



NOVEMBER 2007 NEWSLETTER

IN THIS MONTHS ISSUE

EDITORIAL/NEW LOOK NEWSLETTER

WHATS' NEW IN THE LAND OF PPL

PBE4PPL—TUTORIAL

IN DEVELOPMENT

SEE YA' NEXT MONTH

Voice of the PPL



[Editorial/New Look Newsletter](#)

Welcome to the November issue of the PPL Newsletter. As you can see we have had a complete re-design to create a more professional document and a new name 'Voice of the PPL' (Like it eh?)

We have a new dedicated editor [Mike Halliday], who has taken over creation of format and content and who is committed to bringing you this new look each and every month.

To give us feedback on this new look, eMail info@arianesoft.ca with your comments.

We hope that as the newsletter matures and grows the content value will increase significantly, bringing you the latest news in the world of PPL programming.

We want your help....

Send us your articles and tutorials, code snippets and suggestions. The best ones will be featured in forth coming issues.

Also, if you have any tutorials, we want them too for the community!

Enjoy this new look newsletter which has been lovingly crafted by the PPL for the PPL.

Thanks

The Arianesoft Team



[Whats' new in the land of PPL](#)

Since it's initial release, PPL has grown more powerful each time an update has been issued.

Containing physics and particle engines, sprites and music facilities, PPL is ideally placed to allow creative individuals free raine to produce commercial quality games and applications.

V1.40 is due for release soon, and with it you can expect some of the following.

2 new GameAPI screen update modes. This will speed up some games immensely.

Added new Math.ppl library with functions to convert from and to Binary.

Fixed click in gutter area of text in PIDE to add breakpoint.

Added OnKeyPress event for all controls in PIDE.

Fixed case in form generator where two | characters could stay grouped together causing compile errors.

Fixed huge Windows API event call memory leak.

New g_SetInputOrientation() function to change the orientation of the mouse and keys input only.

New g_InputOrientation() function to retrieve input orientation.

New APE 2d physic engine with tons of new functions.

WOW ... Well done Alain, you have been working really hard to improve PPL and give us really useful extras to help create excellent games! [mike]

As you can see, PPL is maturing at an astounding rate - Is it's aim world domination? Who knows!

You will just have to keep reading future issues of
Voice of the PPL.



PBE4PPL

What is PBE4PPL?

Programming By Example FOR Pocket Programming Language is what we hope will be a monthly tutorial on getting the most from PPL. It will include examples or code snippets and ideas.

To start you off in the first issue we will look at the GameAPI.

The GameAPI allows you to easily create any sort of game from platform to side scroller to strategy and rpg.

An major almost total part of the GameAPI is the Sprite handling.

Dictionary term:

[Sprite] **An arbitrary-shaped bitmap that may be moved without applications having to redraw the underlying screen. Typically used for pointer cursors and for animated figures in games.**

Sprites are an integral part of any game, allowing character interaction and animated effects like explosions. The GameAPI has many sprite operations, many of which will be covered in later articles.

This issue will discuss SPRITE MOVEMENT—This tutorial is brought to you courtesy of one of our users [Donone].—Nice one, keep up the excellent work!

Next months issue will feature an article on the basics of game creation from planning and programming to testing and release so keep an eye out for it.

The tutorials section uses a different font from the rest of the news letter to make it easier to read source code snippets and examples.



Automatic Sprite Movement.

You may already know this or have deduced it for yourself. If not, hopefully it will save you a lot of searching and reading time.

Terminology.

The term **g_Speed** refers to the interval between calls to WM_TIMER and is measured in mS (millisecond, 1000th of a second), this is the main cycle control. Automatic (among other) actions are carried out at each of these cycles, so the longer the interval the slower the action, but remember this has a global effect on the program and covers all action that will be carried out by the engine in that 'cycle', not just one sprite.

The normal figure recommended for **g_speed** is about 5mS and this can be set in **InitGameApiEX** shown above. In fact if -1 is used the cycle stops.

The term **Velocity** refers to the number of pixels a sprite will move during each of the **g_Speed** cycles.

The speed and smoothness of operation relies on these two inter-reacting values

Some of the most common relevant commands.

It is often difficult to find the functions you require because you don't know if they exist in the first place, and if they do, what are they called?

The excellent and difficult to produce Help file will allow you to run down the lists of functions and you will find that you can spend a long time reading each (several times).

Occasionally differing terms are used to refer to the same thing, which may cause you to think that they act differently.



InitGameAPIEX (*, *, *, *, *, Speed, *); refers to g_Speed
InitGameAPI (*, *, Speed); refers to g_Speed
g_Init (*, *, *, *, *, AISpeed, *, *); refers to g_Speed
g_Speed (long Speed);
SetAISpeed (*, *, AutoMoveSpeed); refers to g_Speed
SetAISpeed (Speed, *, *); refers to g_Speed
SetSpriteVelocity (*, float Velocity); refers to sprite Velocity
SetSpriteVelX or Y (*, float Speed); refers to sprite Velocity
SetSpriteDirection (*, *, float Velocity); refers to vVelocity

Movement considerations.

A sprite cannot move other than in complete pixels.

Moving Sprites smoothly requires that each movement is a whole number of pixels per cycle (*which might seem obvious, but read on*).

To accomplish this the velocity of the sprite should use integers for numbers 1 and above. More about <1 later.

If integers are not used then what happens is that the numbers are added each cycle and then truncated to produce integers, but irregularly spaced integers.

For example...

Velocity	Relative movement (pixels)
1.5 = 1	1
1.5 + 1.5 = 3	2
1.5 + 1.5 + 1.5 = 4	1
1.5 + 1.5 + 1.5 + 1.5 = 6	2
1.5 + 1.5 + 1.5 + 1.5 + 1.5 = 7	1

of course the average over the repeat sequence (*in this case two cycles*) is 1.5 (wow).

This will lead to a jerky movement and even at fast speeds can be visible. If acceleration is used at the same time things may not look as good as you might expect, and you may spend a long time trying to track down the effect.



As an experiment (you could use the demo 'Simple2-1.ppl' to save effort) with `g_speed` set to 1000 then the update cycle is once per second (great if you want to see what goes on). If you then set the `velocity` to 1 you will see the sprite move at 1 pixel per second. Set `velocity` to 2 and it moves in jumps of two pixels per second.

The sprite does **not** move at 1 pixel per **half** second!

Finally, set Velocity to 1.5 as in the table above and you will observe the sequence 1,2,1,2,1.

To ensure smoothest working, regular single pixel movement is good if at all possible.

This means that if `g_speed` is effectively fixed at 5mS, visual speed and smoothness must be governed by `velocity` alone.

Slowing it down.

Slower velocities can be achieved in single pixel movements by using numbers <1 i.e. 0.2 or 0.7 which are in effect integers below 1. Therefore, as before, 0.15 will not be smooth.

For all these 'integers' <1 the sprite will still move one pixel at a time, it will just take longer between each movement. Instead of slowing down the whole system with a larger `g_Speed`, use velocities <1 .

i.e. Setting `velocity` to 0.2 (or one fifth) means that 5 cycles of 5mS (`g_Speed`) must pass before the sprite moves one pixel. But it **will** move one pixel.

This could be considered the same as saying that the sprite moves one fifth of a pixel per 5mS cycle (but it doesn't actually move until the fifth cycle).

Other time settings could be used for `g_speed` but it affects the whole of the program cycle, whereas velocity is specific to each sprite.

Did you get that? - Thanks Donone for a great tutorial on Sprite Movement within the GameAPI.—I'm sure you will all find it useful.

Think you can do better? Send us your tutorials and we will include them in these pages.



[In Development](#)

In development will feature PPL projects that are being created by our users. This is YOUR chance to give a project or game exposure to the outside world.

Anyone in the PPL community can use this section to promote their creations.

You can submit your details to mike.halliday@dsl.pipex.com in the following format [to avoid spam and to aide organisation]

To: mike.halliday@dsl.pipex.com

Subject: PPL Development Submission

Attachments: .jpg, .bmp, .gif only—no more than 4 individual images

You can attach demos for review but they must be submitted in.zip format and must be virus free! Any hint of Virus or SPYWare is deleted by my mail scanner immediately and you will miss your chance to show off your coding expertise.

Ideally they should work on all formats but they must work. This is not a debugging service!!!

Hopefully by the next issue we will have some interesting things to show you all. I don't want to be the only one showing off what I create!

So here is your chance to show of and brag I suppose! :)

‘Send em by the bucket load!’

You can all check out my OLDSKOOL demo in the ‘Made with PPL’ section of the web forum to see what I had in development as I was writing this newsletter.

[Mike]



See Ya' Next Month

Well that's it for the first 're-issue' of the PPL newsletter.

Arianesoft hopes that you enjoyed it.

Next Months Issue

With Christmas just around the corner, expect Santa to bring some glad tidings to the PPL Community and this news letter!

YEY! - Its Christmas... - Pressie time!

Boo! Its Christmas.... - Time to buy pressies! :(

I have an interesting article from the Pocket PC World that should tickle your fancy and some Ideas to share with you for games that I think should be re-made for Pocket PC, so if you are stuck for ideas, check back next month and I will be your muse to give you inspiration.

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

The ArianeSoft Team.