RenderWare Graphics

Examples

A Summary

Contact Us

Criterion Software Ltd.

For general information about RenderWare Graphics e-mail info@csl.com.

Developer Relations

For information regarding Support please email devrels@csl.com.

Sales

For sales information contact: rw-sales@csl.com

Introduction

This document gives an overview of all the examples and demos provided with installations of RenderWare Graphics, dPVS and the FX Pack.

Each of these applications is provided in the form of source code and an executable for a particular platform. The "readme" file that accompanies them provides more information. It is called "<*platform name*>.txt". So xbox.txt, win.txt are the "readme" files for Xbox and for Microsoft Windows D3D8 respectively. The "readme" file can be displayed by pressing the spacebar (or the BACK button on the Xbox).

The examples are based on a common code framework called the skeleton. Some examples require extra toolkits or plugins to achieve their effects. Such toolkits and plugins are listed in the API Reference and above the description of their examples in this document.

All the examples can be built using the shipped `.mcp` and `.dsp` files in CodeWarrior and Visual Studio. The project files have been built using the version numbers of the IDEs specified in the RenderWare Graphics readme. (i.e. for RenderWare Graphics' Xbox the readme is called `readme_xbox.pdf`).

Some examples are designed only for a particular platform. These platform specific examples are detailed in the supplementary examples document that is provided for each platform.

Examples

ALPHASRT

The Alpha Sort example displays some cylinders, cones and a sphere, with a degree of transparency as defined by their alpha values (RGBA). They are sorted according to their distance from the camera to render them efficiently. This method delays the rendering process by overriding each atomic's render callback.

ANIM

`RtAnim`

The Anim example illustrates how to use the animation toolkit, `RtAnim`, to create a new interpolator type that animates the colors and radius of lights. It also illustrates how to create and play animation data for the new interpolator type.

BASEBALL

`RpTeam` `RpCrowd`

This is the "demo" for plugins sold separately from RenderWare Graphics, as part of the FX Pack. The Baseball demo shows a baseball stadium populated with a crowd and a team while a game is in progress. The crowd is a series of animating billboards to give the impression of large numbers of people and the team demonstrates many skinned objects with different LODs and separate animations.

BLENDING

`RtPNG`

The Blending example shows two transparent colored squares, the first on the far clip plane and the second on the near clip-plane. The user alters their colors, transparency and blend mode to become familiar with these effects on screen.

BOUNDING

`RwBBox`

The Bounding example illustrates methods of calculating a clump's bounding-box and bounding-sphere, both in clump-space and world-space.

CAMERA

`RwCamera RtPNG`

The Camera example creates a main camera and a sub-camera through which the user can experiment interactively with the effects of changing the camera frustum settings.

CAMTEX

`RtPNG`

The Camera Texture example illustrates how a camera's frame buffer can be used as a texture to render another surface. It sets up two cameras, the first points at a rotating cube. Its raster buffer is then applied as a texture to the sides of a second cube. The image of the second cube is what is displayed on screen, as it is seen by the second camera.

COLLIS1

`RpCollis`

The first Collision example allows the user to control a camera in a bounding sphere, as it moves over a terrain. The example allows the camera to rise above but never below the hilly surface. Alternatively it can keep the camera moving across the terrain while remaining in contact with it.

COLLIS2

`RpCollis`

The second Collision example represents a ball bouncing, within its world geometry, which is also a sphere. The optional effects include gravity, and a damping effect when the ball bounces.

COLLIS3

`RpCollis`

The third Collision example approximates a representation of a series of balls moving between rotating cogwheels. Its collision detection is shown to become much more efficient by using the menu options to isolate only the relevant surfaces for detection.

DECAL

`RpPTank RpCollis RtPick`

The Decal example demonstrates how to efficiently implement and manage decal textures using `RpPTank` and `RpCollis` in RenderWare Graphics. Decal textures are usually used to display effects such as bullet holes, stains on walls, etc.

DMORPH

`RpDMorph RpGeometry RpHAnim RtWorld`

The Delta Morph example uses two different geometry objects to illustrate the two ways in which delta morph targets can be used – a human face which is animated, and a curved surface which can be manipulated – both have a base geometry and two morph targets.

dPVS Easy

`RpDPVS`

The "dPVS" plugin is sold separately from RenderWare Graphics. The "dPVS Easy" example integrates dPVS (dynamic Potentially Visible Sets) into a standard RenderWare Graphics application merely by linking `rpdpvs.lib`, attaching the `RpDPVS` plugin, calling `RpDPVSOpen()`, and appending `rpdpvs.h` to the header file list. This calls dPVS functions in place of the default ones, showing marked efficiency savings, with no other input from the developer. The example allows the user to use standard controls to explore a terrain with a densely filled enclosure of mushroom shaped dwellings. The controls will be familiar from other examples. The developer can un-define `USE_DPVS` in `main.h`, and rebuild the example, to see how fast the code runs without using dPVS.

dPVS Full

`RpDPVS`

The "dPVS" plugin is sold separately from RenderWare Graphics. The previous "`dPVS Easy`" example invoked dPVS automatically. The "`dPVS Full`" example uses the same landscape and mushroom enclosure, but calls API functions directly to apply them more intelligently. This gives better control of the occluding algorithms. The example also toggles several diagnostic displays that show how individual occluding algorithms process the view while the user explores the environment interactively.

dPVS View

`RpDPVS`

The "dPVS" plugin is sold separately from RenderWare Graphics. The "dPVS Viewer" example uses almost the same code as "dPVS Full", but it allows the user to choose a file containing a world to "drop" onto the Example's window. It loads the file and allows the user to explore the world represented in the file using the controls familiar in other examples. In essence, the code is the same as the world viewer application's code but uses dPVS for efficiency. The viewer will follow a spline path if one is provided with the file. As with the "dPVS Full" example, the view can be rendered in different ways to see how efficiently the dPVS algorithms can handle it.

FOG

The Fog example simulates fog allowing the user interactively to alter the types of fog, its distance from the camera, its density and color.

FRAME

`RwFrame`

The Frame example shows the frame hierarchy of a series of atomics in a clump. Each frame affects its children, so any frame that is rotated rotates all its child frames with it. Each atomic has its own frame and can be selected and rotated. The root atomic and clump share the same frame. In the example, the selected frame is represented by highlighting the bounding box of its atomic.

GEOMETRY

`RpGeometry`

The Geometry example shows how to generate clumps dynamically rather than importing them. The vertex data is used to create a geometry, the geometry is used to create an atomic, and finally a clump. This clump can then be treated as if it had been exported.

GCOND

`RtGCond RtWing`

The GCond is used to show how to create a geometry conditioned world from an import world (building on the World example). Geometry Conditioning has a set of parameters that can be used to affect the conditioning in a number of ways. The example is a simple coplanar patch tesselated into 15x15 triangulated quads. Each quartile's UV coordinates are shifted to show how UV translation affects conditioning. The world is displayed on the screen and the results of conditioning can be displayed in wireframe.

HANIM1

`RpHAnim RpSkin RtAnim RtQuat`

The first Hierarchical Animation example shows how to use `RpHAnim` to blend from the end of one animation with the start of another over a period defined by the user in seconds.

HANIM2

`RpHAnim RpSkin RtAnim RtQuat`

The second Hierarchical Animation example shows how to use `RpHAnim` for delta animation. It loads a hierarchy of a base clump and delta clumps and animations that represent a figure running and a hierarchy of the same figure with its head is turning. The example produces a third hierarchy to store the calculated results and shows the figure doing both at once.

HANIM3

`RpHAnim RpSkin RtAnim RtQuat`

The third Hierarchical Animation example shows how to modify an `RpHAnimHierarchy` over the top of a standard animation, and do it procedurally. It includes the interactive scaling of bones to adjust the proportions of a character, and the movement of bones to be controlled independently of the animation.

HANIM4

`RpHAnim RtAnim RtCmpKey RtQuat`

The fourth Hierarchical Animation example illustrates how to use compressed keyframe data to animate `RpHAnim` hierarchies.

HANIMKEY

`RpHAnim RpSkin RtAnim RtQuat`

The Hierarchical Animation Key example shows how to register and use `RpHAnim`'s custom keyframe interpolation scheme. A new, economical keyframe type is used to support animations that require rotations but not translations. Once created, the rotation-only animation may be saved and re-loaded using the standard `RpHAnim` interface.

HANIMSUB

`RpHAnim RpSkin RtAnim RtQuat`

The Hierarchical Animation Sub-Hierarchy animation shows how a base animation updates an entire model while other animations replace parts of the hierarchy so that animations can be mixed at runtime.

IM2D

`RwIm2D`

The 2D Immediate Mode example demonstrates each immediate mode primitive using a different geometry. The menu options allow the user to select the type of object to setup and render, and its use of color and texture.

IM3D

`RwIm3D`

The 3D Immediate Mode example demonstrates each type of immediate mode primitive using a different geometry. The immediate mode object can be rotated, translated to or from the camera, textured and colored interactively.

IMAGETEX

`RtPNG`

The Image to Texture example shows the correlation between different image types, from `.png` and `.bmp` files, the textures that are generated from them and how the texture's raster type can vary across different platforms.

IMGFORMT

The Image Format example illustrates how to extend the SDK to read other image file formats. In this particular example read and write functions for the TGA image file format have been implemented. To show that the image has been loaded correctly it is textured onto a simple rotating cube. (Four formats are already supported by the toolkits `RtBMP`, `RtPNG`, `RtRAS` and `RtTIFF`.)

IMLIGHT

The Immediate Lighting example shows a simple lighting system for use with the 3D immediate mode API. The object illuminated can be rotated and the shading and light source or sources can be controlled interactively.

IMPICK

The Immediate Mode Picking example demonstrates an interface to pick, or select, vertices using the mouse or equivalent controls on a console. The points can be dragged in a plane parallel to the camera's view window.

IMSHADOW

`RpCollis RpHAnim RpSkin`

The Immediate Mode Shadow example shows how to render parallel-projection shadows. A camera follows a runner, showing a preview of its shadow in a corner of the display. Every frame the shadow raster is used as a texture, projected onto ground level and blended, to appear as a shadow.

LIGHTMAP

RpCollis RpLtMap RpMipmapKL RpPVS RtBary RtBMP RtIntSec RtPNG

The LIGHTMAP example demonstrates the use of the RtLtMap toolkit and RpLtMap plugin. For a scene composed of world sectors and atomics, it uses RtLtMap to generate lightmaps (as you watch), lit by point and area light sources and then uses RpLtMap to display the scene as it would appear in-game. Lightmaps may be streamed to disk in platform-independent or platform-dependent texture dictionary formats.

LIGHTS

RpLight

The first Lights example allows the user to experiment with different light types: ambient, directional, point, spot and spot-soft. Each type of light can be modified using the menu options including cone angle, color and optional ambient light.

LIGHTS2

RpLight

The second Lights example demonstrates the effects of different global and local lights on a landscape. The lights used are an ambient, a directional and a point light. Point light is local, the others are general. The user can reposition the world and the lights, and other values can be adjusted.

LODATOM

RpLODAtomic

The Level Of Detail Atomic example shows how to manage the LOD using its plugin. It loads some clumps, each of which represents a torus at a different LOD or mesh resolution. As the model moves away from the camera, the appropriate LOD geometry is calculated and rendered. The model can also be rendered as a wire-frame.

MAESTRO1

Rt2d Rt2dAnim

The maestro1 example shows how to use Maestro to effect as a 2d front end.

MATFX1

RpMatFX

The Material Effects example creates and manipulates clumps with

- environment maps

- bump maps
- combined bump map and environment maps
- dual textures
- single texture with UV transforms
- dual textures each with separate UV transforms

The user can control many variables interactively, including its shininess, bump-mapping, material and light source.

MIPMAP

The Mipmap example loads a clump that uses two textures, one that uses auto-generated mipmaps and one that uses mip-level images down to 1 x 1. The user can move and rotate the clump interactively to compare the effects at different distances and angles.

MORPH

`RpGeometry RpMorph`

The Morph example uses `RpMorph` to show a geometry and a texture that display the world as a globe. It stores three other "morph targets", which, in this case, are stretched globe shapes. Then it morphs continuously, from globe to a morph target and back.

PAKFILE

The PAK file example illustrates how PAK files can be integrated into an application. PAK files are used to create a virtual file system in memory thus reducing the time taken to load individual files.

PATCH

`RpPatch RtBezPat`

The Patch example shows how quad patches can be read in to define a teapot (and covered with a surface texture) and how tri patches can be coded programmatically to represent a segment of a sphere.

PICKING

`RpCollis RtPick`

The Picking Example demonstrates how atomics can be selected (picked) interactively. Picking is performed by casting a line from the near clip plane to the far clip plane and finding the closest atomic that intersects this line.

The user can choose one of two variations on this method. One detects atomics whose bounding spheres intersect the line and highlights the bounding sphere. The other is more accurate because it detects atomics whose triangles intersect the line and highlights the atomic's bounding box.

PLUGIN

The Plugin example demonstrates how user-plugins can be used to extend RenderWare Graphics objects.

PRTADV

`RpPrtStd RpPrtAdv RpPTank`

This is the example for the `RpPrtAdv` plugin sold separately from RenderWare Graphics, as part of the FX Pack. The PrtAdv example illustrates the use of the `RpPrtAdv` plugin. It shows three different particle systems using the advanced properties provided by the `RpPrtAdv` plugin.

- Point List Emitter
- Circle Emitter
- Sphere Emitter

PRTSTD

`RpPrtStd RpPTank`

The PrtStd example illustrates the use of the `RpPrtStd` plugin by creating a default particle system with the following standard properties :

- Box emitter
- Color animation
- Size animation

PRTSTD 2

`RpPrtStd RpPTank`

The PrtStd2 example illustrates the use of a custom property with the `RpPrtStd` plugin. It shows how a custom emitter and particle property can be created and used with other standard property in the `RpPrtStd` plugin.

PTANK 2 and PTANK 3

`RpPTank`

These PTank examples show how to create a PTank. The particles are positioned as if they were placed around the surface of an invisible, rotating dough nut. All of the particles change together according to parameters that the user can experiment with.

The user can see the effect of the alpha blending and source blending values and the number of particles can be altered interactively through the menu. The other parameters can be turned on or off by adding or removing a final "x" to the name of one of the "`#define`" statements at the beginning of the file `ptank.c`.

RWSVIEW

`RtAnim RpHAnim RtPITexD`

The RWSView example illustrates how to load assets from an RWS file into your RenderWare Graphics application. The supplied RWS file contains a world, a platform independent texture dictionary, two clumps, and two animations.

SUBRAST

The Sub Raster example shows how sub-rasters create a multiple-view application. Four sub-cameras as well as the main camera are created each showing a different view in a separate rectangle within the main window. These sub-rasters display different views in different ways at the same time.

TEMPLATE

The Template example is a skeleton project that the user can adopt as a starting point for development.

TEXADRSS

`RwTexture`

The Texture-Addressing example shows the use of different texture-addressing modes. It loads a clump that uses three textures. The user can then alter the texture-addressing mode for the entire clump, altering the u and v addressing modes independently. The implementation is platform-dependent and modes that are not supported are removed from the list of available options.

TOON

`RpMorph RpRandom RpToon`

The Toon example is sold separately from RenderWare Graphics as part of the FX Pack. It example illustrates how to load and render toon models. All the models were exported from 3ds max with toon enabled. It also demonstrates how to create and apply toon inks and paints programmatically or from dictionaries. The menu options allow the user to cycle through different toon clumps, to alter their line characteristics in various ways and to apply different shading textures.

WORLD

`RtWorldImport RtWorld`

The World example shows how to create a world from an import world and display it on the screen. The import world describes a world as a collection of vertices and triangles, and with a set of conversion parameters can be created into an `RpWorld` object and generate the world data. The conversion parameters are used to control how the world is created.