The monthly magazine for the PPL community Issue 12 - May 2008

Voice of the PPL

Index

CONTENTS

INFORMATION

Editor, Design and Layout: Mike Halliday © 2008
Arianesoft President: Alain Deschenes
Development support: Eric Pankoke (with thanks)

Do you have a product created in PPL that you want to share with the community?

Drop me an Private message on the FORUM - I'd love to hear from you!

Submission deadlines:

20th of each month for next edition

When you have finished with this magazine, please recycle it.

A word or two from our Editor

It is 7 months since I took over the reigns as editor of the newsletter. In that time, I have seen some impressive submissions to the web site and to the tutorials section of this newsletter.

Sadly, I still have to ask for content. There does not seem to be a ready supply of willing users sending me items for inclusion. I have a very good submission for the coming months (by NickNack) [many thanks for that!] so some users are helping out!

As a last ditch attempt to bring back the enthusiasm, I am asking, nay begging…

PLEASE - has anyone any code samples, tutorials ANYTHING!!!!!

I really don't want to see the newsletter go the way of the DODO, just as we are getting it looking professional and smart, but without content its just blank pages!

Anyway...

Enjoy the rest of this newsletter and thank you to all those users who have submitted code and tutorials in the past and for coming newsletters.

Mike Halliday.

PPL Release Dates

The following versions of PPL have been released in the last 6 months

Date	PPL Version	Date	PPL Version
30/11/07	1.40	07/03/08	1.51
20/12/07	1.41		
24/12/07	1.42		
29/12/07	1.50		

Current PPL Community User count

At the time of writing (MIddle of May), the current PPL registered user count in the community forum is;

Interesting PPL Development #1

Brainstorming is something I try and do on a regular basis, either to increase my wealth (yeah right) or to generate new programming ideas for PPL.

Well .. .. Recently I had a another retro moment…

Why are there no great point and click adventures for the mobile environment? Sure you can get mobile SCUMM to play the old lucas arts titles, but there are no new adventures to play.

So I have set myself a challenge of creating a flip screen/point and click adventure in the Monkey Island vein. (I have resorted to using the MI© GFX until I design my own)

Whether this comes to completion or not I cannot say, but it has been written down in my little black book for future reference, and I have started creating the basis of the …

PointnClick Engine

Currently, it allows movement between screens, but that's about it!

Eventually, it should support pointing and clicking objects, animated characters and interactive story.

Progress will be posted on the forum and in the forthcoming newsletters.

Mike.

Interesting PPL Development #2

PPL is soo flexible. :)

Here is something I found, which I think makes my life a bit easier when using the GAMEAPI

If you are using the GAMEAPI example files as reference and not using your own G_INIT routines then it is most difficult to change the orientation of the display. The INITGameAPIEX calls default to NORTH orientation.

TIP:

If you edit GAMEAPI.PPL from the RUNTIME/LIB folder you can add another INITGameAPI function to allow user changeable orientation (as below)

```
proc InitGameAPIEx(hWnd$, Proc$, Width$, Height$, FullScreen$, Speed$, FpsSpeed$, Orientation$)
  _init(hWnd$, Proc$, width$, height$, Orientation$, Speed$, FpsSpeed$, fullscreen$, true);
end;
```

By adding this extra function, you are able to specify 0, 1, 2, 3 for North, South, East and West screen orientation.

This should result in the device orientation changing to your required direction.

The only issue is that when the next version of PPL is released, the GAMEAPI.PPL file will be overwritten, with the new version!

Tutorial

Signing your life away! Pt 2

PBE4PPL - Programming By Example for PPL!

As a follow on from last months tutorial on creating a signature input area, capturing and saving the signature, it makes sense to show you all how to re-load and display the stored signature.

These 2 tutorials should allow you to create many sorts of data capture applications and maybe even some sort of character recognition application.

Step 1: Load Signature file
Step 2: Display signature in the signature capture box

And that's it! Really simple isn't it?

Step 1: Load the signature.

In the WINMain function I have created the new form and set it to the size required - In this example 240 x 320 but could be any size you like.

Added the colours to the form and objects then create the event 'ONCREATE'

Next I added just 1 other control - the Bitmap control. As in the create signature module it is blank with text overwriting it, but in this instance we do not need to create 'on click' or 'mouse move' events.

The actual loading of the signature is completed at FORM creation time, but you could include it on a button click event or a menu selection event!

In FUNC FORM100_Create()
Set up the storage array then load the file

```
filename$ = AppPath$ +"signature.dat";
if(FileExists(filename$))
  f$ = fopen(filename$, "r"); //Open the file
  sigData$ = readstring(f$); //Read a line
  fclose(f$); //Close file
else
  ShowMessage("Signature File Missing!");
end;
```

In the ONPaint function of the Signature (Bitmap control) event we can now display the signature over the bitmap control;

Start the paint process
Get the client window structure
Create a rectangle object in the bitmap control
Fill it with a background colour
Delete the object from memory

```
GetClientRect(hWnd$, &r$);
b$ = CreateSolidBrush(RGB(100, 100, 100));
FillRect(hdc$, &r$, b$);
DeleteObject(b$);
```

Set text foreground and background colours
Create a new TTF font object
Load in the "Arial" font and set it to 40 pels/pixels
Select the new font
Write the SIGNATURE text to the object control
Delete the object from memory

(Cont …)

Continued ...

```
SetTextColor(hdc$,RGB(120,120,120));
SetBKColor(hdc$,RGB(100,100,100));
fnt$ = NewFont(Signature$, "Arial", 40, 0, 0, 0);
SetFont(hdc$, fnt$);
SelectObject(hdc$,fnt$);
DrawText(hdc$, "SIGNATURE", 9, &r$, DT_CENTER | DT_VCENTER | DT_SINGLELINE);
DeleteObject(fnt$);
```

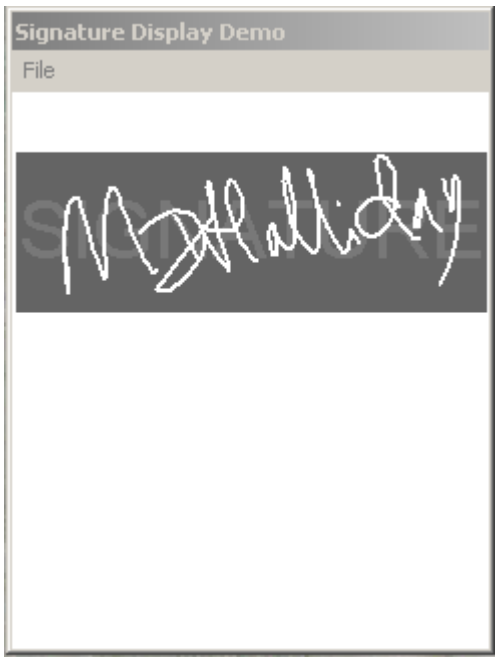
Create a new PEN in white for writing the signature
Store the currently selected object for later use
Select the PEN as the current object
Read in the data from sigData$ in pairs of 2 for the hex numbers
    Convert each pair of hex numbers back to the decimal X & Y co-ordinates
    Plot a line from the last points loaded to these new ones
Repeat until the end of the data stream
Delete the pen object and set the original object back as the current

```
sigPos$ = 0;
While (sigPos$ < Length(sigData$))
  // Convert the hex encoded signature back to X-Y Co-ordinates
  sigBits$ = MID(sigData$, sigPos$, 2);
  PlotX$ = Hex2Dec(sigBits$);
  sigBits$ = MID(sigData$, sigPos$ + 2, 2);
  PlotY$ = Hex2Dec(sigBits$);
  If (PlotY$ > 127)
    PlotXold$ = PlotX$;
    PlotYold$ = PlotY$ - 128;
  Else
    If (PlotX$ > PlotOldX$ AND ploty$ > plotoldY$) // Check if we have reached the end of the signature
      MoveToEx(hdc$, PlotXold$, PlotYold$, NULL); // add an offset here for display at different
      LineTo(hdc$, PlotX$, PlotY$);               // places on the client screen
      PlotXold$ = PlotX$;
      PlotYold$ = PlotY$;
    end;
  End;
  sigPos$ = sigPos$ + 4;
end;
SelectObject(hdc$, op$);
DeleteObject(p$);
```

Finish the paint process

And that's it! You should now have a signature in your bitmap control box that looks something like this!

(This is not my real signature, so don't even think about using it for ID Theft purposes - ha ha)



A useful addition to this module would be to allow the saving of the signature to a bitmap image file.

That concludes this months tutorial (Part 2 of 2)! - I hope you found it insightful and will find the principle useful in your future projects.

Next month .. A tutorial from NICKNACK on '2D Destructable Terrain creation' - ala Worms©

Ideas

Ideas Generator. - Do you have a bright one?

The Ideas generator, apart from sounding like some futuristic pop group is somewhere I hope will help you think of games or applications to create using PPL.

Fuzzball.

How about a basic football game? With CPU against human, bluetooth human against human and demo modes?

eMail Client.

How about a fully functional eMail client in the vein of Thunderbird© or of your own design?

Client/Server.

Full Client/Server app with front end data submission and back end data storage/retrieval. - This would be a good tutorial for the newsletter and a completion of the WINSOCK requests we have been getting on the forum.

FTP Client.

Forms based FTP client that will allow simple file transfer—This would make a good WINSOCK and forms tutorial rolled into one.

(More Ideas next month)

Tutorial #2

This month I thought I would be generous and include another tutorial. It is more of a work in progress and is very incomplete, but because of all the recent questions about WINSOCK tutorials, it seemed a good idea to include.

I am constantly checking my inbox just to see how many messages are waiting for delivery. I use OUTLOOK © and sometimes, the notification icon just does not work. This got me wondering if I could write something in PPL that would show me how many messages are available. This I thought would lead to an application sat on the task bar with pop ups or icon notification. (That will be way in the future!)

Your mail server may respond in a different way than mine did, but you can output all the received bytes to the Console window to see the exact responses.

Here are the steps I used; - I called my app (MAILWATCHER)

1. Contact Pop Server
2. Read back response
3. Send Login details
4. Read number of messages
5. Close connection

1. Contact Pop Server

```
popserver$ = "<POP SERVER ADDRESS>";
popport$ = 110;
popuser$ = "<pop username>";
poppwd$ = "<pop password>";

// Open an internet connection.
sock$ = SW_Socket(typ$);
hConn$ = SW_connect(sock$, typ$, popserver$, popport$);   // connect to server
if (hconn$) // successful connection - Check for connection
```

2. Read back response

```
   // Read 1024 bytes form server  (1024 bytes is a little large, but it is a catch all size to make sure we get all the info)
   z$ = SW_Receive(sock$, 1024);
```

3. Send Login Details

```
  // send username
   hsend$ = "USER " + popuser$ + crlf$;  (Carriage Return and Line Feed are needed after each line sent back to the server)
   sent$ = SW_Send(sock$, hsend$, length(hsend$));
   // Read 1024 bytes form server
   z$ = SW_Receive(sock$, 1024);
   hsend$ = "PASS " + poppwd$ + crlf$;
   sent$ = SW_Send(sock$, hsend$, length(hsend$));
```

(Cont…)

4. Read Number of Messages

```
// Read 1024 bytes form server
z$ = SW_Receive(sock$, 1024);
sent$ = SW_SEND(sock$, "STAT" + crlf$, 6);
writeln("Checking mail");

 // Read 1024 bytes form server
z$ = SW_Receive(sock$, 1024);

// Take off the first 4 chars (+OK ), then find the next space
// anything in between is the number of eMails waiting
m$ = MID(z$,4,length(z$));
NoWaiting$ = MID(m$,0,POS(" ", z$)-1);

writeln("eMails waiting for delivery:" + NoWaiting$);
```

5. Close connection

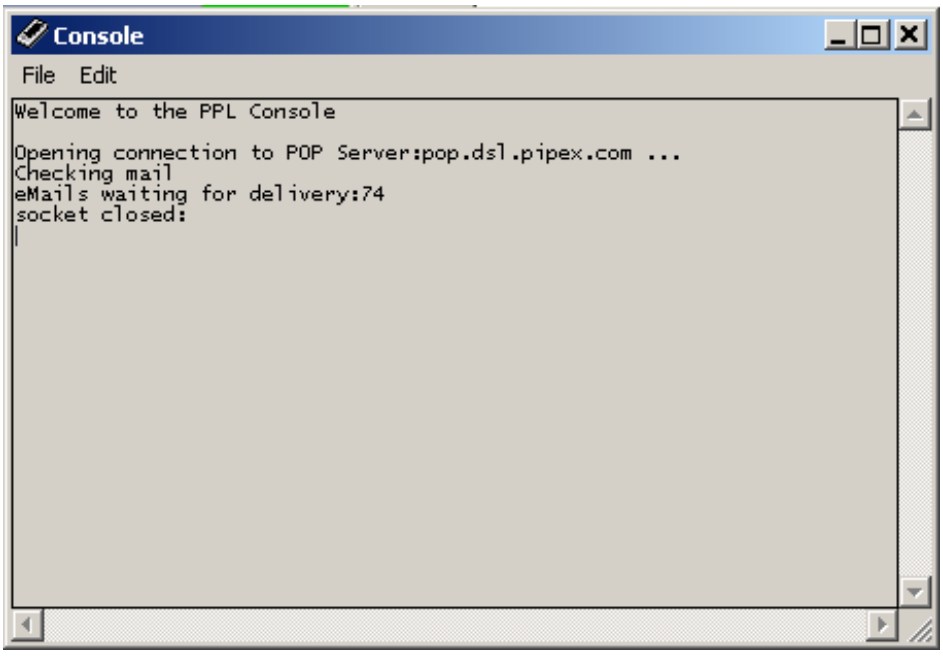
```
sent$ = SW_SEND(sock$, "QUIT" + crlf$, 6);

// Close connection.
SW_CloseSocket(sock$);
writeln("socket closed:");
```

The above example is a very basic test of the PPL Winsock functionality. It CAN be expanded upon to create all sorts of applications.

Check out the full source code included with this months newsletter.

The output should look something like this;



Next months issue!

Coming in next months issue …

Another colour change next month, it will however still reflect the Arianesoft colour scheme!.

Part 1 of a 5 part landscape/terrain tutorial from NickNack. This should enable you to create Worms™ or Tanx style games. - I'm really looking forward to that, as I hope you all are?

As ever I will throw as much content into the newsletter as I can, and I am always on the look out for new items to add.

Join me next month for another edition of Voice of the PPL to catch up on the further developments of the PPL community and development scene.

Regards

Mike & the Arianesoft Team