DOCUMENTATION FOR

SKATE OF THE ART

A skateboard Arcadegame for the Commodore Amiga

1001 Software Developments
Erik van Eykelen
8-20-1988

Skate of the Art - Documentation

1001 Software Developments presents:
Skate of the Art, a horizontally scrolling skateboard game.
Programming: Erik van Eykelen
Graphics: Joost Honig
Music: Frank van Eykelen
Technical advice: Steve van der Horst
Distributor: Linel Switzerland, M. Grimmer

The History of Skate of the Art:

Erik van Eykelen and Joost Honig started writing the game during the Christmas holiday of 1987 (late December). Joost had written RollerBoard on the Commodore 64 and the main ideas and designs of the '64 version can be found in the Amiga version. Because we didn't have a deadline in the beginning we have included many new features that have enhanced the game. But as we didn't have to make a perfect conversion we have left out some ideas of the '64 version that we didn't like.
The Amiga version has been renamed because of problems with selling the `64 version. No legal problems are to be expected because the company which bought RollerBoard has not met the conditions as stated in the contract.
We have finished the game on August 20th 1988. We are very pleased about the final look of the game, it's a good game and many of our (very critical) friends like it a lot. We like to thank them for their support.

The Programming:

The programming of Skate of the Art (from now on referred to as `the game') was for the biggest part done by Erik. About 90 percent of the coding is written in 'C'. I have chosen C because programming on the Amiga consists chiefly of calling Operating System routines (KickStart) and filling pieces of memory (structures) with the right values and putting the Blitter and Copper (co-processors) to work. C is the perfect language for this kind of programming. C allows the programmer to access the system with the desired values in bits, bytes, words or longs, but C is also a real high-level programming language.
The 'body' of the program is in C and only some time-critical routines like playing the sound track, starting and stopping the samples, reading the joystick, generating a random number and decompressing data have been written in 68000 machine-code. The Manx Aztec C compiler allows to include assembly-coding into C source. The mixture of C and assembler also applies to the bootsystem, leader, intro and joystick information program. Although C compilers tend to produce not the most efficient code the game feels smooth and robust. I have been very careful in not allocating and freeing memory during run-time of the game. The game seems bug free, but in about 180 kB of source code a small error can easily be overlooked...

The Graphics:

The main attraction of the game is its beautiful graphics. Joost has put many hours in designing the graphics and has done a great job. Everything that is visible has been drawn. No system fonts are used, no lines are drawn or dots are plotted during run-time. Really every pixel has been put down by Joost. And saved to disk. That's why we decided to compress all the graphics and sprite data. When the game needs data from the disk, it loads the compressed data and decompresses it in less than 2 or 3 seconds.
Most of the pictures are in 32 colors. This consumes huge amounts of memory but the results are quite astonishing. Throughout the whole game we have carefully selected the right colors, the right backgrounds and the best images. We hope that the buyers of the game notice our eye for detail and appreciate the smooth and detailed look.

The Music

The game has two soundtracks, one is played during the intro, the other during the game. The music has been composed by Frank van Eykelen using SoundTracker from Karsten Obarski. The playroutine and samples have been linked into one file. The music is started by calling Execute(), a DOS library function. This function loads the music file and starts it automatically. The music routine hangs itself into the vertical blank interrupt. Execute() requires the Amiga DOS command 'run' to be in the 'c' directory. The samples during the game were created with Steve vanderHorst's synthesizer. These sounds were digitized by our custom made sound digitizer.

The Tools

The levels have been created with the 1001 level editor, written by Steve. All graphics have been designed in DeluxePaint II PAL.
The sprites are drawn in DPaint and converted to attached sprite data using the 1001 sprite converter, written by Erik. All the IFF data is converted into raw format using the 1001 IFF to RAW converter, written by Erik. Graphics and sprite data has been crunched by the 1001 card-cruncher.

The Disk

```
c      --> run                                      ; needed to start music

sys    --> boot.exe                                 ; started in startup-
                                                  ; sequence. loads all the
                                                  ; programs
         r-ror                                    ; gives pseudo guru error
         r-or.cru                                 ; if two or more drives are
         r-or.col                                 ; connected
         read-me

l      --> Disk-Validator

leader --> l.exe                                  ; leader program
         b12.cru                                  ; crunched pictures
         b1.col                                   ; IFF color file
         b2.col                                   ; see b1.col

s     --> startup-sequence                        ; boots sys/boot.exe
```

```
intro  --> s.exe                           ; intro program
         m1.exe                            ; intro music program
         f.cru                             ; font picture
         f.col                             ; IFF color file
         i1.cru                            ; crunched picture
         i1.col                            ; IFF color file
         i2.cru
         i2.col
         i3.cru
         i3.col
         if.asc                            ; intro text in ASCII

joy    --> ji.exe                          ; joyinfo program
         ji.cru                            ; crunched picture
         ji.col                            ; IFF color file

game   --> sk.exe                          ; game program
         highscores                        ; highscore file

           lvl1234 --> level1234.cru       ; crunched brush picture
                     bg1234.cru            ; crunched background
                                           ; picture
                     level1                ; 1001 level editor file
                     level2
                     level3
                     level4
                     com1                  ; command code file
                     com2
                     com3
                     com4

           lvl5678 --> level5678.cru
                     bg5678.cru
                     level5
                     level6
                     level7
                     level8
                     com5
                     com6
                     com7
                     com8

           lvl9abc --> level9abc.cru
                     bg9abc.cru
                     level9
                     levela
                     levelb
                     levelc
                     com9
                     coma
                     comb
                     comc
           lvldefg --> leveldefg.cru
                     bgdefg.cru
                     leveld
                     levele
                     levelf
                     levelg
                     comd
                     come
                     comf
                     comg

           lvlhijk --> levelhijk.cru
                     levelh
                     leveli
                     levelj
                     levelk
                     comh
                     comi
                     comj
                     comk

         all     --> m2.exe                ; game music program
                     spd1.cru              ; crunched sprite data
                     bd.cru                ; board picture
                     bd.col                ; IFF color file
                     bd1.cru               ; board picture, used in
                     bd2.cru               ; falling animation sequence
                     bd3.cru
                     bd4.cru
                     bd5.cru
                     bd6.cru
                     bd7.cru
                     sm1.raw               ; sample data in raw format
                     sm2.raw
                     sm3.raw

         bonus   --> spd2.cru              ; crunched bonus sprite data
                     b-all.cru             ; ramp, pool and lower
                                           ; picture
                     b-down.col            ; IFF color file
                     b-mid-1.cru           ; middle picture
                     b-mid-1.col           ; IFF color file
                     b-mid-2.cru
                     b-mid-2.col
                     b-mid-3.cru
                     b-mid-3.col
                     b1com                 ; command code file
                     b2com
                     b3com

system-configuration
disk.info
.info
```

The Game Instructions

- Switch off your Amiga.
- Disconnect all external drives.
- Insert disk in internal drive (df0:), turn the Amiga on.
- The program bootsys is loaded. After it has been loaded and started the system LED goes off for a few seconds. Pressing ESC during this pause will skip the leader and the intro (and joyinfo).
- Now the 1001 Software Developments logo appears. Press the fire button, the left mouse button or wait.
- The intro is loaded. It can be terminated by pressing the firebutton or the left mouse button.
- Now you are presented with a menu. By pushing or pulling the joystick north or south you can select to load the game immediately or to obtain joystick information first. The joystick info lets you train the jumps, flips and other movements.
- Joystick info option:
  - FIRE increase speed until speed is 2. When speed is 2 (only visible in game because the scrolling is faster) no further speed can be made. In the bonus games there are 4 speeds available. When speed is 0, no jumps can be made, so first press fire once or twice in the joystick info and game.
  - NORTH jump.
  - SOUTH brake, slow down to speed 1.
  - WEST position to skate up a hill.
  - FIRE & WEST double flip, possible in speed 1 and 2.
  - FIRE & NORTH single flip, possible in speed 1 and 2.
  - FIRE & EAST super flip, only possible in speed 1. speed is cranked up to speed 2.
  - F1 exits joystick info and loads game.
- Game options: F1 pause mode, pressing F1 again continues the game. Pressing in the pause mode F2 two times, F3 three times, F4 four times and F5 five times activates the cheat mode. You must press the keys not too fast and if the green led doesn't turn red, press F1 to escape from the pause mode. To retry to switch on the cheat mode press F1 again. It takes a few tries at first but finally the led becomes red. F10 takes you to the next level and you are given 99 boards. Enough for even the Atari owners.

  F2 slow motion mode, only flip scores are counted.

  F3 game over

When the game starts, text appears at the lower screen. First the name of the game and the authors, then a screen with the function key options, then the highscore list and then the game screen (a skateboard on which the score is printed). When you don't press fire and wait until the board appears for the second time, the demo mode is activated. This mode shows several levels. This repeats after a certain time.

The Score

If you make a single flip (FIRE & UP) 50 points are added to your score, a double flip (FIRE & WEST) adds 100 points and a super flip gives you an extra 150 points. If the total amount of flip points earned during one level is greater then or equal to 300 then a flip bonus is given of 1000 points and this is shown to you when you have reached the end of the level.
If you skate four fifths of the track at speed 2, a time bonus is given of 1000 points. This is also shown to the player when he has completed a level.

If you earn 10000 points you are offered to buy a skateboard. Press B to buy a board (10000 points will be subtracted from your score) or press fire to turn down the offer. At 20000 points this is repeated. It is your choice to get into the highscores and to be out of boards quicker or to play further with the two extra boards.

The Levels

levels 1-4 : The Holland Levels, 4 fairly easy levels.

bonuslevel 1: The Balance Game: at speed 1 you must keep the skater in balance by pushing the joystick into the opposite direction to which the skater hangs. Before you try to jump over the swimming pool accelerate to speed 4. Land by pulling the joystick south. If you make it (you may try it three times) you get 10000 points. Falling during the bonus games doesn't deduct boards from your supply.

levels 5-8: The Miami Levels, the levels are getting more and more difficult.

bonuslevel 2: The Hurdle Game: jump over the hurdles by pushing the joystick to the west. The skater jumps into the air but his board rides under the hurdle. A very neat jump, loved by the genuine skaters!

levels 9-12: The Metro Levels.

bonuslevel 3: The Big Jump Level. The only way to make it is speeding up to speed 4 and pushing the joystick to the west. A giant 'ollie' (jump) is made and you jump through the gap in the wall.

levels 13-16: The City Levels.

bonuslevel 1: the same as after level 4.

levels 17-20: The SkatePark Levels. A new feature is added in these levels. Some surfaces are white (white horizontal lines). You are not allowed to step (make speed) or jump on these lines.

The HighScore Table

When your score is sufficient to be included, you are allowed to enter your name. A name can be 8 characters long. Pushing the joystick to the north and south lets you select the desired character, pushing to the west and east lets you choose the right position in the name and fire adds the name to the list. The list is sorted on the score, the highest score is at the top of the list. The list has five entries. If the disk is write-protected the highscore is not saved, if the disk is write enabled the score is saved and loaded the next time the game is played. Pressing the DEL key when entering the name, erases the highscore table.

The Programs

bootsys.c : This program loads 'df0:leader/l.exe' and executes it. Then it executes 'df0:intro/s.exe;'. After this it deallocates all the memory it had allocated and executes the game: 'df0:game/sk.exe'.

leader.c : This program loads some graphics and two colormaps. It shows the rotating 1001 Software Developments logo and the game picture.

intro.c : This program shows the Linel logo, two pictures of skaters and prints some text on the screen. The text can be found on the disk as an ASCII file (see The Disk). After pressing fire or the left mouse button you can select to run the game or to see the joystick info first.

joyinfo.c : The joystick info program. All the game animations are shown, including a joystick which mimics the user's actions. A very useful addition.

skate.c : The actual game.

The Programming Environment

The source files are written in TxEd. This editor can be found on the Aztec C disk. The programs can be compiled in the following way:

```
boot the Aztec disk
type: execute diskcomp
insert the source disk in df1:
type: cc df1:sourcefile
the file is being compiled and assembled
type: ln df1:sourcefile.o +ccdb -lc
```

Some sourcefiles need extra options when being compiled. They can be found in the headers of the sourcefiles.

The End

The authors, Erik van Eykelen and Joost Honig, hope that the buyers of Skate of the Art will enjoy the game for a long period of time.