RenderWare Graphics

White Paper

Portal Culling

Contact Us

Criterion Software Ltd.

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

Contributors

RenderWare Graphics development and documentation teams.

Table of Contents

1. Introduction

Portal culling is a technique used to reduce the amount of geometry processed when rendering a closed environment such as the inside of a building. Portal culling takes advantage of the extremely high occlusion and the static nature of the environment, setting up portals that describe the "doorways" between open areas. The rendering engine only needs to render the sections of the world that are visible through the portals, which will improve performance considerably. This is usually achieved by manipulating the camera frustum. The RenderWare Graphics graphics library provides tools that can be used to implement portal culling easily.

2. Setting Up A Scene

To take advantage of portal culling, the game world needs to be partitioned into discrete sections that are delimited by portals. The portals define a view from within one section to another, connecting the sections together. A portal can define a translation that will map co-ordinates in the world-space of a section on one side of the portal to the world-space of the section on the other side of the portal. In this way, all sections do not have to share the same origin or orientation, allowing easier construction of levels.

Scene Structure

In RenderWare Graphics, the easiest way to create a world that takes advantage of portal culling is to create a map of `RpWorld` objects, connected by application-defined portals.

The `RpWorld` object needs to be extended (using a plugin) to include pointers to all the portals that lead from it.

The information each Portal object needs to contain is:

- A description of its geometric position;
- A pointer to the `RpWorld` it leads to;
- The transformation from the co-ordinate space of the world the portal is "in" to the co-ordinate space of the world it leads to.

A decision can be made as to whether a single portal object should be used to describe the transition between worlds in both directions (as in Fig 1), or if a separate portal would be defined for each direction (as in Fig 2). The latter method allows for some interesting effects such as one way teleports.

Figure 1: Bi-directional portal

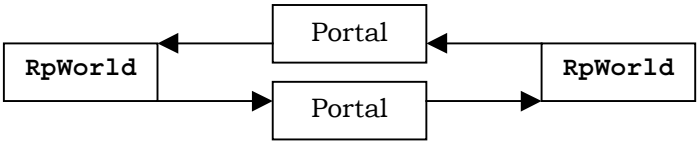

Figure 2: Uni-directional portals

Constructing Portal Data

In RenderWare Graphics, the portal data and world map can be generated in one of three ways; runtime generation, during export, or as a post-processing stage after export. These are explained below.

Runtime Generation

If the world is small, it may be best to create the portals and the world map at runtime. This is the simplest and quickest method. Data describing the portals can be hard-coded or loaded in from a file and used to generate portal objects.

During Export

The artwork exporter could be customized to partition the world and generate portals either automatically or according to cues placed in the artwork. This method would require considerable effort to implement successfully.

Post-Processing

The artwork would be exported as separate BSP files that would then be connected using portals and re-serialized, ready to be imported into the application as a complete map of `RpWorld` objects and portals. This approach should be implemented with a plugin to allow portals and extended `RpWorld` objects to be serialized and de-serialized effectively.

3. Rendering Using Portals

The algorithm for rendering a scene using portal culling is as follows:

- Start at the `RpWorld` object the camera is currently attached to.
- For all portals in this `RpWorld` which are currently in the camera view frustum:
  - Calculate the sub-raster that encloses the portal in screen-space and is contained within the raster used to render the current `RpWorld`.
  - If the sub-raster does not have width or height of 0 pixels:
    - Remove the camera from the current `RpWorld` and attach it to the new `RpWorld` through the portal, transforming the camera into the co-ordinate system of this new `RpWorld` using the transform specified in the portal.
    - Set up the camera view window, offset and raster to render just the enclosing sub-raster of the portal.
    - Recursively render the world through the portal using this algorithm
    - Restore the camera's view window, offset and raster.
    - Re-attach the camera to the original `RpWorld`, transforming it back into the original co-ordinate system using the inverse of the transform specified in the portal.
- Render this world using `RpWorldRender`.

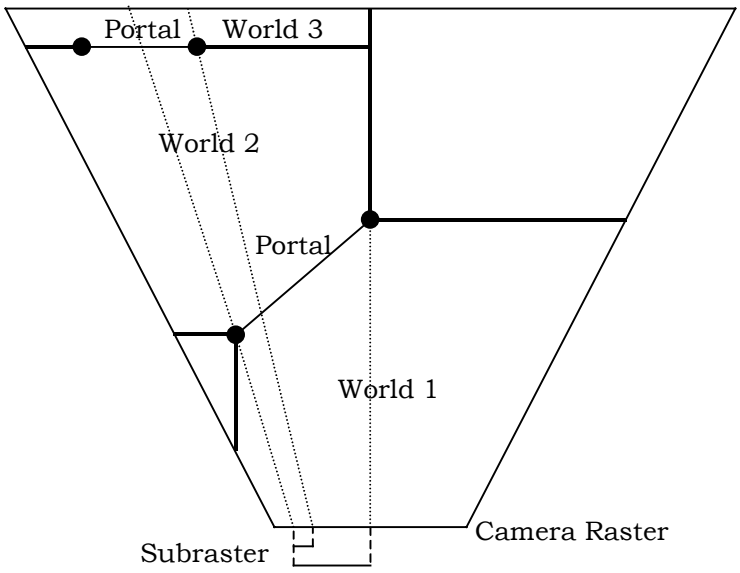

Figure 3: Portal culling example.
Note how the first portal delimits the second portal's view frustum.

4. Moving Through Portals

A consideration when using a portal culling scheme is that game characters are going to move through the portals as they travel around the world. The player will need to be removed from one `RpWorld` and placed in the next as they move through portals. This is further complicated when using a third-person perspective, where the camera needs to move with the player. It is possible that the two objects will need to be moved synchronously whilst in two distinct co-ordinate systems.

5. Lighting

Care will need to be taken if dynamic lighting is being used in a scene. Geometry in one world will not be affected by local lighting in another world. This means that local lighting will appear to stop suddenly at the portal threshold. The method used to prevent this effect is to detach lights from their native world and add them to the world currently being rendered, remembering to transform them using the connecting portals transform matrix. The lights will need to be re-attached to their original world after rendering has finished.

6. Mirrors

Portal culling can also be very easily applied to the task of creating mirrors. Setting up a portal that leads back into the same world it is in will achieve this effect. A transform that performs a scale by –1 through the normal to the portal is required.

7. Further reading

There are Internet resources available to those interested in portal culling:

http://www.flipcode.com

Provides a good deal of information on various schemes using portals.

http://www.cs.virginia.edu/~luebke/publications/portals.html

An interesting paper on portal culling including reference to portals as mirrors.

http://www.gamedev.net/reference/programming/features/culling/

An discussion of various cell-based culling techniques.