OpenVG 1.x SDK

User Guide

Filename	:	OpenVG 1.1 SDK.User Guide.mht
Version	:	1.1.2.1f (PowerVR SDK 2.03.23.1196)
Issue Date	:	30 Oct 2008
Author	:	PowerVR

Contents

1. OpenVG 1.1 SDK Content

1.1. Introduction

The PowerVR OpenVG SDK provides a set of documentation, source code and utilities that help developers create applications using the OpenVG graphics library on PowerVR platforms. This document describes the contents of the SDK and gives guidelines for installing it on different platforms.

1.2. Documents

These documents cover some technology overviews, performance recommendations and specifications.

Documents are located in `\SDKPackage` folder or in `\SDKPackage\Documents`.

Name	**OpenVG SDK.User Guide**
Description	Description of the OpenVG SDK contents and Installation.

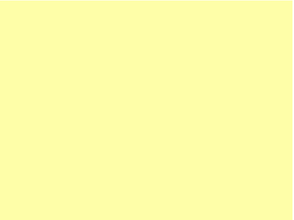
Name	**PowerVR.OpenVG Application Development Recommendations**
Description	Some recommendations to get best performance on PowerVR devices.

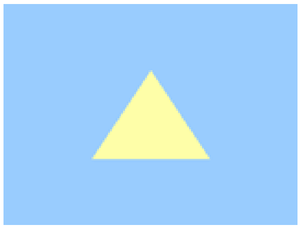
1.3. Training Course

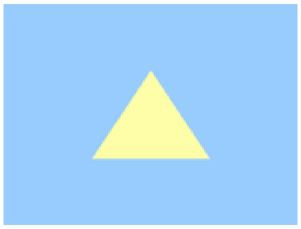
Training Course is composed of several simple applications to show specific features in a simplified form. The code has been thoroughly commented to help developers to understand the API and get started.

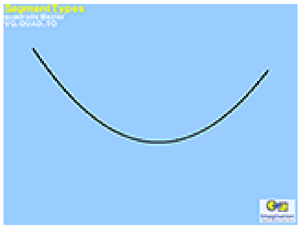
All Training Course demos are located in the `\SDKPackage\TrainingCourse` folder.

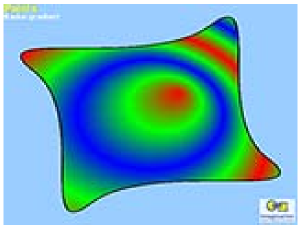
Note: The demos in the TrainingCourse do not handle screen rotation to keep the code as simple as possible. In devices with a portrait display the images below might be shown stretched.

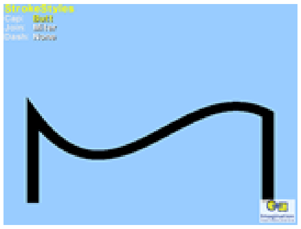
	Initialization
	Shows how to initialise OpenVG. It does a simple background clear.

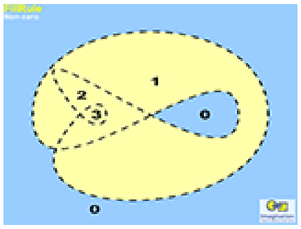
	HelloTriangle
	Shows how to draw a very simple shape.

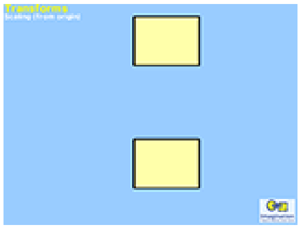
	IntroducingPVRShell
	Shows how to use PVRShell for the OS and API initialisation framework. Uses the same geometry used in the previous example.

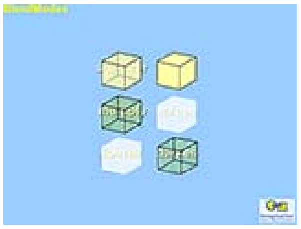
	SegmentTypes
	Shows the different path segment types supported by OpenVG. This demo will cycle through all the 11 different types.

	Paints
	Shows the different paint types that are used to add colour to strokes and filled shapes. This demo will cycle through all the 3 different types of paint: solid colour, linear gradient and radial gradient.

	StrokeStyles
	Shows different styles that can be applied to strokes. This demo is interactive and there are three options which can be selected by the user: caps, joins and dash.

	FillRule
	This training course shows you how to use the fill rules in OpenVG. The fill rules are used to determine whether an overlapping part of a path is filled or not.

	Transforms
	Shows how to apply different transformations to position and move shapes. This demo cycles through translations, scaling, rotations and shear.

	BlendModes
	Shows how to use different blend modes in OpenVg. To do this it shows 6, each cube is identical except for the blend mode they use. Behind each cube is a text string showing the name of the blend mode that it is using.

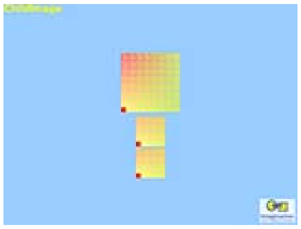
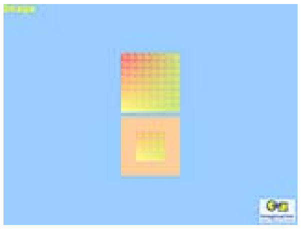
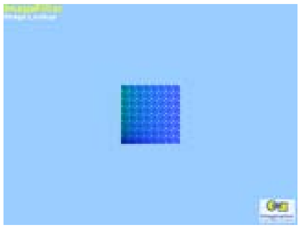
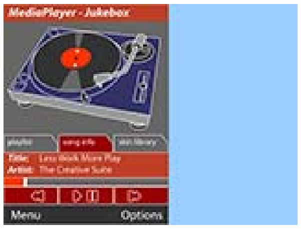
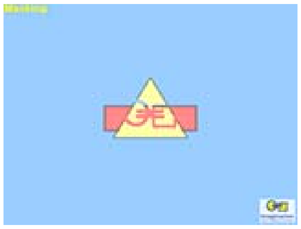
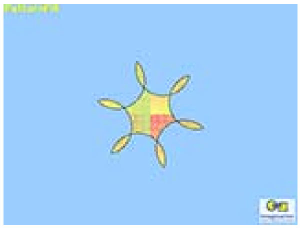
	ChildImage
	Shows how to use OpenVG child images. It starts off with the creation of a single image that is then used to spawn a child image. This child then spawns its own child. The first child is then modified and this modification can be seen in all its relatives.

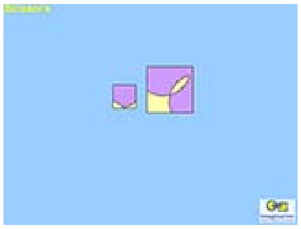
	Image
	Shows how to use OpenVG images. It starts off with the creation of two images. The first is populated with image data from memory. The second image is initially cleared to a single colour and then a section of the first image is copied to it.

	ImageFilter
	Shows how to apply several of the inbuilt filters in OpenVG. The filters shown are the Gaussian blur, convolution, separable convolution and an image lookup.

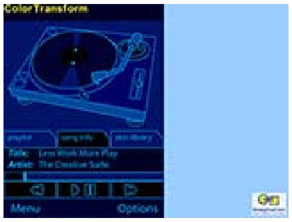
	IntroducingPVG
	Basic Tutorial that shows step-by-step how to load a PVG file (exported from Illustrator) into OpenVG

	Masking
	Shows how to use an OpenVG image to set up the alpha mask.

	PatternFill
	Shows how to take an image and use it to pattern fill a shape.

	Scissors
	Shows how to set up and enable scissor rectangles so you can limit the area that you can draw to.

	IntroducingSVG
	Shows how to load an SVG file using the loading code in our tools.

	ColorTransform
 ColorTransform playlist song info skin library Title: Less Work More Play Artist: The Creative Suite Menu Options	Basic Tutorial that loads a pvg file and shows how to use OpenVG 1.1's colour transformation to alter the way it is displayed.

	Font
Font Texty! Texty! Texty! Te Texty! Texty! Texty! Texty! Texty! Texty! Texty Texty! Texty! Texty! Te Texty! Texty! Texty! Texty! Texty! Texty! Texty	Shows how to load path based and image based glyphs into a vgFont and use them to display text to the screen.

	MaskLayer
	Shows how to use OpenVG's MaskLayers.

	RenderToMask
	Shows how to use vgRenderToMask..

1.4. Shell

The PowerVR Shell is used in all demos to provide a common framework for developing OpenVG applications. Across all platforms, the shell takes a set of command-line arguments which allow things like the position and size of a demo to be controlled. The table below shows these options.

The Shell source is located in the `\SDKPackage\Shell` folder.

Table 1 - Shell Command-Line Options

Option	Description
-width=N	Sets the viewport width to N.
-height=N	Sets the viewport height to N.
-posx=N	Sets the x coordinate of the viewport.

Option	Description
-posy=N	Sets the y coordinate of the viewport.
-FSAAMode=N or –aa=N	Sets full screen anti-aliasing. N can be: 0=no AA , 1=2x AA , 2=4x AA
-fullscreen=[1,0]	Runs in fullscreen mode (1) or windowed (0).
-powersaving=[1,0]	Where available enable/disable power saving.
-quitaftertime=N or –qat=N	Quits after N seconds.
-quitafterframe=N or –qaf=N	Quits after N frames.
-vsync=N	Where available modify the apps vsync parameters.
-version	Output the SDK version to the debug output.
-info	Output setup information to the debug output.

1.5. Tools

The tools library consists of a collection of various modules to help developers achieve common tasks, e.g. mathematical operations, matrix handling, svg loading, pvg loading and loading VGImages.

The Tools library is located in the `\SDKPackage\Tools` folder.

1.6. Utilities

These are utility programs or libraries useful for openvg application development.

All utilities are located in the `\SDKPackage\Utilities` folder.

Name	PVRTexTool
Description	Tool to convert bitmap files (e.g. BMP, TGA, etc.) to any image format type supported by OpenVG.

Name	PVRVecEx
Description	Plug-in for Adobe Illustrator to export path data as the PowerVR PVG format.

Name	FileWrap
Description	Command-line utility to wrap external files in .cpp files.

2. OpenVG SDK Installation

See below the SDK installation and build notes for all platforms supported by PowerVR cores. Please, refer to the platform you are using.

2.1. Linux

1. Copy the driver files (libOpenVG.so, libOpenVGU.so, libEGL.so , libIMGegl.so and libsrv<version>.so) from your platform support package or driver package to the Builds/OVG/<platform>/Lib directory.
2. Install the latest platform toolchain in your Linux system.
3. Change Builds/OVG/<platform>/make_platform.mak to point to your platform toolchain (the path is at the beginning of the file).
4. To build individual components go to directory Demos/<DemoName>/OVG/Build/LinuxGeneric or TrainingCourse/<TCName>/OVG/Build/LinuxGeneric and run command “make PLATFORM=<platform>”.
5. The executables for the demos and training courses will go to:
   Demos/<DemoName>/OVG/Build/<platform>/Release
   TraininCourse/<DemoName>/OVG/Build/<platform>/Release respectively.
6. Be sure that the PowerVR drivers are in your discimage (please refer to DDK / Driver installation instructions).
7. If the standard c++ libraries are not present on your discimage, copy libc++* from the toolchain into /usr/lib on the discimage, libdl and libgcc may also be required.
8. Ensure the drivers are running (e.g. type /etc/init.d/rc/pvr start).
9. Copy the executables to your ‘discimage’ to run them from your platform system.

2.2. Symbian

1. Install the latest Symbian release.

   Note: In the following [SYMBIAN] is used to refer to the Symbian installation directory.

1. To build individual components go to the /Build/SymbianTextShell sub-directory of the component to build and type:

   bldmake bldfiles

   abld build <platform> urel

2. The exe will be automatically copied in the correct place in the Symbian SDK tree to be included in the image through the OBY file.
3. Pre-built binaries can be found in the Binaries folder ready to be dropped into your Symbian SDK
4. Refer to your platform BasePort documentation to know how to build an image and how to download it to the platform. We supply in the package the OBY files that will allow you to include the SDK demos in the image.

Note: Due to a Symbian DevKit problem with long path names, the TrainingCourse demos will not build out of the package. Please, install SDKPackage/ in your root path and rename it to a shorter name (e.g. SP/) to be able to build.

2.3. Windows Mobile 6.1

1. Download and install the Windows Mobile 6.1 SDK you wish to use (Standard or Professional) from the following location: http://www.microsoft.com
2. Unpack the contents of the SDK to a local folder on a PC equipped with the Windows XP operating system. The project and solution files provided (.vcproj and .sln, for Microsoft Visual

Studio 2005) do not require the SDK to be installed in a pre-defined location and are configured to use relative-paths.

3. The OpenVG libraries should be in the folder Builds/OVG/WindowsMob61ARMV4I/Lib.
4. To build the Demos and Training Courses in Visual Studio make sure the “Active solution platform” is set to either “Windows Mobile 6 Professional SDK (ARMV4I)” or “Windows Mobile 6 Standard SDK (ARMV4I)” depending on the Windows Mobile 6.1 SDK you installed.

2.1. Windows Mobile 7

1. Firstly, open up a command prompt and setup your Windows Mobile 7 build environment.
2. Set the environment variable SDKROOT to the root of the installed SDKPackage

```
e.g. set SDKROOT=c:\SDKPackage
```

3. To build all SDK apps at the root of the SDKPackage type

```
build –c
```

4. To build an individual app navigate to the application’s folder and type

```
build –c
```

The makefiles for the majority of apps are kept in [app]\OVG\Build\PlatformBuilder and the executables are built in [app]\OVG\Build\PlatformBuilder\oak\target\ARMV4I\retail.

3. Support Contact

For further support contact:

devtech@powervr.com

Imagination Technologies Ltd.
FAO: Developer Technology
Home Park Estate
Kings Langley
HERTS WD4 8LZ
United Kingdom

Tel: +44 (0) 1923 260511
Fax: +44 (0) 1923 277463

For more information about PowerVR Technologies or Imagination Technologies Ltd. visit our web pages at:

www.imgtec.com
www.powervr.com
www.powervrinsider.com