

DogFight!

Sean E. Russell

COLLABORATORS

	<i>TITLE :</i> DogFight!		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Sean E. Russell	August 3, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	DogFight!	1
1.1	Index	1
1.2	General Information	1
1.3	Installation	1
1.4	Controlling DogFight!	3
1.5	The Preferences Window	3
1.6	Controlling DogFight!	4
1.7	Legal Info	4
1.8	Bug lists	4
1.9	About the Author	5
1.10	History & ToDo list	5
1.11	Baking Powder!	6
1.12	Even MORE people I'm in debt to...	7
1.13	Magic User Interface	7

Chapter 1

DogFight!

1.1 Index

DogFight! v1.22

21-May-94

by Sean Russell

MCMXCIV Apocalypse Productions

[General intro](#)

[Installation](#) System requirements

[Rules](#) Playing the game (Rules)

[Legal crap](#)

[Bug list](#)

[The author](#) and how to contact him

[History](#) and future

[The Sourcecode](#) (or: "Oh... that's ASCII??")

[Credits](#) (or: Gee, like, thanx, dude.)

[MUI](#) It'll change your life!

1.2 General Information

Back in the old days (when 48K of RAM was a lot of memory) there was a game for the Apple][called "DogFight". One of the simplest games I know, it was also possibly the most fun, due to it's amost unique ability to allow up to 7 people to simultaneously play.

I'm not going to claim that this program is DogFight reincarnate. However, DogFight! does share some similarities with the "original" game. Anyone who remembers the old DogFight will notice that several features are missing in this version; look for future releases of DogFight! for improvements. In addition, DogFight! contains some options the original doesn't.

1.3 Installation

This distribution contains the following files:

Directory of "Eris:MyProgs/DogFight!" on 07-Apr-94 16:50

```

----rwx Today 16:23 50 25.076 DogFight!
----rwx Mar 21 94 17:52 4 1.256 DogFight!.dockbrush4
----rwx Today 16:48 39 19.562 DogFight!.guide
----rw-d Mar 21 94 18:00 11 4.694 DogFight!.guide.info
----rw-d Mar 21 94 18:00 9 4.003 DogFight!.info
----rwx Today 16:23 23 10.876 Sounds.8SVX
----rwx Today 16:25 37 18.268 Sources.lha

```

7 files - 0 dirs - 0 links - 173 blocks - 83345 bytes

The CRCs for the executables are:

CRC File

\$B3BEE7A1 DogFight!

\$F12B40A6 ms

The Sources.lha file contains:

Listing of archive 'Sources.lha':

Original Packed Ratio Date Time Name

```

-----
42571 8719 79.5% 21-May-94 14:15:34 DogFight!_1.0.cpp
4372 1801 58.8% 16-May-94 15:08:40 DogFight!_1.0.h
18318 4272 76.6% 21-May-94 14:13:34 DogFight!_class9.cpp
7204 2056 71.4% 21-May-94 13:01:36 DogFight!_class9.h
1080 677 37.3% 16-May-94 15:08:42 ms
1201 570 52.5% 21-May-94 14:02:14 ms.c
891 417 53.1% 16-May-94 15:08:40 snd.h
9235 1922 79.1% 21-May-94 14:02:14 sound.c
160 88 45.0% 21-May-94 13:46:26 sound.h
364 195 46.4% 16-May-94 15:19:18 smakefile
6778 5797 14.4% 16-May-94 15:08:40 explode.8svx
2890 2023 30.0% 16-May-94 15:08:40 shoot.8svx
1196 987 17.4% 16-May-94 15:08:40 start.8svx

```

```

-----
96260 29524 69.3% 21-May-94 14:41:46 13 files

```

You must have MUIMaster.library version 7.973 in your LIBS: directory for DogFight! to work. This library is part of the **MUI** package, and is not supplied with DogFight!, on request of the author of MUI.

You need at least intuition.library v39 since I use those doublebuffering routines.

The file sounds.8svx must be in the same directory as the main program.

See the **Bugs** section for incompatibilities and troubleshooting.

Double-click on the icon (or start DogFight! from a CLI) and off you go.

As of version 1.2, you may now set your own sounds for the game. To do this, extract the file ms from the sources archive. Take your 8SVX files which you want as your sounds and rename them "explosion.8svx", "start.8svx", and "shoot.8svx". Make a backup copy of your original Sounds.8svx file, and run the ms program. The next time that you run DogFight!, your new sounds will be used. I make no guarantees that your sounds will work with DogFight!. However, if you keep the sounds reasonably short and have enough memory for them, there should be no problems. Important is that the three sound files are in standard IFF-8SVX format.

1.4 Controlling DogFight!

In this version of DogFight, from 0 to 4 people may play at one time.

The Start

First you will get a **MUI preferences** window. At this point you should consider changing the screen mode if you do not have an AA-chipset Amiga, or if you desire a screen other than the default mode (Double-PAL hires 640x512).

You may then set your game options. After every game this preferences window will open, at which time you may change the options or quit the program.

To start the game press the Go! button in the **preferences** window. The prefs window will close and the **game** screen will come forward.

To quit the game, press the Escape key if the game screen is in front, and press the Quit button from the **preferences** window.

1.5 The Preferences Window

Player Pages

The MUI control window allows the user to configure the game. In each player page (labeled 1-4) is a list of options and a short text describing the controls for that Jet.

In every player page is a cycle gadget with three entries:

Not in play - If this entry is chosen, this jet will not appear in the following game.

Human - This entry means that this jet will be controlled by a human.

Hector - This jet is computer controlled.

Each jet can be set to have certain characteristics:

Compromise - reasonably fast, reasonably maneuverable.

Speed - A fast Jet, but not very maneuverable

Maneuverability - Slow but with a tight turn radius

General Gadgets

In the prefs window are also two checkboxes. If the Fast Game box is checked, the game will run at high speed. If the Safe Start box is checked, jets may not shoot and may not be shot until the player turns the Jet for the first time. This "invulnerability" is in effect after each time the jet is shot down and then reappears.

There are also three buttons:

Go! - starts the game

Save - saves the settings so that next time you play DogFight! the Jets will have the same characteristics.

Quit - ends the game.

The Save button will have no effect if you do not have a registered version of MUI.

NOTE:

The Save button does not (yet) save your screenmode preference. You must set your preferred screenmode (if you are unsatisfied with the default) every time you run the program.

Menus

The prefs window has a menu with four options:

About - gives you the usual version number, etc;

Screen - opens the screenmode requester

Save - does the same as the Save button

Quit - does the same as the Quit button.

The ScreenMode requester

When you choose the Screen option from the menu, you will get a screenmode requester allowing you to choose on what type of screen DogFight! runs. If you have asl.library v39 or greater, you'll get a system requester which should allow you to choose any screen mode that supports double buffering. If you don't have WB2.1(+), you'll get a MUI requester with 3 options: LoRes, Hires PAL Laced, and HiRes NTSC Laced. Choose one that is suitable to your computer. Note that some screenmodes do not support doublebuffering. Be especially wary of graphic-cards screenmodes which appear in the requester.

1.6 Controlling DogFight!

Flying the Jets

Each player has three keys: right, left, and fire. These three keys are arranged as to minimize elbow-bumping. Since DogFight! reads the keyboard grid directly, it is possible that if you don't have a German keyboard you may encounter some discrepancies in the control descriptions and the actual controls.

Pressing right or left once will start your Jet into a curve. The Jet will continue in this arc until you press the key for the opposite direction, at which time the Jet will begin to fly straight again.

Pressing fire will launch a shot. You will not be able to fire again until that shot hits an opponant or disappears. You may hold down the fire key which will cause a sort of repeating fire; however, the rate of fire is very slow, and you may find you get better results by aiming your shots.

When a Jet gets destroyed, it will remain out of the game for about 15 seconds. At the end of this time, it will reappear in it's corner, in it's,original starting possition.

In addition to the keys which control the Jets, there are three general game control keys:

<ESC> (the escape key) - end the current game; returns you to the prefs window.

'p' - pauses the game

'o' - unpauses the game.

1.7 Legal Info

This program is freely released into the Public Domain.

This program may not be sold without my (specific) permission.

This program may be included into Public Domain series, such as Fred Fish, SAAR, Time, etc.

I do not guarantee anything. I will not be responsible if DogFight! fries your computer, although I'd appreciate it if you tell me if that happens. On the other hand, I'm not doing any strange hardware accesses or any other bizarre peeks and pokes. Ultimately, it's your risk (isn't it always?).

Feel free to modify the sourcecode. If you change and redistribute it, make it clear in the dox that the code was modified by you; if you rewrite the dox, make it clear that you stole some of my code ==-). I only require that if you redistribute it, you notify me and let me know where I can get a copy; I also retain the right to see the modified sourcecode. Please look at the [sourcecode](#) notes before you attempt anything short of a complete re-write.

1.8 Bug lists

I've given up calling this a beta release. It should play just fine. However, I make no guarantees that the program will work for you.

DogFight! exits nicely if it has any problems. If you try to run DogFight! and nothing happens, run it from a CLI and most likely you'll get an error message in English, which may be able to help you find the problem (missing library, etc.).

DogFight! may encounter problems with programs like CPUClr. If you do have problems with DogFight!, try removing these programs and turning off any CPU caches.

And now for the bug list:

I have to explain my situation first. DogFight! started with MaxonC++, and is C++ code. I always had problems with the occasional GURU when I ran DogFight!, and I picked through the sourcecode with a fine-toothed sonic-screwdriver for more hours than it took to write the program looking for the problem to no avail. During a project I worked on with a friend, it came to my attention that MaxonC++ was anything other than bug free, and I had the luck to get a good opportunity to switch to SAS/C. Upon returning to the DogFight! project, I found out (to my dismay) that SAS/C requires at least 4 megs of free mem to compile C++ sourcecode. Well, to make this long story short, with a bare-bones system configuration, I can compile DogFight!, but it's not the ideal programming environment.

The point of all this is that by switching to SAS/C I've removed the occurrence of GURUs, but am unable to take advantage of SAS/C's superior debugging utilities for lack of memory.

(b) indicates a bug

(i) indicates a desired improvement

(r) indicates a feature that needs to be implimented

(u) indicates an undesirable feature

(b) Screws up the workbench on occasion. I use an overscanned, oversized, DBLPal Hires NoFlicker screen for my workbench, and DogFight! draws little colored lines all over it when I run the game. Performing a "Redraw All" solves the problem, but it is annoying.

1.9 About the Author

DogFight! is written by Sean E. Russell, who is serving time at the Oregon State University. If he passes the Parol Board, he will be released sometime in 1995, at which point he will be moving back to Germany. Probably.

To be honest, I'm not back at OSU yet, and so don't know what my EMail address will be for this last year. I'm not going to bother with a snail-mail address, since nobody bothers to write PD authors anyway. You'll just have to wait until the next release of DogFight! to contact me. Sorry.

1.10 History & ToDo list

History

v0.0 24-FEB-94 Got started because of a request by a friend for a multi-player game.

v0.1 - 0.3 Various stages of basic game developement.

v0.4 26-FEB-94 Added MUI interface and double-buffering.

v0.6 28-FEB-94 Wrote Dox, added alert for players<2; added info window (ooooh). Decided that DogFight! needs a built in mail reader.

01-Mar-94 Added menus (don't want to offend the Style-Guide Police, now do we?).

v0.9b 05-Mar-94 Changed MUI a bit; added scores display while pausing; changed color scheme; changed general speed of the game. Changed game controls.

v1.0 18-Mar-94 Reduced the frequency of crashes, though not eliminated. Added computer controlled Jets.

19-Mar-94 Took out ASL screnmode requester. It was more annoying than useful.

v1.1 21-Mar-94 Put the ASL requester back in and hid it in the menu. Took out the scores page from the Preferences window.

v1.2 30-Mar-94 Added the sound routines.

07-Apr-94 Sucked all the sounds into one file, called Sounds.8SVX.

17-Apr-94 Changed the stack requirements to TOO MUCH! Doesn't seem to crash any more. Am beginning to think I can release DogFight! into the PD after all.

21-May-94 The stack wasn't a real solution. Converted to SAS/C and solved the GURU problem. Have definitely decided to release DF! into PD.

ToDo

2 - Fix the crashes

4 - Parachuters

1.11 Baking Powder!

Before I go on to talk about the stuff I wrote, I must give some credit to a couple people who helped me (without their knowledge) make DogFight! as fluid as it is.

1 - The person at C= who wrote the DoubleBuffering example included in the MaxonC++ demos section; DogFight! would not exist without that sourcecode.

2 - Micheal D. Bayne, the author of Gershneblanker. I blatantly stole the FastSQRT code directly out of the ASwarm module sourcecode.

3 - Dave Schreiber, author of DSound. I also took parts of the Audio.device allocation routines from his DSound sourcecode.

Thanx to all of you mysterious strangers.

I've included the sourcecode.

This was a difficult decision for me. There are several reasons why I shouldn't have, and few reasons why I should. In any case, it's done, and I'll try to explain why I think I shouldn't (which should help you in compiling it yourself).

1 - This was an (almost) overnight hack, and I (initially) didn't think that any more than 4 people would see it. As such, it contains a number of problems inherent to hacks:

A - I didn't comment it as much as I should have.

B - The code is somewhat hectic. Well, all code is hectic, regardless of what the CIS professors say. But DogFight! really is a Frankenstein program, so good luck.

2 - Oh, yea... IMPORTANT: In case you didn't notice, this is C++ code. I know that I may not write the ideal OOP code, but I do use C++ features. Therefore, this program needs to be compiled with a C++ compiler (unless you really change the sourcecode).

3 - I still have absolutely no documentation on how to program the Amiga. No RKMs, nix. Everything I use in my programs I derived from other sourcecodes, or learned from MaxonC++'s HotHelp. Do Not consider this code as an example of Correct Programming of Amiga Specific Functions!!

Those are not nearly all of my fears and reasons for hesitating to release the sourcecode. However, ultimately, I live on examining other's code, and if only one person benefits from this, it was worth it. And perhaps someone will bother to debug and improve it before I get around to it.

Development environment

DogFight! was developed on a 1-year old german Amiga 4000/030.

It has been tested on:

⌘ A4000/030-25MHz 5MB RAM, 80MB HD

NEC MultiSync 3D

⌘ A3000 68030-25MHz 8MB RAM, several HDs

NEC MultiSync 2D

OmniBus graphics card

1.12 Even MORE people I'm in debt to...

Asher Y. Royce - Who started me on DogFight! in the first place and then kept me from working on it with incessant chess games.

Stefan Stuntz - The author of **MUI**.

Karin Niggel - Who was inhumanly patient throughout the entire process. (Isn't there always someone who was "inhumanly patient throughout the creative process"???)

Helmut (Lutz) Lutzenberger - Who lent me his ARKRM manuals and supplied me with tremendous amounts of example source-code. It was also through interaction with Helmut that I discovered the mysterious Stack Effect, which eventually solved all of my problems and made me a better person.

Olaf (Anrufbeantworter) Lange - Who made it possible for me to convert to SAS/C

My Brain - For functioning most of the time.

1.13 Magic User Interface

This application uses

MUI - MagicUserInterface

(c) Copyright 1993 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXXusr.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 20.- or US\$ 15.-

to

Stefan Stuntz

Eduard-Spranger-Straße 7

80935 München

GERMANY

This is the standard text included with MUI applications, but I'd like to add a few more words about MUI. You can skip this part if you are already registered.

I've always considered one of the most important aspects of a computer to be its user interface. The Amiga has one of the best designed GUIs, but is, of course, not perfect. Almost everything that the Amiga GUI is lacking is supplied by MUI. For example:

- ° Completely user definable look-and-feel, with a wide selection of images (for gadgets, etc).
- ° Almost everything that Windows has that the Amiga doesn't is in MUI, such as popup-lists, cool multi-color 3-D buttons, and popup-cycles.
- ° Font sensitivity.
- ° Each application can have it's own prefs.

The list goes on and on. There are already several MUI applications, and the list gets bigger every day. As a user MUI makes my Amiga feel like a workstation, and as a programmer MUI removes a lot of the work involved in generating a user interface for me.

If the look-and-feel of your interface is important to you, I must recommend that you register for MUI or obtain the MUI archive if you don't have it already. You may use MUI applications without registering (IE, without paying), but the full power and beauty of MUI (the ability to customize & save the prefs) is first yours when you register.

It's cheap, and it's worth it, and after all -- you spend more time interacting with the interface than anything else in the computer.
