RenderWare Graphics

White Paper

Camera View Matrix

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

RenderWare Graphics uses a matrix called the Camera View Matrix. This matrix transforms coordinates in the world coordinate system to RenderWare Graphics camera space. This white paper outlines the steps taken to create the camera view matrix. It should be understood that this matrix is re-computed for every frame, by the `RwCameraBeginUpdate` function. In RenderWare Graphics terminology this is called syncing the camera.

2. Creating the View Matrix

RenderWare Graphics will combine a number of simpler transformation matrices together to create the view matrix. These are shown in the equations below.

$$\mathbf{C} = \mathbf{TOSPX}$$

where

$$T = \begin{pmatrix} -r_x & u_x & a_x & 0 \\ -r_y & u_y & a_y & 0 \\ -r_z & u_z & a_z & 0 \\ p_x & -p_y & -p_z & 1 \end{pmatrix} \quad O = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ o_x & o_y & 1 & 0 \\ -o_x & -o_y & 0 & 1 \end{pmatrix}$$

r	right vector
u	up vector
a	at vector
o	view offset
w	screen width
h	screen height

$$S = \begin{pmatrix} \frac{1}{2w} & 0 & 0 & 0 \\ 0 & \frac{1}{2h} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

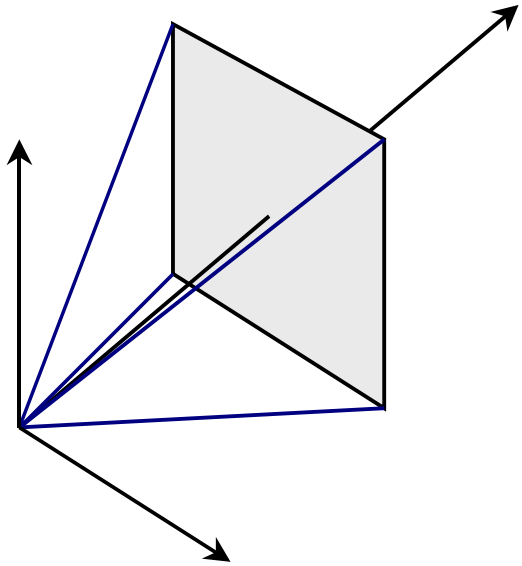
$$X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 1 \end{pmatrix}$$

It can be seen that 5 distinct matrices are used to compute the camera view matrix. The first of these, T, is the re-orientation matrix that converts into a 3D coordinate system oriented with the camera looking in a positive z-direction, and located at the origin.

Matrix O performs the offset necessary if the application has established a view offset using `RwCameraSetViewOffset`. Similarly, S resizes camera space, adjusting it for the view window that a call to `RwCameraSetViewWindow` establishes.

The P matrix maps from a 3D coordinate system to a camera frustum, and hence is a projection. (For a parallel camera model, this matrix is replaced by a zero matrix with a leading row of (1,-1,1,1)).

Finally, matrix X applies a shear to the coordinate space. The camera frustum in RenderWare Graphics occupies only the first octant, effectively transforming from a frustum defined by the origin and a unit square centered at the point (0,0,1) to one defined by the origin and a unit square centered at (0.5, 0.5).



3. Implementation

It should be noted that the concatenation of these matrices is not performed as explained in this white paper. The implementation is optimized and the view matrix computed directly, and not using matrix multiplication. Source code licensees can see the code for the camera sync functions in `src\bacamera.c`.