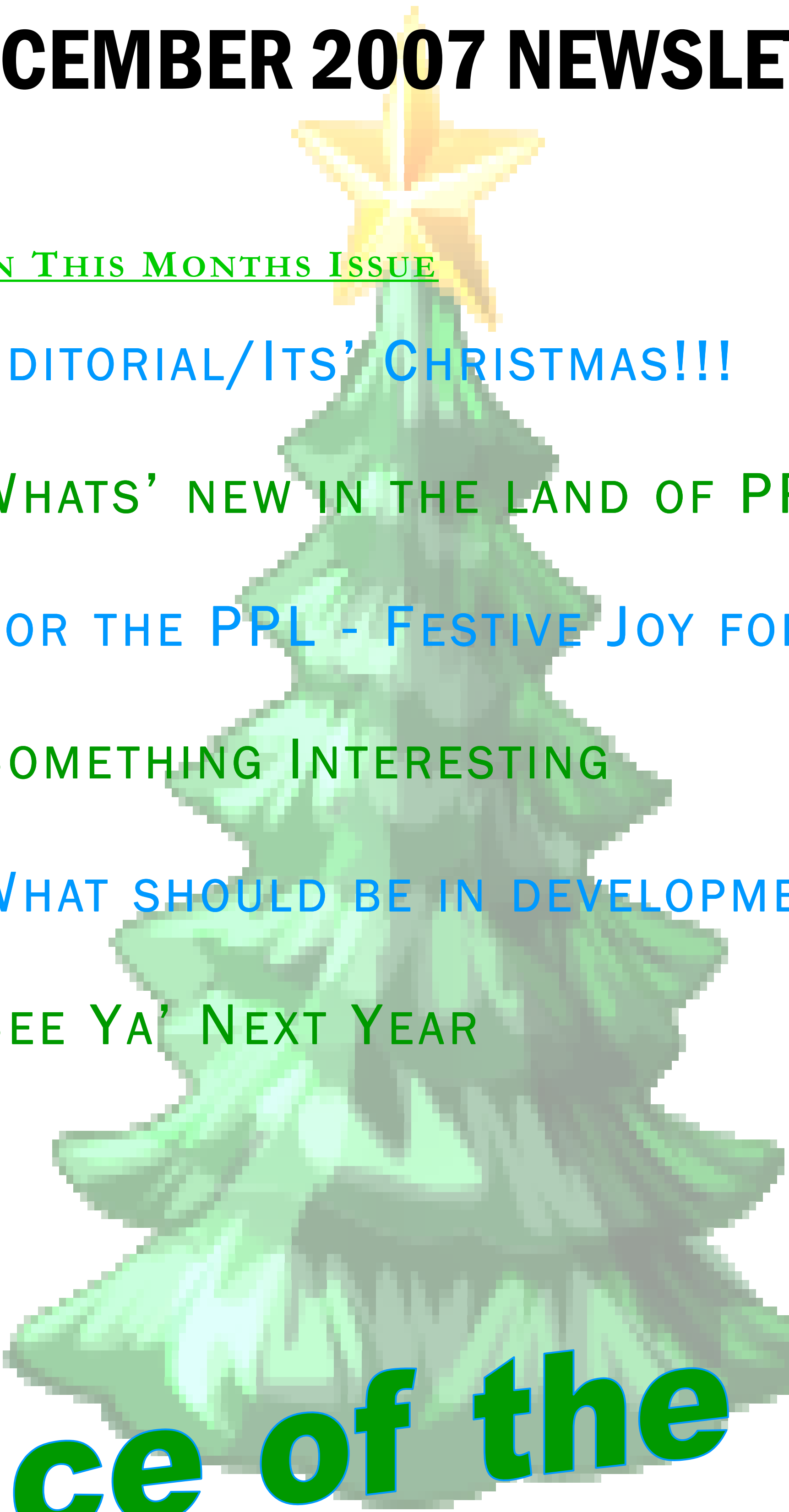
DECEMBER 2007 NEWSLETTER



In This Months Issue

Voice of the PPL

ArianeSoft
The power to create

Editorial/ Its' Christmas!!!

Welcome to the December issue of the PPL Newsletter.

A Merry Christmas to you all.

Blimey, another Christmas—that snook up on us! Where has 2007 gone?

This issue is a bumper one to tide you over the festive period and we hope you enjoy the extra content.

A mass of things have happened in 2007 as far as PPL is concerned, and you can guarantee that 2008 will be even more jam packed!

V1.4 has just been release and contains only what can be described as a plethora of updates and extras to keep you busy well into the new year.

Also..

Keep an eye out for some great software releases that were developed using PPL.

In the new year we hope to bring you a new demos section where you the reader can provide links to demo versions of your creations—The idea behind this is to get some publicity and awareness out in the wild.

Enjoy this extended edition of the newsletter and many happy hours of PPL coding.

LOOK ———->

25%
off
Until Christmas

Thanks

The Arianesoft Team

Whats' new in the land of PPL

At the time of writing (start of December), we have 788 registered users on the forum. Of that only a small percentage of users have purchased the PRO version of PPL and even less you have a STD version.

Come on guys, lets buy some PPL licenses for friends, family and colleagues this Christmas—I'm sure it would make any budding smartphone/pds programmer smile. Even if its only the STD version!!!! WHAT ARE YOU WAITING FOR?

Once your trial key runs out, PPL is functional but with there being no GameAPI or compile to EXE it's only real use is for applications development.

Special offer until Christmas!
PPL 1.40Pro for $49.95 (25% saving)

For your money you get an unbelievable amount of functionality, features and development ease.—V1.40 adds a whole host of updates and functions.

A certain large corporation would and does charge $$$$ for their mobile development platform, and theirs is not cross platform compatible right out of the box. Hey Alain, maybe there is a new year buy-out looming? - I certainly hope not! - We will all help you fight off the big monster that is MS...

We as a community need to support Arianesoft buy purchasing more licenses, contributing tutorials, code and suggesting improvements to PPL.

Thus making it grow and hopefully become a major league player in mobile development. This way we all benefit from a fine product.

I am actively working on more code to share with the PPL community and they should all be available for download in the new year!

So Alain, keep up the good work and the community, keep supporting Arianesoft!!!!

Just a thought to keep you going ... Regards until next month

Mike.

Make sure you all download and install V1.40 (right away) as it contains a host of new features and some important bug fixes!

For the PPL - Festive joy for all

In these articles I will try to give you all a Christmas Pressie! - Many examples of varying PPL technical standings and tutorials.

1. Surfaces - Just scratching at them.

A surface is an area of memory that we can use to manipulate images and other graphic content.

Surfaces do not automatically get displayed on the screen when loaded! Instead we use one of the drawsurface commands.

Surfaces are great for loading and 'fiddling' with bitmaps.

Handle$ = loadsurface(AppPath$ + "surface.bmp", RGB(0,0,0));

The above command loads a bitmap from the current application path into memory and returns the pointer as Handle$ it also sets the transparency to black.

DrawSurface(source handle$, destination Handle$, X,Y, Width, Height);

The above command will display all of the image at co-ords X,Y with a size of Width and Height

Just like sprites, the loaded surface can be moved around the screen by changing the X and Y values and re-issuing the drawsurface command.

Of course, we can be really clever with surfaces. PPL gives us 4 powerful surface drawing commands.

DrawSurface, DrawSurface2, DrawSurface3 and DrawSurfaceEX.

They are all similar except that each one adds extra functionality to the last. Check the help for syntax and examples of each.

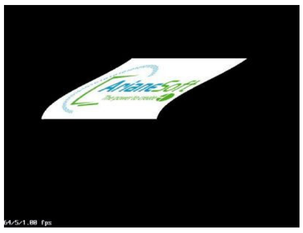
Using BUFFER% returns the memory pointer to the current display area instead of another surface.

Using DrawSurface2 you can specify which scan lines in the source image are displayed on the destination surface

DrawSurface3 does everything drawsurface2 does and also adds a 'rotational angle' option.

DrawSurfaceEX duplicates DrawSurface3 and adds the ability to control the clipping of the surfaces.

(See the PPL help for examples and exact syntax)



See the Tutorial archive that is included with this newsletter for full source code!

2. PIDE - A quick way to nicer looking code

While I'm coding I like to make sure that everything is structured and in an easy readable order. There are times how ever when I just code and code without stopping. This is when it becomes difficult to see what is going on.

```
For(x$,1,1000)
For(y$,10,1,-1)
Value$=X$+Y$;
If(X$==100)
Value$=a%/b%;
End;
//display something here
End;
End;
```

Ok, so this is a poor example of coding and of examples—but its good enough for you to get the idea. The above code block is a bugger to read (especially when it gets over a full page in size) .. Nothing spaced out, nothing structured. Difficult to read.

PIDE has a great option that allows you to re-format the code within your game or application.

Select the EDIT menu, then select the 'Format Code' option.

PIDE will now scan through your code and indent all loops, conditions and procedures, making your code much more readable and easier to debug!

```
For (x$, 1, 1000)
  For (y$, 10, 1, -1)
    Value$ = X$ + Y$;
    If (X$ == 100)
      Value$ = a% / b%;
    End;
    //display something here
  End;
End;
```

Simple I know, but invaluable when making your code more readable.

3. PIDE - NO Comments!

Ever needed to block comment a few lines of code? So have I, except that I was doing it manually, adding // to each line!

There is an easier way! - highlight the code you want to comment and select ‘edit’ then ‘comment code’ - voila code commented with /* at the start and */ at the end of the comment block.

Easy peasy!

4. Sprites - Collision detection

Sprite collision detection in PPL was something that I had not played with properly and thought it might be a bit tricky to accomplish. I had a few moments to spare so I had a quick look at doing simple collision detection.

Here is an easy way to get your head around it. - The PPL instructions on the SETSPRITECOLLIDE command are a bit thin on the ground and don’t really give an example. So … This is what I have found by playing around.

In WinMain

```
  // Load the sprite in to memory handle
handle$ = loadsprite(AppPath$ + "sprite.bmp", g_rgb(255, 255, 255), 1, -1, NULL);
  // Set the sprite ID for collision purposes
setSpriteID(handle$, "hero");
  // Add collision options to this sprite
AddSpriteOption(handle$, SO_COLLIDE | SO_CHECKCOLLIDE | SO_PIXELCHECK | SO_BORDER);
enemy$ = loadsprite(AppPath$ + "enemy.bmp", g_rgb(255, 255, 255), 1, -1, NULL);
AddSpriteOption(enemy$, SO_COLLIDE | SO_CHECKCOLLIDE | SO_PIXELCHECK | SO_PLATFORM);
setSpriteID(enemy$, "enemy");
  // Set the collision rules for the sprites
SetSpriteCollide(enemy$, "hero");
  // This sets up collision detection between enemy and hero IDs

  // Have the API detect collisions that can be processed in WM_COLLIDE
 GameCollide(TRUE);
```

WM_COLLIDE is called every time a sprite collision occurs.

Within WM_COLLIDE processing of the collision results should be performed using

```
Coll$ = COLLIDE(handle$,X,Y, X1, Y1)
```

Because collide returns the handle of the sprite collided with you can tell if your character has hit say a wall or a missile.

As long as every sprite in your playing area has an ID and a SETSPRITECOLLIDE set you should be able to detect the collision and return the collided sprite handle.

Using the SO_BORDER parameter in the ADDSPRITEOPTION also allows for collision detection with the screen border.

I have used this technique in my ‘BLOCKRUNNER’ game. A simple idea of dodging blocks as they move around the play area. - Source code will be included with newsletter archive file so you can inspect how it was written.

5. Dear Santa … can I have an IPhone?

Well not exactly … but you can have the next best thing - A smart phone that actually turns on and is not locked to a provider.

PPL has been used to re-create I-Phone functionality on a PDA - It was available from many web sites, but APPLE in their infinite wisdom (or stupidity) have barred any duplication of their icons and images - The really funny thing is that APPLE only mentioned the icons in their writ .. So does that mean the source code (which is in PPL of course) and the executable can still be used, as long as you change your icons and unlock background?

I have the complete source code (in fact I have 2 different versions of the source code, but 2 different people) - It’s a great in-sight into how to code a PPL application.

This is also a good tutorial on how to develop gesture based input on your mobile device.

PM me on the forum if you want copies of it - It does seem to be harder to find these days.

PLEASE NOTE - ONLY THE SOURCE CODE IS INCLUDED, YOU WILL HAVE TO FIND YOUR OWN ICONS AND BACKGROUNDS. (That should please the APPLE legal pundits!)

6. 3D - Understanding the fundamentals - As easy as A, B, C

A. Introduction to 3D Space

To many of you, the concept of 3D space is unfamiliar, despite the fact that we live in it everyday. The most important thing to remember when thinking 3D is that objects that are far away appear smaller than those that are close. This is obvious to us, our brain has learned to interpret that fact, however it's not so obvious to a computer. You've got some work to do if you want to turn a 2D monitor into a 3D world.

First of all, let's look at graphing in space. 3D space (R3) looks something like this on a graph.

```
     Z
     |
     |
     | i
     |________
     /     Y
    / é
   /
  /
 X
```

Where Y is the horizontal axis
Z is the vertical axis
X is the axis coming out of the screen

é is the angle in the XY plane
i is the angle from the Z axis

The symbols above will be used throughout this text.

The two problems we have to deal with are this:
1. How do you create the effect of depth on a flat screen?
2. How do you rotate points in space?

Now that we've got the basis, let's answer number one.

B. Drawing a 3D Object on a 2D Monitor

The answer to this problem is surprisingly simple. First think about what's going on. Let's say we have a transparent box on the screen, we want the side that is closest to us to be bigger than the side that is farther away. So how can we do this, simple, divide the points making up the far away side by the distance to that side.

So what does that mean for plotting points in 3D. We start off with three coordinates, but we want two, so divide two of the coordinates by the other one. Here are the formulas:

$$x2D = 256 * (x3D / (z3D + zCenter)) + xCenter$$
$$y2D = 256 * (y3D / (z3D + zCenter)) + yCenter$$

In the above formulas, (x2D, y2D) is the point that will be plotted on the screen, (x3D, y3D, z3D) is the point in 3D space, xCenter and yCenter represent the location of the center of the object on the screen, and zCenter represents the location of the 3D center of the screen or the location of you, the viewer (this value is usually 256). Both equations are multiplied by 256 to relocate the object adjusted to the viewer's perspective.

Using those formulas makes it possible to draw a 3D object on a 2D screen.

C. 3D Rotations

A 3D object that just sits on the screen looking pretty gets boring after a while. It's a lot more interesting if you can have it moving, or maybe rotating. So how do we do that. Well, let me start by saying that the math involved here is pretty complicated. I'm no expert in it myself so I won't waste your time with a bad explanation. I'll just give you the final formulas and you can make spinning boxes all night.

First of all you will need the two angles that can be found in the diagram in section 1. They were theta(é) and phi(í). We will also need to deal with rho(p) which represents the distance to the point from the origin (0,0,0). Now I'll just jump straight to the formulas. In the formulas listed below, (xO, yO, zO) represent the original location of the point in 3D space and (xR, yR, zR) represent the rotated point.

xR = -xO * SIN(é) + yO * COS(é)
yR = -xO * COS(é) * SIN(í) - yO * SIN(é) * SIN(í) - zO * COS(í) + p
zR = -xO * COS(é) * COS(í) - yO * SIN(é) * COS(í) + zO * SIN(í)

This is the very basics of 3D within a 2D space. I hope this helps someone grasp it better and implement something good in PPL. - Check out my example in the Code Sharing section.

As there is sooo much else to do before the release of this newsletter, I have taken the above from an open source tutorial that I use as reference when doing my little bits of 3D. Because of this I take no responsibility for it being inaccurate.

When I have more time (In a later edition) I will re-visit 3D properly, by which time I should have some proper examples of 3D world creation to share with you all. - That will make for a really interesting article.

(The above is only intended to be an introduction to 3D geometry and should not be taken as the most correct way to do things) - Please examine the code for the 3D rotating cube that is in the archive that accompanies this newsletter for a better example of 3D. - Play with it, break it and have fun.

Something Interesting

There seems to have been a really interesting development for the PPC. With the release of the IPhone, everyone has gone gesture control mad.

HTC - purveyor of find Pocket PC devices has made the TOUCH an almost button less device (Their OS extension is very smooth)

Then along came Mr Nintendo and brought us the WII - For those of you who don't know (Where have you been for the last 12 months? - Probably queuing to buy one! Lol) the WII uses motion control handsets to play games with.

Well. - Although this sort of game play is not possible on PPC devices (No mercury switches or telemetry feedback) some clever programming has been done via the built in camera.

Check this out …

http://www.youtube.com/watch?v=MNq8Xcn1e70

This software is available from

http://www.concretesoftware.com/product/3dlawndarts.shtml

So we have .. Motion control, blue tooth multi player and forced feedback vibrations.

Amazing…

Now if only we can program the camera in PPL - Hmmmm… the possibilities

What should be in development

Here is my list of 10 things I think are possible with PPL or what I would like to see developed.

1. How to control the camera that is built into smartphones or PDAs, or web-cams on the PC.
2. Control of the internal Microphone for recording and sound analysis.
3. A good platform game in the style of JetPac on the Spectrum.
4. 3D Library or API for both PC and PPC—I am working hard to either find one, or code one.
5. Form based applications with maybe SQL and client server connectivity.
6. Large RPG - Xassar's Quest should fulfil this request for now! :)
7. Side scroller shoot'em up in the style of Cybernoid.
8. Vertical scroller in the style of Flying Shark.
9. More graphics demos released - I have some more in the pipeline!
10. Some sort of GPS application that has Bluetooth connectivity and mapping.

These are only ideas and I guess dreams .. Or are they.

Please please please, submit your software to the newly created online shop so that we can all benefit from our devices.

Also, don't stop submitting code snippets to the forum so that other users can get an insight into PPL coding and how to achieve remarkable programs.

(The top 10 list is mine and mine alone, it does not reflect the thoughts and views of anyone at Arianesoft! - Just in case you all thought Alain was pushing you to creating software!)

If I could do all of the above I would, but my time is limited and as I discover more about PPL and how stuff functions, I will release either the code or the executable for all to share!

[Mike]

See Ya' Next Year

It feels strange saying that, but its' true.

When the next issue of the newsletter is released it will be a whole new year.

A new year to start that mobile game or application that will make you loads of money?

A new year that will see V2 of PPL released (Hopefully)!

Hopefully the festive season will be pleasant for all of us with no worries of woes!

May good fortune smile on everyone in the coming year!

Stop Press:

Next months issue will include an optimisation tutorial by Zehlein!

Thanks again for sticking with us and PPL and take care until next month!

The ArianeSoft Team.