000011A8
~0011A2 JSR     00006C8C
~0011A8 NOP
~0011AA NOP

?fffffe4a+7d0
%00000000000000000000011000011010 = $0000061A = !0000001562

m 61a
:00061A FF FF 03 33 0E EE 0E 6E 0C C0 00 0F 00 0F 00 00 ...3...n........
```

Voyons voir...

```
001172   CMPI.B   #26,000007DF.S   Compare si $7DF 'compteur TIME' = 26, si c'est le cas Z=1
001178   BNE      00001194         Est ce que Z=0 ? Si Z=0 alors on branche en 1194 => Boucle normal
00117C   LEA      FFFFFE4A.S,A0    Sinon on continue en positionnant FFFFFE4A dans A0
001800   TST.W    7D0(A0)          On test si l'adresse '7D0+A0' est égale à 0
                                   FFFFFE4A.S + 7D0 = $61A
                                   Donc est testé que l'adresse en $61A contient des 00, si c'est le cas Z=1

001184   BEQ      00001194         Si Z=1, alors on branche en $1194
001188   MOVE.W   #C8,00000952.S   Si c'est pas le cas.... on positionne C8 à l'adresse $952
00118E   MOVE.W   #C18,00000954.S  et C18 à l'adresse $954

001194   NOP                       ← Début mode 'normal', TIME ≠ 26
001196   MOVE.B   00000708.S,D1    Début de routine pour décompte compteur, aucune importance dans notre cas.
00119A   ANDI.W   #3,D1
00119E   BNE      000011A8
0011A2   JSR      00006C8C
```

Allons jetez un coup d'œil en **$61A**
Tapez : M 61A

Comme le montre l'image ci dessus, l'adresse mémoire **$61A** contient les valeurs **FF FF**

Voyons voir, si en modifiant les valeurs en **$61A** on arrive à éviter le crash
Tapez : M 61A
et **modifiez**

```
:0061A FF FF 03 33 0E EE 0E 6E 0C C0 00 0F 00 0F 00 00 ...3...n........
```

en

```
:0061A 00 00 03 33 0E EE 0E 6E 0C C0 00 0F 00 0F 00 00 ...3...n........
```

puis **Tapez : x**

Le compteur **TIME** dépasse le 26 et le jeu semble ne pas planter.
On pourrai se contenter de positionner ces valeurs et sauver le tout mais peu être existe-t-il une sécurité autre part !
On préféra donc rechercher l'adresse du code qui modifie cette valeur.

Redémarrez encore votre Amiga, **recharger** et **exécuter** le fichier comme précédemment,
Mais **1s** juste après avoir taper le **G 600**
Re-entrez dans votre **AR**

Vérifiez qu'il y a bien des **00** à l'adresse **$61A**
Taper : M 61A

Puis, positionnez un 'Match Point'.
#MS, alias Set Match Point, Active l'AR des que la zone mémoire en question est modifiée
Tapez : MS 61A puis **MW**
Puis : X

```
*******************************************************************************
                        ACTION REPLAY AMIGA MK III
                  (c) 1990/1991 by Olaf Boehm & Jörg Zanger
                        (p) by Datel Electronics Ltd
*******************************************************************************
No known virus in memory!
Ready.
lm a,600
Loading from 000600 to 07D000
Disk ok
g 600
No known virus in memory!
Ready.
m 61a
:00061A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................

ms 61a
Memwatchpoint set
Ready.
mw
List of memwatchpoints: 00061A
Ready.
x_
```

Attendez quelques secondes que votre **AR** reprenne la main.

```
Ready.
Memorybyte 00061A has changed from $00 to $FF
```

Et désassemblons ce bout de code en prenant le peine de remonter un peu en adresse mémoire afin de mieux comprendre ce qui se passe.
Tapez : D

```
~024806 ADDQ.W  #2,A0
~024808 AND.L   D1,D4
~02480A CMP.L   D0,D4
~02480C DBEQ    D2,00024804
~024810 BEQ     0002482E
~024812 LSR.L   #1,D0
~024814 ROR.L   #1,D1
~024816 ADDQ.W  #1,D3
~024818 CMP.W   #10,D3
~02481C BNE     000247FA
~02481E LEA     00000962.S,A0
~024822 MOVE.W  #FFFF,-348(A0)
~024828 BSR     00024878
~02482C RTS
;=====================================


?962-348
%00000000000000000000011000011010 = $0000061A = !0000001562
_
```

Encore une fois, la commande située en **$24822** saute au yeux.
En effet, l'instruction en **$2481E** place la valeur 962 en **A0**
Est placé ensuite en **$24822** les valeurs **#FFFF** à l'adresse mémoire de **A0**-348 (donc 962-348)
à savoir : **$61A**

BINGO, il ne reste plus qu'à changer ces deux valeurs par **00 00**
Tapez : M 24822
et **modifier**

```
:024822  31 7C FF FF FC B8 61 00 00 4E 4E 75 59 48 28 10      1|..ü.a..NNuYH(.
```

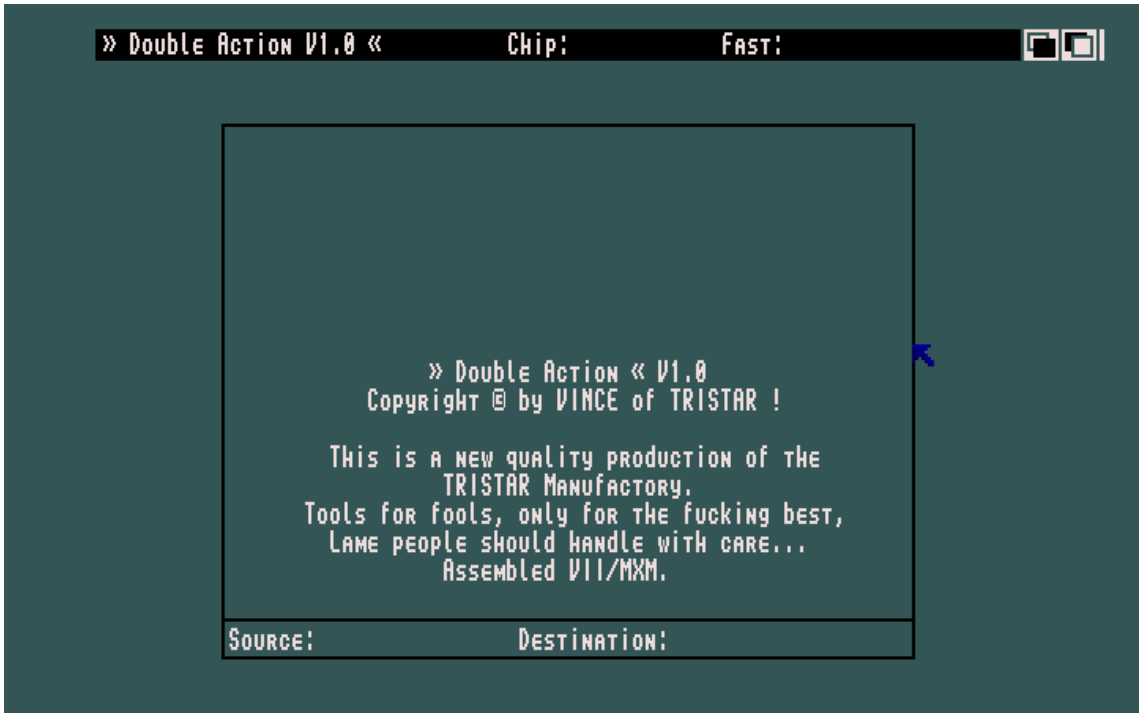
en

```
:024822  31 7C 00 00 FC B8 61 00 00 4E 4E 75 59 48 28 10      1|..ü.a..NNuYH(.
:024832  C8 81 20 3C 24 44 00 00 E6 A8 B8 00 66 DE 41 F8      .. <$D......f.A.
```

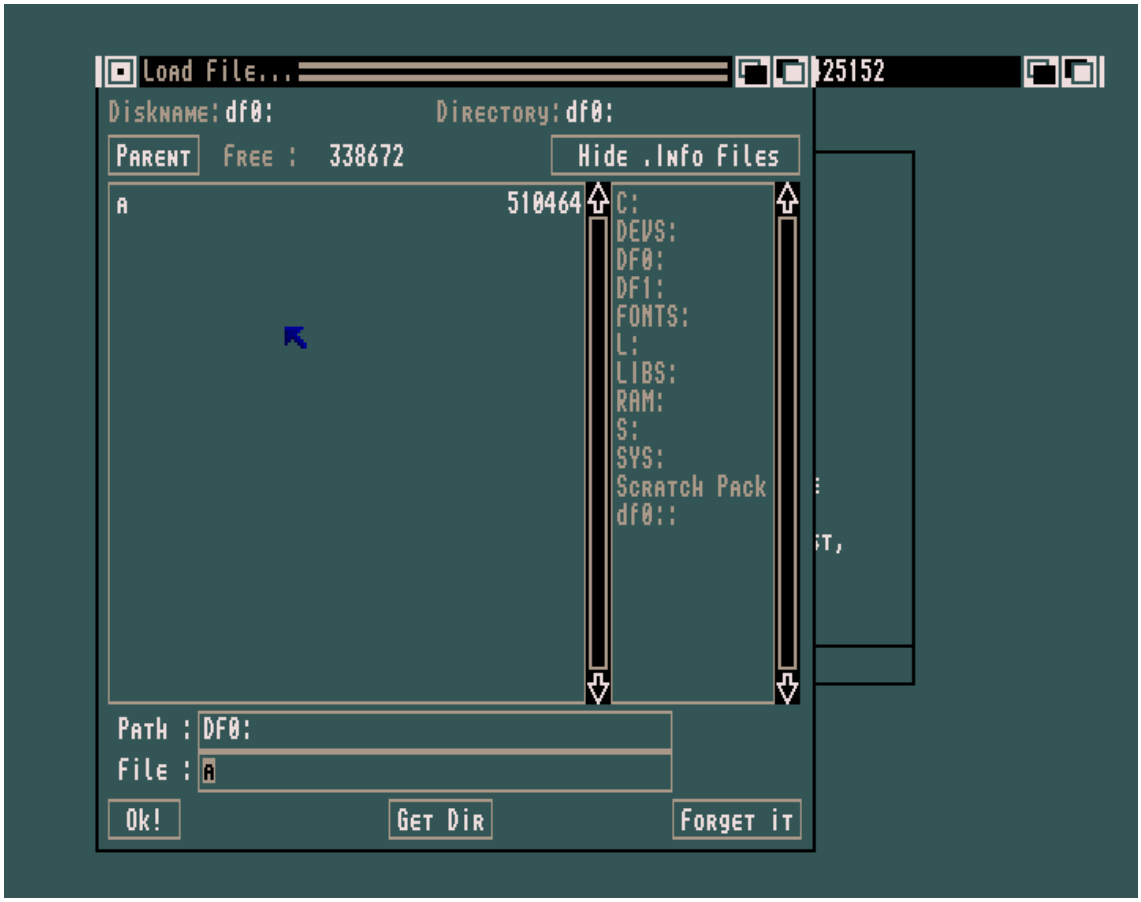
Sauvons maintenant le tout
Tapez : SM A,600 7D000

```
m 24822
:024822 31 7C FF FF FC B8 61 00 00 4E 4E 75 59 48 28 10 1|..ü.a..NNuYH(.

m 24822
:024822 31 7C 00 00 FC B8 61 00 00 4E 4E 75 59 48 28 10 1|..ü.a..NNuYH(.
:024832 C8 81 20 3C 24 44 00 00 E6 A8 B8 00 66 DE 41 F8 .. <$D......f.A.

sm a, 600 7d000
```

Il ne reste maintenant plus qu'à compresser le fichier sauvé et le rendre exécutable.
Insérez dans l'Amiga la disquette contenant le logiciel de compression 'Double Action' de VINCE of TRISTAR
Redémarrez et lancer cet outils.

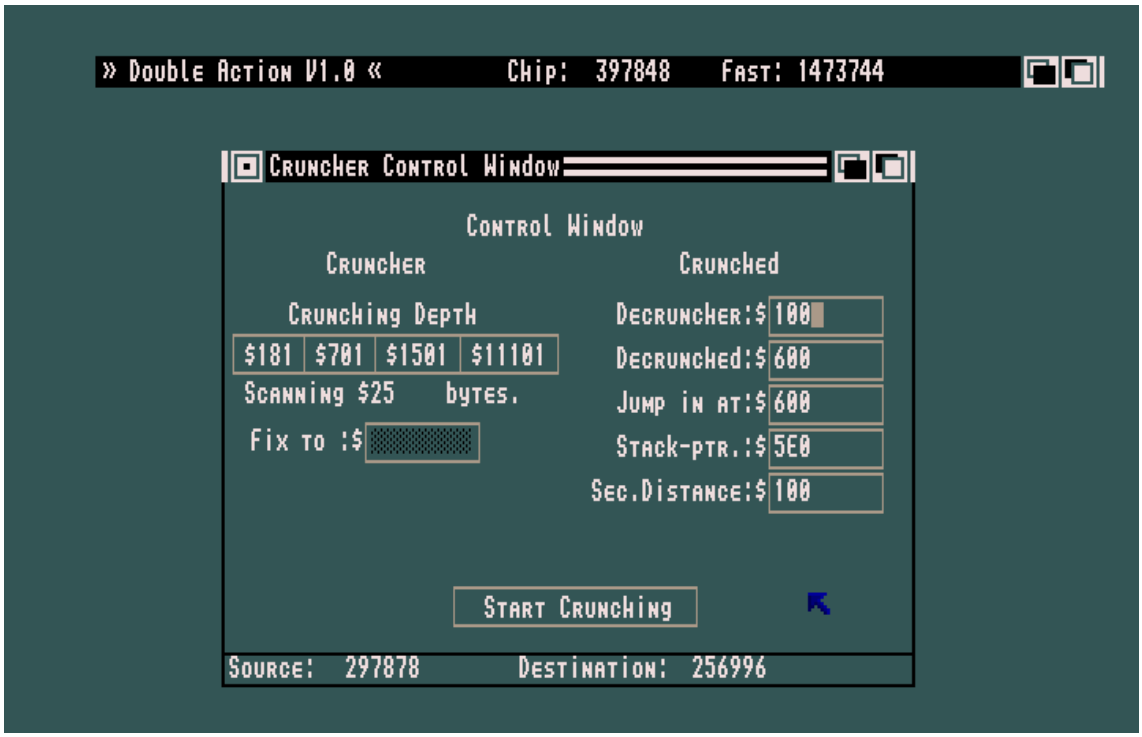


Comme indiqué sur les images prises, chargez votre fichier préalablement sauvé :
PROJEKT => LOAD FILE => a



Concernant les valeurs de compression :

```
Crunching depth   : $25    # meilleur compromis
Decruncher        : $100   # Le jeu n'utilise pas cette zone mémoire
Decrunched        : $600   # exactement comme le jeu original
Jump in AT        : $600   # Adresse de départ du jeu
Stack             : $5E0   # Juste au dessus du code, le jeu original utilise d'ailleurs cette valeur, voir 3eme image de ce tuto
Sec distance      : $100
```



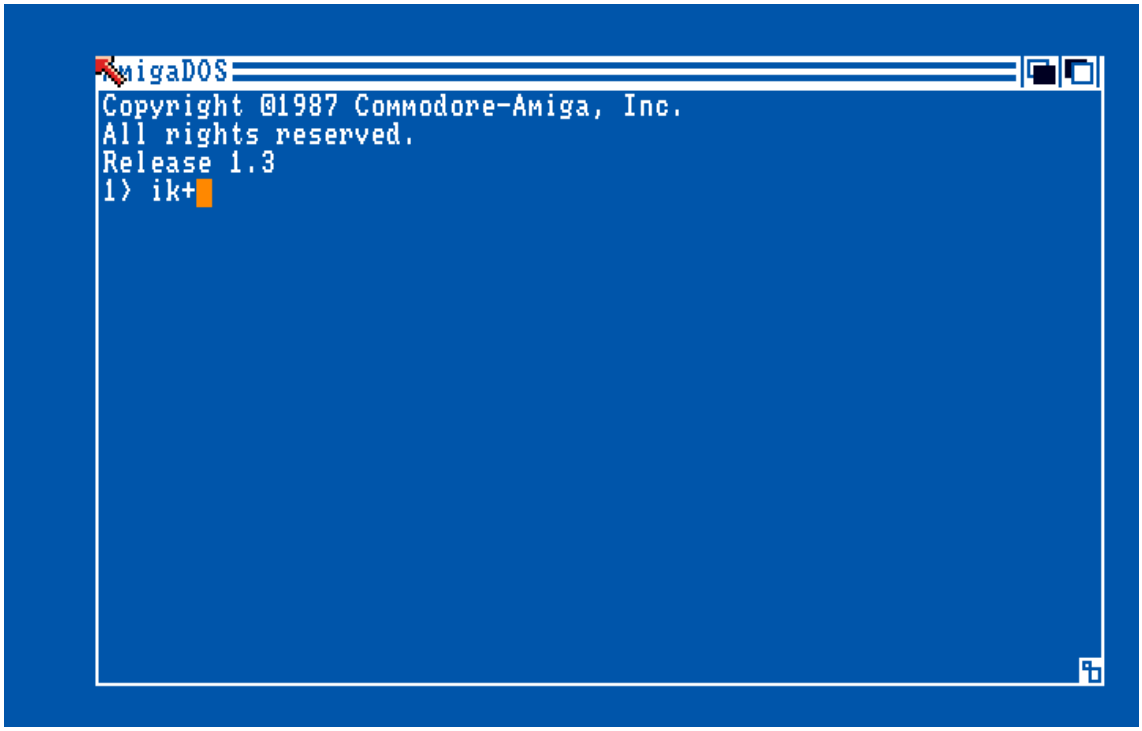
Puis **CLIQUEZ** sur **START CRUNCHING**

La compression est en court …

Une fois terminée, sauvez le tout sur une nouvelle disquette vierge par exemple :
PROJEKT => SAVE FILE => ik+

Il ne reste plus qu'a rendre la disquette bootable, **entrez** dans votre **AR**
#INSTALL, alias BOOTBLOCK INSTALL, permet d'installer un secteur de boot sur l'unité indiquée, 0=DF0, 1=DF1
et **Tapez : INSTALL 0**

Effectuer un **reset** de votre Amiga et à l'invite de commande, **tapez** le **nom du fichier** préalablement sauvé, à savoir **ik+**



Après une phase de décompression, le jeu se lance.