Jaguar Sample Programs

This section describes the various sample programs that are included with the Jaguar development system which are not a part of the **Jaguar Workshop** series. Each subsection describes a particular program, and will discuss what the program does, what techniques it is supposed to illustrate, and to some degree how the code works. If you have not read the **Jaguar Software Reference Manual** already, you should do it before reading this section.

Please note that the sample programs are often intended to illustrate a particular idea in an easy to understand way. In most cases, this will not be the fastest method, or use the least memory, because such optimization frequently makes it harder to understand what's going on. Once you understand the Jaguar hardware, you will undoubtedly find a number of ways to do the same thing faster and with less code.

Atari is constantly creating new sample code, so in the event that there are changes or additions to the sample programs, there will be README.TXT files located in the SOURCE directory and/or within the specific subdirectory of the sample. You should also check the online services at least every couple of weeks to see what updates and additions are available.

Please note that in order to reduce the size of the archives containing the sample programs, the executable program itself is not provided in most cases; the project must be built using the tools in your Jaguar developer's kit. (This should serve as a useful reality check to be sure your installation is correct.)

Jaguar Mandelbrot / Fractal Demo

What it Does

This program demonstrates how to set up a full-screen bitmap object and then uses the GPU to draw a Mandelbrot fractal into it. Once the Mandlebrot set has been drawn, a Julia set is drawn, and the program then switches back and forth between the two images.

The 68000 is used to set up the parameters for the GPU, and then the entire screen is drawn by the GPU. As implemented, the whole screen is drawn in about 5 seconds, and could be sped up by a factor of 100% or more with a little more optimization (like using the DSP to calculate half the picture while the GPU calculates the other half).

Where is it?

This example is normally found in the \JAGUAR\SOURCE\JAGMAND directory. Below is a list of all the files which are included.

Filename	Description
CALCMAND.S	This is the actual Mandlebrot calculation code that runs in the Jaguar GPU.
CRY.PAL	This file contains data for a 256-entry CRY-mode color palette for palette-based objects.
JAGMAND.S	This file takes control after the startup code has initialized the system. It creates an object list for the background picture, installs an object list refresh routine, and then calls the code in MANDLE.S.
MAKEFILE	Used with MAKE utility to build executable program file from source code and data files.
MANDLE.S	This uses the 68000 to set up the fractal parameters and then calls the GPU to calculate the image.
STARTUP.RGB	This file is actually in the \JAGUAR\SOURCE directory. This is the screen displayed by the startup code that is used by several of the sample programs in the Jaguar Developer's Kit.
STARTUP.S	Standard Jaguar Startup Code. This module contains all the code necessary to properly initialize the Jaguar hardware and display a simple startup picture. Then it passes control to the *_start* label in the JAGMAND.S module. ***(See the* Sample Programs *section for further information on the Standard Jaguar Startup Code.)***

STARTUP.S

This file is where the program execution begins. This is the standard Jaguar Startup Code responsible for initializing the system. It sets up interrupts, sets the video registers correctly for either NTSC or PAL, and does other related things that must be done properly at startup time for your program to function. It also displays a startup screen. Once it is finished, it passes control to the *_start* label somewhere in your program (JAGMAND.S in this example).

Note that STARTUP.S has been modified slightly from the version in \JAGUAR\STARTUP to allow the use of a different startup picture. This type of change is only one allowed in this file. Making changes to other portions of the file may result in errors which can prevent your program from functioning properly.

JAGMAND.S

This file is where the program execution begins after the startup code has initialized the system. It basically delays for a few seconds so that we can look at the startup screen, then it creates an object list for our background picture, installs an interrupt handler to refresh the object list, and then sets the video mode to 320-pixel CRY mode. Finally, it clears the memory that will be used for our bitmap, and then jumps into the ***Mandle*** function, located in MANDLE.S.

Note that the object list creation routine ***make_list*** is almost identical to the routine ***InitLister*** in the STARTUP.S module. The only parts that changed were the labels for the address where the list information is stored.

MANDLE.S

This contains the 68000 routine that sets up the fractal parameters (coordinates, zoom range, etc.) and tells the GPU to start creating the fractal image.

CALCMAND.S

This contains the GPU routine that calculates the fractal image for each pixel of the picture, using the parameters (coordinates, zoom range, etc.) which are set up by the 68000.

JagLine, JagSlant, JagBlock, JagSkew, JagShade

These are very simple programs which demonstrate how to do specific tasks using the blitter.

Warning! *Please note that the current versions of these programs are not intended as general examples of Jaguar programming. They are intended as simple examples of specific blitter operations, and they take short cuts to this end.* ***Do not use these examples to obtain startup code or as a shell for creating your own programs.***

What They Do

JagLine - This program demonstrates how to draw a horizontal line using the blitter. It sets up a narrow bitmap object and then draws a single yellow line into the top of it.

JagSlant - This program demonstrates how to draw a diagonal line using the blitter. It sets up a narrow bitmap object and then draws a single yellow line into the top of it.

JagBlock - This program demonstrates how to draw a solid rectangle using the blitter. It sets up a narrow bitmap object and then draws a single yellow box into the top of it.

JagSkew - This program demonstrates how to draw a skewed rectangle using the blitter. It sets up a narrow bitmap object and then draws a non-shaded yellow polygon into it.

JagShade - This program demonstrates how to draw a shaded parallelogram using the blitter. It sets up a narrow bitmap object and then draws a shaded yellow 4-sided polygon into the top of it.

Where are they?

This example is normally found in the \JAGUAR\SOURCE\BLIT directory. This directory contains several demos which share a number of common source code files. Below is a list of all the files which are included.

Filename	Description
BLITBLCK.S	This is the code for **JagBlock** that calls the blitter
BLITLINE.S	This is the code for **JagLine** that calls the blitter
BLITSHAD.S	This is the code for **JagShade** that calls the blitter
BLITSKEW.S	This is the code for **JagSkew** that calls the blitter
BLITSLNT.S	This is the code for **JagSlant** that calls the blitter
CLEARBAR.S	The routine in this file uses the blitter to clear the bitmap memory used by the program
CRY.PAL	This file contains data for a 256-entry CRY-mode color palette for palette-based objects.
INTSERV.S	This file contains the interrupt handling routines used by all the programs.
JAGLINE.S	This is the main program file for **JagBlock, JagLine, JagShade, JagSkew**, and **JagSlant**
LISTBAR.S	The routines in this file set up the object list used by all the programs
MAKEFILE	Used with MAKE utility to build executable program files from source code and data files.
VIDEOINI.S	The routines in this file set up the video display used by all the programs.

BLITBLCK.S, BLITLINE.S, BLITSHAD.S, BLITSKEW.S, BLITSLNT.S

These files contain the code for the blitter for the individual programs. Only one file is used by each program (see table above).

CLEARBAR.S

This file contains a simple subroutine which uses the blitter to clear the memory used by the bitmap object we use to display our picture in all of these programs. It sets up a pattern containing all zeroes, and then blits this pattern into the bitmap.

CRY.PAL

This file contains data for a CRY-mode color palette, which will be used by objects with 8 bits per pixel or less.

INTSERV.S

This file contains the routine that installs our vertical blank interrupt, as well as the vertical blank interrupt service routine (ISR). The ISR simply calls the *Lister* function (contained in LISTBAR.S) which creates the object list.

Note that re-creating it from scratch during each vertical blank is a terrible way to maintain your object list; ***please don't do it this way***. It's much more efficient to change only those fields of those objects which get changed every frame by the object processor. For better examples of creating and maintaining an object list, see the programs in the \JAGUAR\WORKSHOP directory, which create object lists of various sizes and complexity. For a specific example of an object list like those used by **JagLine,** etc., see the routines in the file MOU_LIST.S, located in the \JAGUAR\WORKSHOP\MOU directory.

JAGLINE.S

This file is the main source file for these programs. It performs program initialization, and then transfers control to the *DoBlit* function, which is different for each program (this routine is contained in the BLITBLCK.S, BLITLINE.S, BLITSHAD.S, BLITSKEW.S, and BLITSLNT.S files; each program uses just one of these).

LISTBAR.S

This file contains the *Lister* routine we use to create our object list, as well as the routines which save and restore the fields of the object list which are modified during each frame by the object processor.

VIDINIT.S

This file contains the routine that detects the current video standard (NTSC or PAL) and sets up the video registers which control aspects of the video such as the size and position of the borders at the edges of the screen.

Joypad Reading Example

What it Does

This program demonstrates how to read the Jaguar joypad controllers. It is quite simple; the current buttons pressed on the joypad are printed to the screen. Controller #1 is shown on the left side, and Controller #2 is shown on the right side.

Where is it?

This example is normally found in the \JAGUAR\SOURCE\JOYTEST directory.

EEPROM Example

What it Does

This program demonstrates how to read and write information to the EEPROM of a cartridge.

The EEPROM is 128 bytes of non-volatile memory on a standard Jaguar cartridge that is normally used for storing the user's controller preference settings, high scores, etc. This program demonstrates how to access it.

Note: This program demonstrates **the exact** method required for accessing the EEPROM. Use the code from this program as is, without change.

Where is it?

This example is normally found in the \JAGUAR\SOURCE\EEPROM directory.

RGB True Color Bitmap Display Example

What it Does

This program demonstrates how to set the system up for RGB mode instead of CRY mode, and creates a 16-bit true color RGB bitmap object. It then draws a number of bands of color into the object. This program uses only the 68000, and while it's not exactly slow, it could be done much faster using the GPU and/or Blitter.

Where is it?

This example is normally found in the \JAGUAR\SOURCE\TESTRGB directory.

Simple DSP Waveform Output

Warning! *Please note that the current version of this program is not intended as a general example of Jaguar programming. It is a simple example of a specific DSP operation, and it takes short cuts to this end.* ***Do not use this example to obtain startup code or as a shell for creating your own programs.***

What it Does

This program demonstrates how to playback a simple waveform using one of the samples in the DSP waveform ROM. Nothing is shown on screen, but you should hear a tone from your speakers.

Where is it?

This example is normally found in the \JAGUAR\SOURCE\SIMPLE directory.

Blitter Demo

What it Does

This program is a sort of Blitter recipe book written by Francois-Yves Bertrand. It uses the blitter to copy a bitmapped picture from the source bitmap to the screen. Then it allows you to plug values into the blitter registers to see what happens.

This program is really as much a tool you can use to figure out what values to use with your own blitter code as it is a sample program. Playing with this program as you read through the blitter sections of the **Jaguar Software Reference Manual - Tom & Jerry** is really a great way to learn the Jaguar blitter.

With this tool, you can program any of the blitter register and see the result directly on screen. The actual program uses two main objects:

- The first one is an ATARI logo, 64 x 64, 16 bits per pixel. This is used as the source.
- The second one is the destination buffer. It is 320 x 256, 16 bits per pixel, 3 layers (2 for double buffering and one for Zbuffer)

You can move around the register with the UP/DOWN keys or faster with 1/7 keys on paddle 1. You can change the value of a register with LEFT/RIGHT keys or faster with C/B keys. The only register you cannot change is the base register (for both A1 and A2). If you set the DSTA2 register (so A1 is the source and A2 the reception), the program swaps the A1 base and A2 base. You will have to swap manually all the other registers (PITCH,PIXEL SIZE...) to have the correct result on screen.

The source code for this program is not provided. While the program itself is interesting to play with, and useful as a tool to help figure out your own blitting routines, the source code is not really a good Jaguar programming example in general.

Where is it?

This example is normally found in the \JAGUAR\BLITTER directory.

BPEG Decompression Example

What it Does

TESTBPEG is a sample program for the Jaguar that demonstrates how to take the files created with the BPEG image compression tools and use them in a program with the BPEG routine and tools.

For further information, please see the **Libraries** section.

Where is it?

This example is normally found in the \JAGUAR\BPEG directory.

Jaguar Synth Example

Warning! *Please note that the current version of this program is not intended as a general example of Jaguar programming. It is a simple example of using the Jaguar Synth and Music Driver, and it takes short cuts to this end.* ***Do not use this example to obtain startup code or as a shell for creating your own programs.***

What it Does

This program demonstrates how to use the Jaguar Synthesizer and Jaguar Music Driver to play music in your programs.

For further information, please see the **Libraries** section.

Where is it?

This example is normally found in the \JAGUAR\MUSIC\SYNDEMO directory.

A different example that uses a wider variety of patches for the synthesizer may be found in the \JAGUAR\MUSIC\MUSICDRV directory.

3D Rendering & Texture Mapping Demo

Warning! *Please note that the current version of this program is not intended as a general example of Jaguar programming. It is an example of using the 3D Graphics library, and it takes short cuts to this end.* ***Do not use these examples to obtain startup code or as a shell for creating your own programs.***

What it Does

This program encompasses and demonstrates the Jaguar 3D Graphics routines supplied by Atari. The program draws a fully light-shaded and texture mapped space fighter on screen. Using the joypad controller, you can control the fighter's position and orientation.

Controller Button	Movement
'up'	Rotates you forward
'down'	Rotates you backward
'right'	Rotates you to the right
'left;	Rotates you to the left
'A'	Moves you forward.
'B' or 'C'	Changes the light shading
'1'	Rotates you counter-clockwise
'3'	Rotates you clockwise
'6'	Changes light intensity
'7'	Reduces number of objects
'8'	Turns on/off object rotation
'9'	Increases number of objects

The number of objects on screen increases/decreases exponentially when you use the '7' and '9' keys; you can have 1 object (1^1), 8 objects (2^2), 27 objects (3^3), and so on.

Where is it?

This example is normally found in the \JAGUAR\3DDEMO directory.