

CARTOON CAPERS



Programming, Graphics and Sound by Simon Cook.

Written in STOS Basic

© Mandarin Software 1990

Merrie Melodies tune by Cantor Tobias and Mencher

© Warner / Chappell Music

Write to Mandarin Software for help with defective discs or other initial problems at:
Customer Services, Mandarin Software, Europa House, Adlington Park, Adlington,
Macclesfield, Cheshire SK10 4NP.

The Story

Inside a rickety, forgotten old movie studio somewhere on the edge of Hollywood, a film crew gets ready to shoot the next scene in an epic drama...

Well, maybe not quite an epic. Perhaps not even drama. In fact, this is the set for the latest cartoon caper starring those lovable heroes, Judo Jake and Karate Kat. The film is entitled "Paw To Claw Combat", and follows a 'completely original' cartoon plot involving never-ending acts of animated splats, crashes, ker-pows and boings. There are four different sets on which the 16 different scenes must be performed, with many crazy traps and props to be used.

Your job as a starring actor in the film is to take the part of Judo Jake and ensure that he performs his job well. After all, the Oscar nominations for Best Animated Comedy are still to be announced. As a measure of his performance, damage meters are mounted above the set. These record the amount of damage that Jake has inflicted upon his co-star, Karate Kat. As the hero of the film, Jake must ensure that he inflicts more damage to the Kat than he sustains himself in the 60 seconds allowed per scene. This damage may be caused directly by landing a well-timed flying kick or punch, or may be caused indirectly by luring the Kat into one of the many traps occurring on the film sets. But watch out, as Jake himself could be caught in the traps very easily, causing unnecessary damage to himself. In the case of equal damage being scored by both characters then the scene will be reshot.

Movements

Characters are controlled by the joystick. The eight directions in which the joystick may be moved, combined with whether the fire button is held down or not, give 16 possible movements which may be performed. To make things more difficult, the script dictates that different moves are performed on different film sets, giving a total of four different groups of moves, with only one set available in each scene. Somehow, Jake forgot to read his script last night (maybe it has something to do with an all-night party at a very famous rabbit's Hollywood mansion!) and therefore he doesn't know what the moves are.

Rehearsals

Fortunately for Jake, the director will allow him a 30-second rehearsal at the beginning of a scene in which to learn his moves. However the director only allows Jake this privilege in the first four scenes, so the moves must be remembered after that (Hint: There is a pattern to the order in which groups of moves appear).

Props and traps

Different moves are also available with the various traps and props in the film. These cannot be rehearsed and must be ad-libbed. The director only allows one take per scene and so Jake must learn quickly and improvise well. (Hint: By observing Karate Kat's actions you can often gain an insight into how you are expected to use, avoid or collect the many traps and props).

Time bonuses

As filming is very expensive, the faster a scene is completed, the cheaper it is to produce, and the happier the producer is. Therefore Judo Jake will be awarded bonuses for early completion of a scene. A special bonus is awarded if he does it in half the 60 seconds allowed.

Controlling Karate Kat

Normally, Karate Kat is controlled by the computer. However a second player may take control of him using the second joystick. In a two-player game, both players must score more than four points of damage to their co-star to advance to the next scene. If four points are not scored then the computer takes over control of the losing player.

Starting the game

To load the game, insert the disc and turn on the computer. Once loaded, a one-player game is started by pressing the fire button on joystick one from the title screen. A two-player game is started by pressing the fire button on joystick two.

During the game it may be necessary to swap discs. Please press fire after doing so. Also during the game several keys may be pressed:

F1 - Restarts a one-player game.

F2 - Restarts a two-player game.

F3 - Pauses the game and the fire button unpauses the game.

STOS Double Joysticks

The two player option offered by Cartoon Capers is made possible by the use of Bobby Earl's Double Joystick Extension, and we've included it on Disc One.

To use the STOS Double Joystick Extension within your own STOS Basic programs, you must first install it onto your STOS language disc:

- Load STOS Basic.
- Insert Cartoon Capers Disc One into drive A.
- Load INSTJOYS.BAS, found in the root directory.
- Remove Cartoon Capers Disc One from drive A and replace it with the STOS Basic disc on which you wish to install the extension.
- RUN the INSTJOYS.BAS program and follow the on-screen prompts.
- Re-boot STOS so that the extension loads up and installs itself.

If you have a Compiler disc then you can install the relevant extension as well.

Your STOS Basic is now able to monitor the actions of two joysticks simultaneously.

The commands in the new extension are as follows:

STICKS ON/OFF *(Set / reset double joystick mode)*

STICKS ON

Activates the Double Joystick extension.

STICKS OFF

Deactivates the Double Joystick extension.

=STICK1 *(Return joystick states)*

=STICK2

s1=STICK1

s2=STICK2

These are variables, the values returned being determined by the current positions of joysticks 1 and 2 respectively. For example:

```
10 sticks on
20 A = STICK1
30 if A <> 0 then print "Stick 1 = ";A else goto 20
40 goto 20
```

With a little knowledge of binary mathematics you can easily calculate the position of each joystick and its fire button.

The following table illustrates the status of each bit for the basic four joystick positions and the state of the fire button:

```
Bit 0 = Up
Bit 1 = Down
Bit 2 = Left
Bit 3 = Right
Bit 4 = Not Used
Bit 5 = Not Used
Bit 6 = Not Used
Bit 7 = Fire Button
```

Here are some examples of values returned by the STICK variables and the positions which produced them:

POSITION	STICK value
Up	1
Down	2
Left	4
Right	8
Fire	128
Up + Left	5
Fire + Down	130
Fire + Down + Right	138

If you are still unsure about how to interpret the values returned by these STICK variables, try loading and running the second joystick program that we've provided on the Cartoon Capers Disc One (JOY_TEST.BAS). With this program running all you need to do is move the joystick to the desired position and then make a note of the value displayed.

STOS Squasher

The large amount of graphics data used by Cartoon Capers has been compacted to a fraction of its size using the amazing STOS Squasher utility. This utility is included with Games Galore – a collection of four addictive STOS games retailing for £19.95

Cartoon Capers and STOS

Cartoon Capers was written using Mandarin's *STOS – The Game Creator*, the flexible yet powerful programming package. This game was the winning entry in the STOS 1989 £5,000 Gameswriter of the year competition.

The more inquisitive among you can lay your hands on a fully annotated listing of the program (on disc) by sending the token that is attached to these instructions and a cheque or postal order for £2.00 to the STOS Public Domain Library at the following address:

Sandra Sharkey, 78 Merton Road, Highfield, Wigan WN3 6AT.

New Ideas

Cartoon Capers is just one, high-quality example of what you can produce using STOS. If you are an ardent STOSer who has written, or is currently writing, an original game, then we would like to hear from you. Alternatively you can send potential public domain titles to Sandra Sharkey at the above address.

We realise that original ideas are not the sole province of the computer whiz kid, so if you are the type of individual that couldn't program his way out of a paper bag, but has three flashes of genius before lunch, then please drop us a line. In the software business originality is worth its weight in gold – literally.