

Introduction to the Propeller Animation & Video Game Programming

By Jeff Ledger ‘Oldbitcollector’
March 2010 (pre-release 1.1)

Welcome to the world of the Parallax Propeller! While many microcontrollers exist on the market, the Propeller is unique due to its ability to generate video graphics and text using little more than a few simple components. The Propeller is a capable platform for learning object oriented programming as well as basic video game design. It is the video game design which this text will cover. Don't worry about having to be some kind of programming genius or super geek. I'll take you step by step. Even if you have never used a microcontroller before you'll soon be working with graphics and animation!

The recommended requirements for this chapter are a Propeller Demoboard, or any Propeller board which has had the three required resistors added to display TV video. You will also need access to a television or composite monitor which has an **AV IN** connection. Connect an RCA-to-RCA cable from the TV connection on the Propeller Demoboard and connect that to your television. Make sure that you have your TV set for AV-IN or game mode.

You will also need access to a Windows PC. Download and install the following tools.

The Propeller Tool

<http://www.parallax.com/tabid/442/Default.aspx>

The ORE Graphics Engine

<http://www.orrtech.us/oredemo.zip>

Download and install the Propeller Tool.

Download the **OREDEMO.ZIP** file and extract the content to an empty folder.

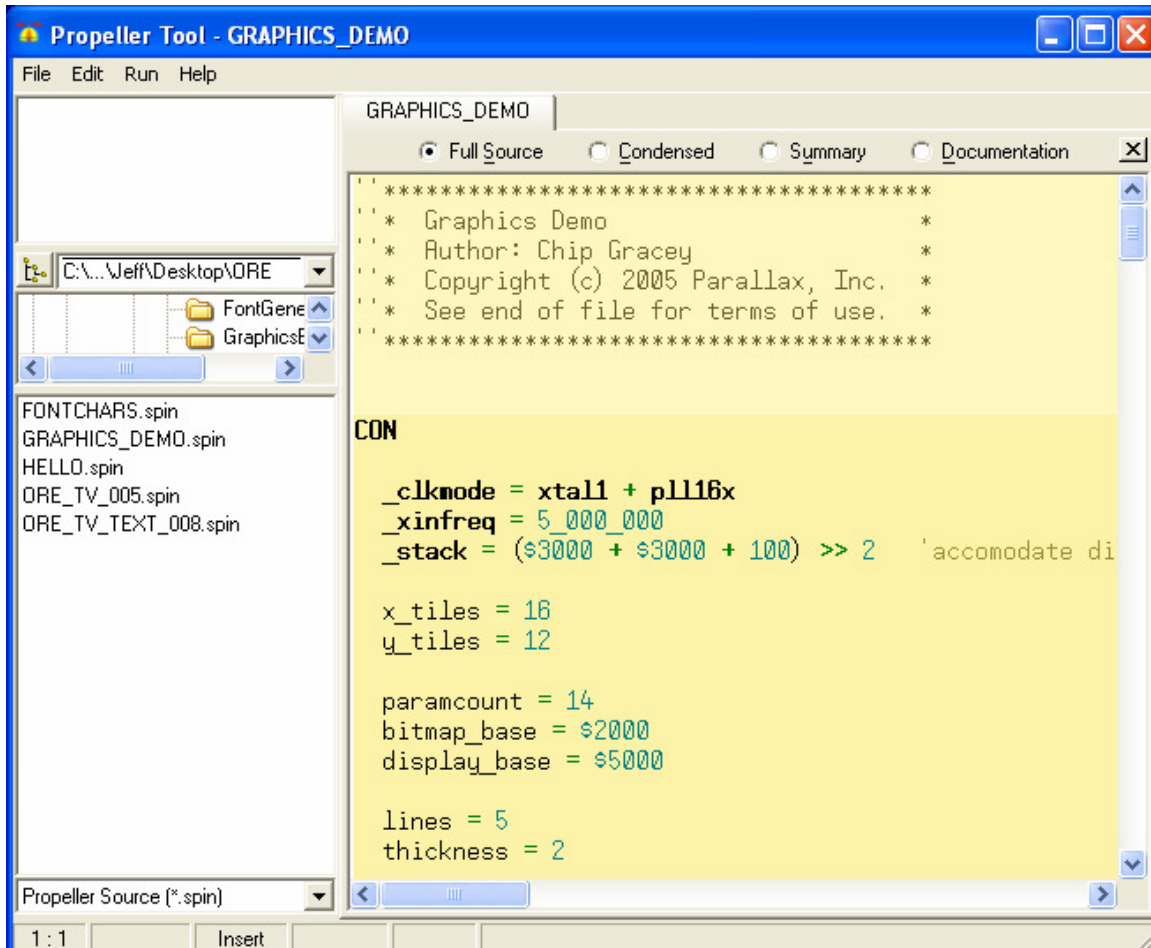
The Propeller Tool is the official programming environment (IDE) for the Propeller and is provided free of charge. We'll use it to load programs from the computer into the Propeller. The electronic form manual is also provided free of charge. You may want to invest in a printed copy from Parallax as you start programming in "spin", the language used in the Propeller Tool.

When you install and run the Propeller tool, you'll be asked to “Associate .spin .binary and .eprom files.” Answering yes will configure the tool properly to open .spin files.

Getting Started: Chapter 1

Connect the Propeller Demoboard to your PC using the supplied USB cable. Make sure you have the video cable connected as well. Switch on the Propeller Demoboard.

Let's start with a simple graphics demo to make sure everything is working properly. Remember that folder you extracted the **OREDEMO** files to? Double click on the file called "**GRAPHICS_DEMO**". The Propeller Tool should open with a screen that looks like the one below.



Press the **F11** key on your PC.

This will send the Graphics Demo code to the **EEPROM** on your Propeller Demoboard. The EEPROM is the tiny chip next to the Propeller chip in the center of your Demoboard. The eeprom will store your program even when the power is off. Each time you switch on the Demoboard it will load the **Graphics Demo** until you use the Propeller Tool (F11) to upload a new program.

If all has gone well, you should have a colorful, graphics display on the TV.

Animation Basics

Now that you have the hardware working properly, let's take a step back for a moment and look at the basics of animation. Have you ever draw a series of characters on a corner of a notebook and watched them move when you flipped the corner quickly? We will be using the same type of "page flipping" animation, only instead of drawing them in the corner of a book, we'll place the characters on the screen, animating them by clearing the screen very rapidly, each time, moving or changing the character.



Animating a walking character simply requires us to create a graphic for each position. Making the character walk across the screen will only require us to display the first image, clear the screen, display the next image one step to the right, clear, and repeat.

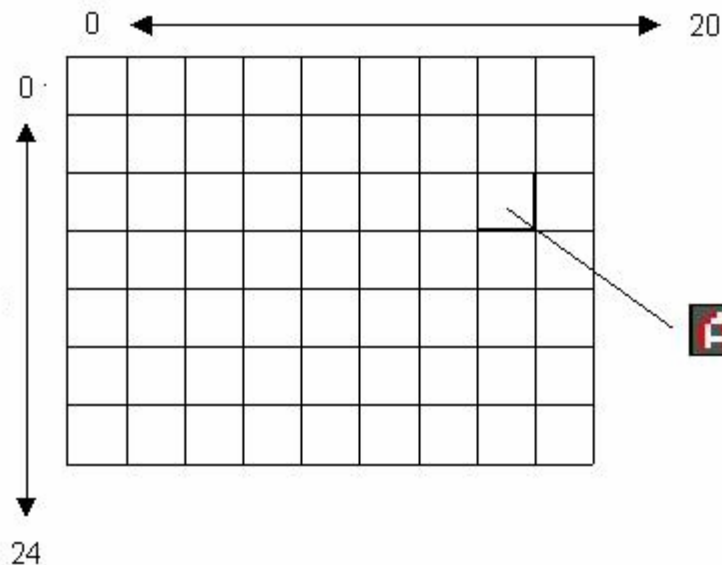
Sound easy? Animating and creating video games on the Parallax Propeller is both fun and easy with the ORE Text Driver. The ORE software takes care of all the difficult video programming, leaving us to do the creative part.

ORE Text Driver Specifications

The Parallax Propeller is capable of hi-resolution graphics with thousands of colors. The ORE Driver scales that back just a bit to provide a balance between memory requirements and ease of use.

Display Mode: Text & Characters (each character 8x8 pixels)
Resolution: 20x24 Characters per page. (160x192)
Number of Colors: 126 total colors, 4 per 8x8 character.

ORE SCREEN LAYOUT



Getting Started with ORE

Double click on the HELLO file you extracted from the ORE Text Driver archive. Make sure your Propeller Demoboard is turned on with the USB cable connecting it to your PC. Press the F10 key to download and run the HELLO program.

If everything worked correctly, you should see HELLO on the screen.



Looking at the spin code, let's focus on the section of code (in the blue section) that looks like this:

```
REPEAT
  TEXT.CLS
  TEXT.POKECHAR(3,10,COLOR_WHITE,COLOR_BLUE,COLOR_RED,72)
  TEXT.POKECHAR(4,10,COLOR_WHITE,COLOR_BLUE,COLOR_RED,69)
  TEXT.POKECHAR(5,10,COLOR_WHITE,COLOR_BLUE,COLOR_RED,76)
  TEXT.POKECHAR(6,10,COLOR_WHITE,COLOR_BLUE,COLOR_RED,76)
  TEXT.POKECHAR(7,10,COLOR_WHITE,COLOR_BLUE,COLOR_RED,79)
  TEXT.UPDATESCREEN
```

The **REPEAT** command sets up a continuous loop. Every line indented beneath it will be repeated over and over.

The **TEXT.CLS** clears the screen each time the loop repeats.

Each of the **TEXT.POKECHAR** commands place each character on the screen.

Finally, the **TEXT.UPDATESCREEN** command tells ORE to display the information

TEXT.POKECHAR(X,Y,COLOR1,COLOR2,COLOR3,CHARACTER)

X = Row (0-20)

Y = Column (0-24)

Color1,Color2,Color3 = Any defined color.

Character = 0-126

How to edit the characters

There are actually several ways we can edit the characters displayed on screen. The file **OREFont.spin** contains the actual spin code for each letter, number and graphic. Opening it with the Propeller Tool, we can make changes to the code itself.

It looks like this...

```
word %%00000000      | ..... $45
word %%01111120     | .S*****.
word %%00000120     | .S*.....
word %%00111120     | .S*****.
word %%00000120     | .S*.....
word %%00000120     | .S*.....
word %%00000120     | .S*.....
word %%01111120     | .S*****.
word %%00000000     | .....
```

The actual font data is the backward E on the left. Each of the numbers represents colors used in displaying this letter. 0 = background color, 1=color 1, and 2=color 2.

The E displayed on the right is simply a comment. Any time you see a `'` mark in spin code it means that the text after it is a remark or note by the programmer.

Fortunately, there is an even easier method of editing characters for the ORE Driver.



I think Leonardo would have loved Propeller animation. Don't you?

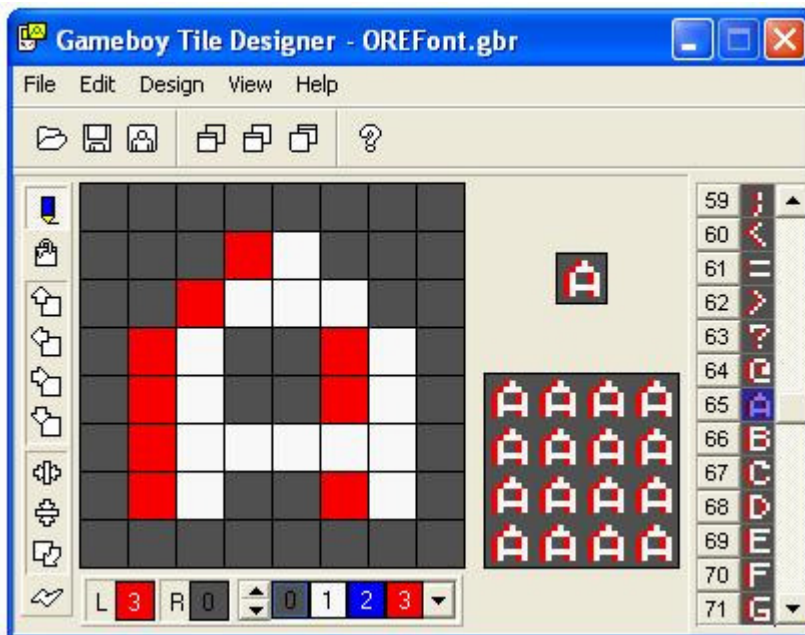
Introducing Gameboy Tile Designer

Gameboy Tile Designer was written by Harry Mulder © 1999. He has released the program as freeware which may be redistributed without charge. His website is: <http://www.devs.com/gb/hmgd/gbtd.html>

GBTD (Gameboy Tile Designer) is obviously not an official Propeller tool. Because the ORE Driver uses the same 8×8 *2bpp* type of graphics also used by the Gameboy, adapting the output from this program is pretty straightforward. (Thanks to Alex Schafer)

Double click on **GBTD** and open the file **OREFont.gbr**. (You can accomplish the same thing by simply dragging the **OREFont.grb** icon onto the **GBTD** icon.)

It should open with a screen like the one below:
(I've scrolled down to edit the letter A.)



Notice the number 65 near the character I've scrolled to? This is the character number used in end of the **POKECHAR** command. So to display an A, we would use:

```
TEXT.POKECHAR (10, 3, COLOR_WHITE, COLOR_BLUE, COLOR_RED, 65)
```

Also notice the numbered colors at the bottom of the window. The grey 0 is the background color. The white 1 is color 1, the blue two is color 2, red three is... You get the picture.

While editing the characters in **GBTD** we are limited to grey, white, blue, and red, but when we call them with the **POKECHAR** command we can use any color we please. Let's make some changes to our HELLO program.

Scroll down to character 69, the uppercase E and make some changes to it with **GBTD**.

Really get creative with it! Select your working color by click on it at the bottom of the window, clicking each of the cells to change its color.

Now, click on **File**, and **Save** recording these changes to the **OREFont.grb** file.

Now minimize the **GBTD** program and double click on a file called: **makefont**
Makefont is a simple batch file which calls on the Graphicconvert program to create a new OREFont.spin from your changes. You should see a small, black window appear for a moment, and then disappear. At this point, you have created a new ORE compatible font based on your changes.

Keep in mind the steps to make this work are in this order:

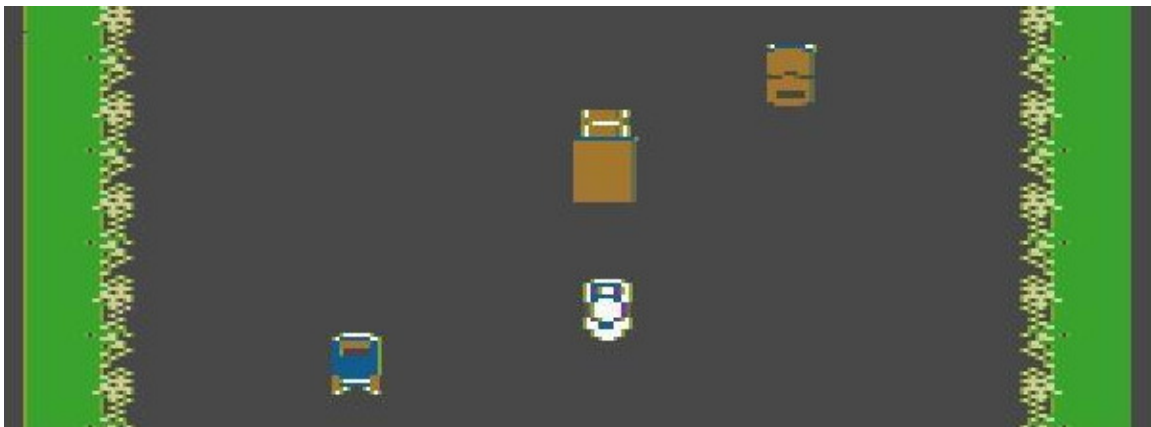
- 1.. Make your changes in **GBTD**.
- 2.. Click on File and Save in **GBTD** to (re)save the .grb file.
- 3.. Double click on **makefont** to (re)convert the .grb file to spin code.
- 4.. Run your main program with the Propeller Tool using **F10**. (F11 to save it to eeprom)

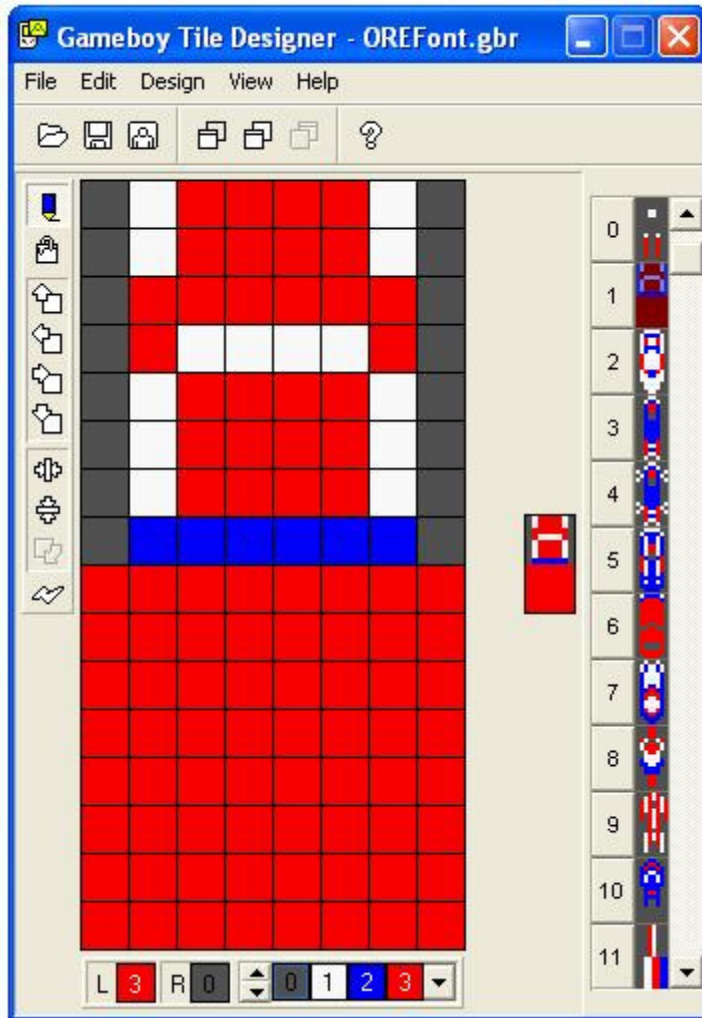
Ready to see your work of art?

Close any other files you have open in Propeller Tool and open the HELLO program again. Use **F10** to send the program to the Propeller once again. This time you should see HELLO with your glorious new **E** character on the screen!

The GBTD has several handy features for editing your graphics. You can shift, or flip your character with the small buttons on the left. You can also change your view to edit more than one character at a time. By clicking on **View** and **Tile Size**, you create larger characters by combining them. Just remember that when you call on them with the **POKECHAR** command, you will need to use two commands to display each part of the character you created.

Take a look at the example on the next page of some two character sized graphics I created for my Spyhunter game. (pictured below)





The numbers displayed in this mode are not correct for use with **POKECHAR**, however you can easily see the advantages of being able to edit two or more characters at the same time.

NOTE:

Characters tend to look a little stretched in the editor, but look fine when used with the Propeller. ORE blocks tend to appear more rectangular in actual use.

IMPORTANT NOTE:

Do not use characters **0** or **32**. These are used for blanking the screen as well as generating spaces.

You can try to use them. It'll create some strange results. ☺

If you create some interesting characters with GBTD, please drop by the Propeller Powered forums and share them with us! <http://www.propellerpowered.com>