Graficación por Computadora

OpenGL

- Un elemento importante en la creación de escenas en el exterior es la de tener el cielo.

- Existen muchas técnicas para el rendereado del cielo.

-En particular se verán:

Skyplanes
Skyboxes
Skydomes.

Rendereado del cielo

OpenGL

- Un elemento importante en la creación de escenas en el exterior es la de tener el cielo.

- Existen muchas técnicas para el rendereado del cielo.

-En particular se verán:

Skyplanes
Skyboxes
Skydomes.

Rendereado del cielo

OpenGL

Skyplanes

- Es la simulación mas básica del cielo.

- Se realiza con el uso de planos.

- Es la técnica mas fácil pero necesita ajustes para lograr verse bien.

Rendereado del cielo

OpenGL

Skyplanes

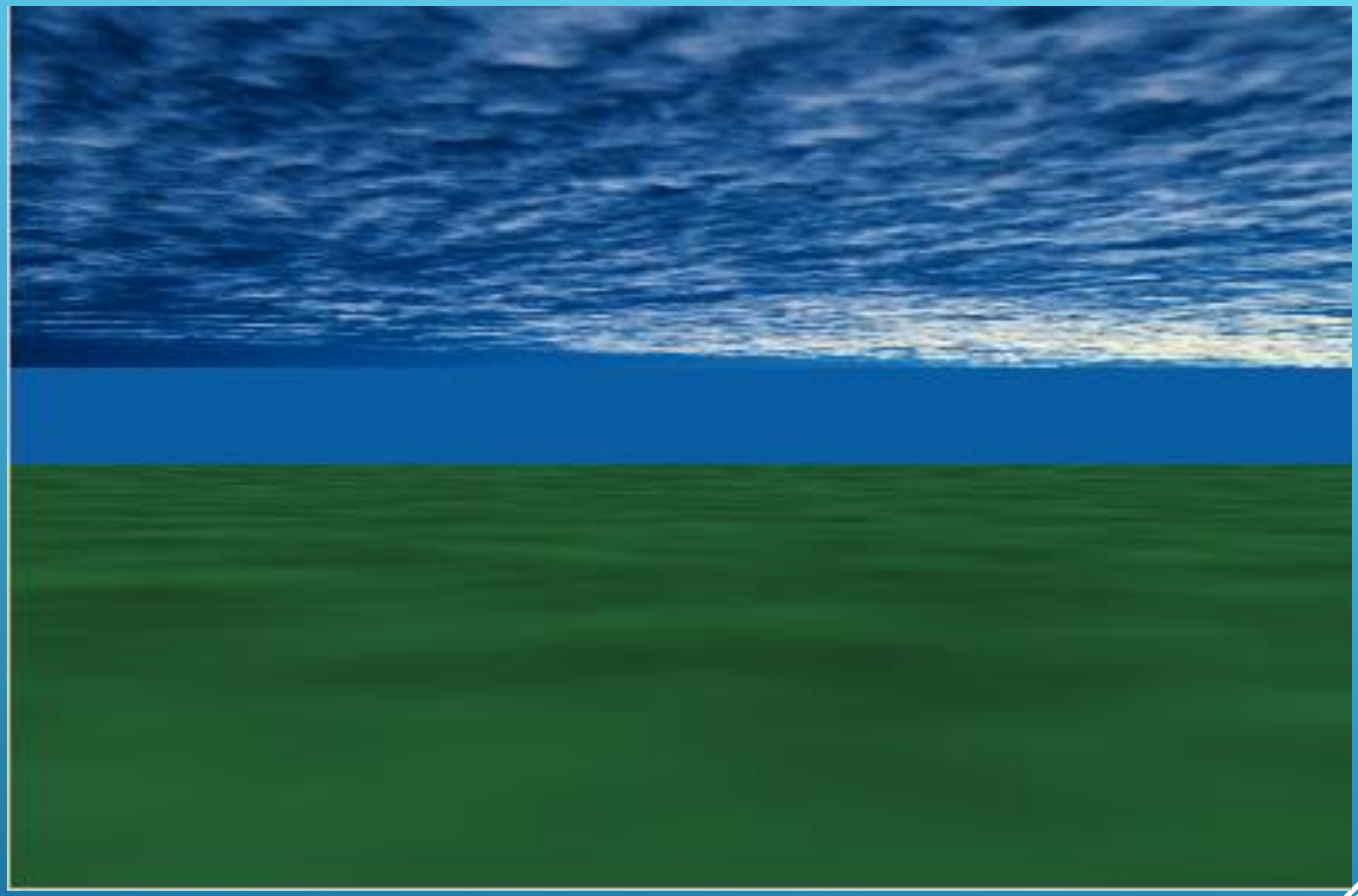
- En la forma mas simple solo se tiene un plano que es texturizado usando pequeñas texturas o una gran textura.

- El plano se coloca para cubrir la escena global.

- El problema con usar este enfoque es que se puede identificar el final del plano.

Rendereado del cielo

OpenGL



Rendereado del cielo

OpenGL

Skyplanes

-Esta técnica no encierra la cámara completamente.

-Se debe usar con ambientes donde el piso, montañas, objetos u otra cosa cubran la porción del fondo no que se llena.

Rendereado del cielo

OpenGL

Skyplanes

-Una alternativa para mitigar el problema es usar un plano curvado.

-El plano es dividido en triángulos y las esquinas son jalados hacia abajo, produciendo una forma de paracaídas.

Rendereado del cielo

OpenGL

Skyplanes

Los principales atributos geométricos que se pueden especificar cuando creamos un plano así son:

1. La resolución del plano(la división en triángulos).

2.El radio del circulo que envuelve los skyplanes.

3. La altura de la cima del skyplane.

4. Las repeticiones de la textura mapeada en el plano.

Rendereado del cielo

OpenGL

Skyplanes

-El plano final consiste de vértices e índices que indexan el arreglo de vértices y crean el triangle strip.

-Este numero depende de las divisiones especificadas. Cada una nos da 4 vértices 2 triángulos y 6 índices.

- NumVertices = (Division + 1) * (Division + 1)
  NumIndices = (Division * Division) * 2 * 3

Rendereado del cielo

OpenGL

Skyplanes

-El radio del circulo y la altura de la cima definen su extensión y curvatura.

-Conociendo el radio del plano se puede calcular el tamaño de los lados

Planesize = sqrt($(2 * radius)^2 / 2$)

Rendereado del cielo

OpenGL

```
float planesize = (float)sqrt(((2.0f * radius)*(2.0f * radius) )*0.5f );
float delta = planesize / (float)divs;
for(i=0; i<= divs; i++){
        for(j=0; j<= divs; j++){
                xdist = (-0.5f * planesize) + ((float)j * delta);
                zdist = (-0.5f * planesize) + ((float)i * delta);
                zheight = xdist * xdist;
                zheight = zdist * zdist;
                 Height = (xheight+zheight)/radius_squared;
                SV.x = xdist;
                SV.y = (1.0f – height) * peakHeight;
                Sv.z = zdist;
                SV.u = htile * ((float)j*textdelta);
                SV.v = vtile * ((float)i*textdelta);
                PlaneVertices[i*(divs+1)+j] = SV;}}
```

Rendereado del cielo

OpenGL

Skyplanes – Creación

```
//Calculo de los índices
Int index = 0;
for(i=0; i<= divs; i++){
        for(j=0; j<= divs; j++){
                int starvertex = (i * (divs+ 1)+j);
                // Triángulo 1
                Indices[index++] = startvertex;
                Indices[index++] = startvertex+1;
                Indices[index++] = startvertex+divs+1;
                // triangulo 2
                Indices[index++] = startvertex+1;
                Indices[index++] = startvertex+divs+2;
                Indices[index++] = startvertex+divs+1;
                        }
}
```

Rendereado del cielo

OpenGL

Skyplanes – Creación

- xdist y zdist calculan los puntos en el plano.

- Están en el rango [-planesize/2,planesize/2].

-El centro del plano es el origen, delta y textdelta son valores incrementales para la generación de vértices y coordenadas de textura.

Rendereado del cielo

OpenGL

Skyplanes – Rendereado

- Para renderear el skyplane, hay que deshabilitar los estados que no se necesitan.

- Depth testing puede deshabilitarse también para asegurar que no ocurran intersecciones.

Rendereado del cielo

OpenGL

Skyplanes – Rendereado

Para mezclar el skyplane con el fondo se usan loas funciones:

glBlendFunc(GL_SRC_ALPHA,GL_ONE_MINUS_SRC_ALPHA);

glBlendFunc(GL_ONE, GL_ONE)g;

Una vez asignado lo anterior el rendereado del skyplane es directo.

Rendereado del cielo

OpenGL

Skyplanes – Rendereado

```
glBegin(GL_TRANGLES);
        for(i=; i < NumIndices;i++){
                glTexCoord2f(PlaneVertices[indices[i]].u,
                                                PlaneVertices[indices[i]].v);
                glVertex3f(PlaneVertices[indices[i]].x,
                                        PlaneVertices[indices[i]].y,
                                        PlaneVertices[indices[i]].z);
}
glEnd();
```

Rendereado del cielo

OpenGL

Skyplanes – Desvanecimiento

- El skyplane puede verse poco natural si la escena no cuenta con objetos o montañas.

- Aun falta el efecto de desvanecimiento en la distancia.

- Se debe habilitar y especificar un valor alfa para cada vértice.

- Los vértices mas alejados tienen un valor menor y se desvanecen totalmente.

Rendereado del cielo

OpenGL

Skyplanes – Desvanecimiento

```
for(i = 0; i< divs;i++){
        for(j = 0; j< divs;j++){
                //calculo de las coordenadas geométricas y de textura
                //Calculo del valor alfa por vértice.
                SV.alpha = (xheight + zheight)/alpha_radius;
                SV.alpha = 1 -SV.alpha;
                if(SV.alpha  < 0.0f)
                        SV.alpha = 0.0f;
                //Calculo de las coordenadas de textura usado antes
                PlaneVertices[i*(divs+1)+j] = SV;
        }
}
```

Rendereado del cielo

OpenGL

Skyplanes – Desvanecimiento

- alpharadius_squared es el radio especificado por la transparencia.

- Todos los vértices mas allá de este radio son totalmente transparentes.

- El radio de transparencia no puede ser mayor que el radio del skyplane.

Skyplanes – Desvanecimiento

- La transición de la transparencia es lineal desde el centro al borde exterior del plano.

- Se puede usar una función exponencial para ajustar como el valor alfa es calculado para cada vértice.

```
If( exponencialFadeout)
        SV.alpha = SV.alpha * SV.alpha  * SV.alpha ;
```

Rendereado del cielo

OpenGL

Skyplanes – Animación

- Los skyplane simulan una capa estática de nubes.

- Se puede mejorar los skyplanes con animación.

- La forma mas común para la animación de skyplanes es la de animación de texturas.

- Simula el movimiento de las nubes.

Skyplanes – Desvanecimiento

- Usando la matriz de texturas su puede animar las texturas de un objeto.

```
glMatrixMode(GL_TEXTURE);
glPushMatrix(ttrans[0], ttrans[1],0.0f);
RenderSkyplane();
glPopMatrix();
glMatrixMode(GL_MODELVIEW);
```

- La translación es especificada por ttrans[0] y ttrans[1].

Rendereado del cielo

OpenGL

Skyboxes

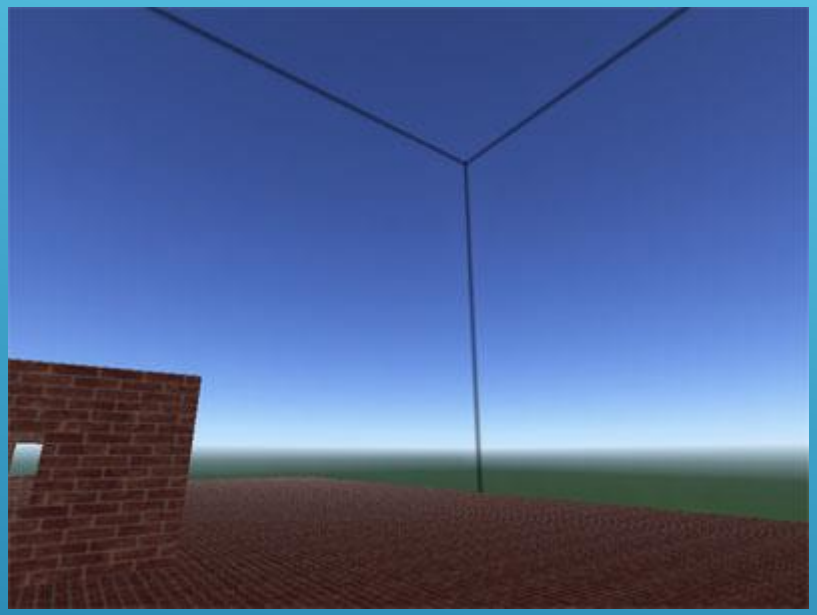
- Es una técnica que sugiere una gran ambiente.

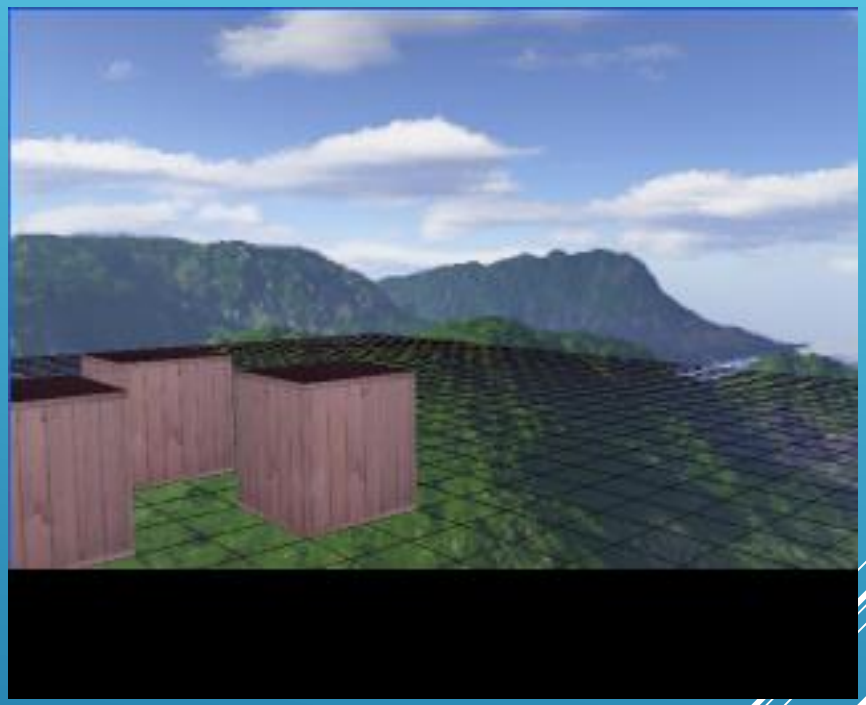
- Puede crear la ilusión del ambiente y elevar la sensación de inmersión.

- El escenario es rendereado en 6 texturas, las cuales son aplicadas a los lados de un cubo.

Rendereado del cielo

OpenGL





Rendereado del cielo

OpenGL

Skyboxes

- La cámara es colocada en el centro del cubo.

- La distancia a los muros del cubo debe mantenerse constante.

- Podemos rotar libremente, pero **nunca** se debe alcanzar los muros.

Rendereado del cielo

OpenGL

Skyboxes - Rendereado

- La técnica mas simple para el rendereado de un skybox es el de renderear un cubo ordinario texturizado alineado con el eje del centro y colocando la cámara en el origen.

-El skybox es lo primero que se debe renderear.

- Esto significa que no se debe limpiar el color buffer.

Rendereado del cielo

OpenGL

Skyboxes - Rendereado

- La escritura en el depth buffer y el depth testing deben deshabilitarse.

- La niebla, iluminación y otras operaciones que alteran la imagen final debe ser deshabilitadas.

- Las imágenes del skybox debe renderearse sin efectos atmosféricos.

Rendereado del cielo

OpenGL

Skyboxes - Rendereado

- El tamaño del skybox no importa mientras este dentro del volumen de visión.

- Las dimensiones deben escogerse de tal forma que los puntos mas lejanos(las esquinas) estén mas cerca de la cámara que el plano de corte far.

Rendereado del cielo

OpenGL

Skyboxes - Rendereado

- $\text{width} <= \frac{(2\sqrt{3})}{3} * \text{zfar}$

- zfar es la distancia del pano de corte far.

- Las dimensiones del cubo no tiene que ser idénticas, la altura puede ser mas pequeña.

Rendereado del cielo

OpenGL

Skyboxes - Rendereado

Los pasos para el rendereado del skybox son:

1.- Limpiar el depth buffer.
2.-Deshabilitar depth test.
3.-Deshabilitar la escritura en el depth buffer.
4.-Deshabilitar niebla e iluminacion.
5.-Renderear la caja.
6.-Habilitar depth test y la escritura en el depth buffer.
7.-Habilitar niebla e iluminacion.
8.-Dibujar el resto de la escena.

Rendereado del cielo

OpenGL

Skyboxes – Problemas con las aristas del skybox

- Ciertos problemas surgen en el renderado del skybox.

- El proceso de filtrado produces manchas en las aristas de la caja, lo que arruina la ilusión.

- Este problema es causado por la interpolación lineal de los texeles, usando GL_LINEAR como filtro.

Skyboxes – Problemas con las aristas del skybox

-Deshabilitar la interpolación y siempre usar el texel mas cercano, usando GL_NEAREST, resuelve el problema.

-Esto produce otros problemas como bloques y brillo excesivo.

- La razón de que la interpolación cause problemas en las aristas de una textura es una inadecuada envolvente de la textura.

Rendereado del cielo

OpenGL

Skyboxes – Problemas con las aristas del skybox

- Los píxeles en las aristas del skybox no tiene vecinos en todos los lados.

-El comportamiento por omisión es usar el texel de la arista opuesta(GL_REPEAT). No sirve para este caso.

- OpenGL tiene el modo de restricción GL_CLAMP, el cual interpola los texeles en las aristas a un color especifico.

Skyboxes – Problemas con las aristas del skybox

- Esta técnica introduce discrepancias en las aristas porque la interpolación de los ejes con el color del borde, causa diferentes coloraciones en los texeles.

Skyboxes – Arreglando el problema con las aristas del skybox.

- Una solución fácil para la restricción cuando se renderea el skybox fue introducida en la version 1.2 de OpenGL con el modo GL_CLAMP_TO_EDGE.

- Este modo no incluye los texeles de las aristas cuando interpolamos.

Skyboxes – Posicionando el skybox

- Cuando el skybox es posicionado correctamente, el usuario nunca podrá alcanzar el final de este.

- La caja debe seguir a la cámara para mantener la distancia a sus muros constante.

- La matriz de modelview de OpenGL puede usarse para hacer que el skybox siga a la cámara.

Rendereado del cielo

OpenGL

Skyboxes – Posicionando el skybox

- Después de que las transformaciones de la cámara se aplican, la matriz del modelview se lee nuevamente y la parte de la translación(entradas 12,13 y 14) es asignada a (0,0,0).

- Se carga entonces la matriz a la pila de OpenGL.

Skyboxes – Posicionando el skybox

Los pasos específicos son los siguientes:

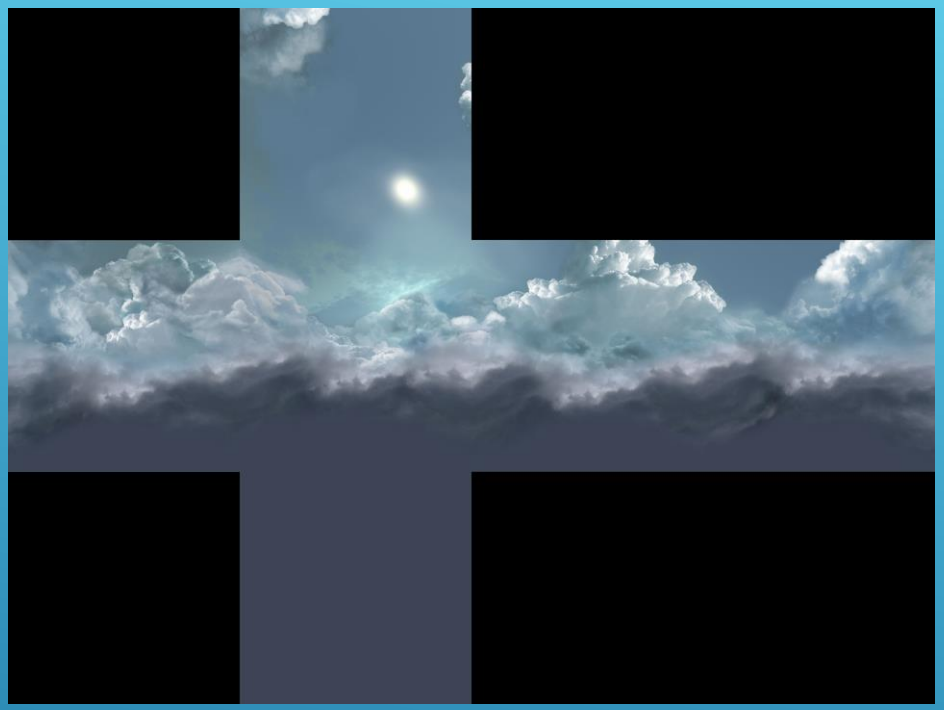
```
gluLookAT(...);// transformación de la cámara
glPushMatrix();
glGetFloatv(GL_MODELVIEW_MATRIX,mat);//obtenemos
Mat[12] = mat[13] = mat[14] = 0.0f;
glLoadMatrix(mat);//translación a (0,0,0)
DrawSkybox();
glPopMatrix();
//Se dibuja el resto de la escena
```

Rendereado del cielo

Skyboxes – Creación de texturas

- Hay varios métodos para la creación de texturas de skybox usando diferentes paquetes.

- También hay herramientas que producen las texturas, como terragen.



Rendereado del cielo

OpenGL

Skyboxes – Creación de texturas

El procedimiento general para crear las texturas necesarias es lo mismo para cada paquete, y es el siguiente:

1.-Modela la escena.
2.-Asigna la cámara.
3.-Asigna el campo de visión. 90 grados.
4.-Rendereado de las 6 imágenes usando diferentes valores de pitch(rotación lo largo del eje X) y heading(rotación a lo largo del eje y).

Rendereado del cielo

OpenGL

Skyboxes – Creación de texturas

Valores de pitch y heading para las distintas imágenes del skybox.

Imagen	Pitch	Heading
Frente	0	0
Derecha	0	90
Atras	0	180
Izquierda	0	270
Topo	90	0
Fondo	270	0

Rendereado del cielo

OpenGL

Skyboxes – Creación de texturas

- El tamaño de la textura usado debe ser de un texel por cada píxel.

Imagewidth = screenResolution/(tam(FOV/2))

- La ecuación calcula la resolución ideal para una resolución de
- pantalla y campo de visión.(FOV) específicos.

Rendereado del cielo

OpenGL

Skyboxes – Multicapa

- Se pueden extender los skyboxes añadiendo el uso de multiples capas y animación.

- Estas técnicas se usan para mejorar el fondo estático.

- Los skyboxes multicapas pueden implementarse rendereando skyboxes de diferentes tamaños y mezclarlos usando una función especifica de blending.

Rendereado del cielo

OpenGL

Skyboxes – Animación

- Para implementar la animación puede usarse la translación de textura de los skyplanes.

- Mezclando skyplanes con skyboxes introducimos un nuevos problemas:
    * Distorcion creada donde el skyplane intersecta la capa mas alejada del skybox, o
    * La aristas del skyplane son visibles cuando se termina el plano dentro del skybox.

Rendereado del cielo

OpenGL

Skyboxes – Animación

- Para evitar la intersección se puede usar el depth buffer antes de dibujar la capa mas cercana del skybox.

Este procedimiento involucra los siguientes pasos:
1.-Dibujar las capas mas alejadas del skybox.
2.-Deshabilitar la escritura de la profundidad (glDepthMask(GL_FALSE)).
3.-Dibuja el skyplane.
4.- Habilitar la escritura de profundidad (glDepthMask(GL_True)).
5.-Dibujar las capas mas cercanas del skybox

Rendereado del cielo

OpenGL

Skydomes

- Es importante decidir cual método de sky-rendering usar.

- Debemos saber las ventajas y desventajas de cada método.

- Ejemplo:
Para un escena del interior de una complejo no se necesita un cielo realista, pero para el rendereado de ambientes al aire libre se necesita un buen algoritmo de rendereado.

Rendereado del cielo

OpenGL

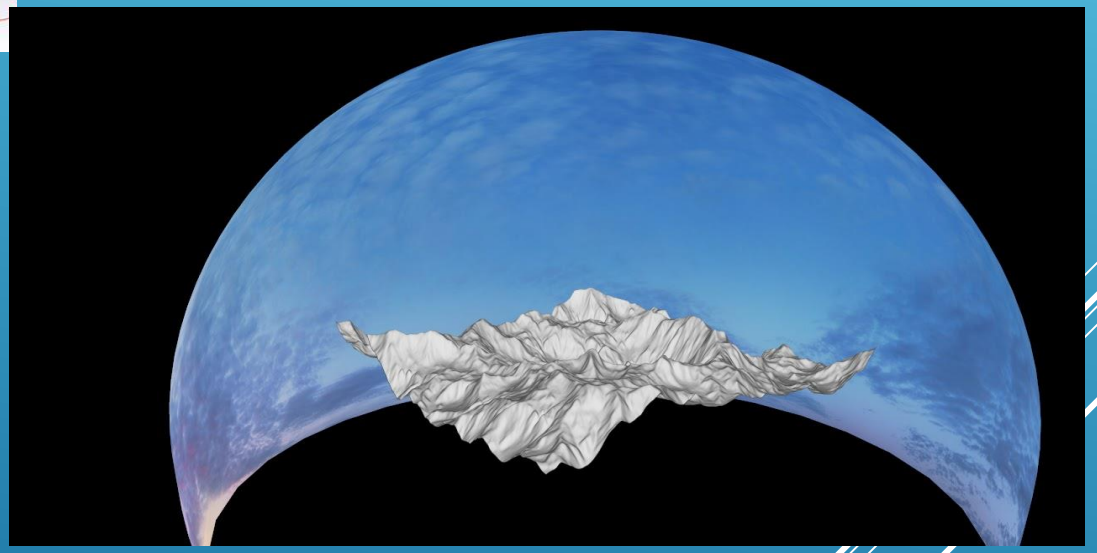
Skydomes

- Para estos casos tenemos el método de skydome.

Las ventajas del skydome son:
- Produce colores realistas, que permiten mostrar la transición entre día y noche.
- Trabaja bien con algoritmos de coloreado avanzados.
- Si es texturizado, se requiere de una solo textura.
- Permite calcular fácilmente el color de la niebla.

Rendereado del cielo

OpenGL



Rendereado del cielo

OpenGL

Skydomes

- Las ventajas del skydome lo hace un método fácil de usar cuando necesitas un cielo dinámico o cuando se quiere usar una textura de alta definición.

- Cuando se utiliza un método dinámico de cielo se calcula a menudo el ángulo entres un vértice. especifico del domo y el sol.

Rendereado del cielo

OpenGL

Skydomes

Algunas desventajas de los skydomes son:

- Requieren mas polígonos que los skyboxes.

- Las texturas para los skydomes pueden ser difíciles de crear u obtener.

-Solo muestra el cielo arriba del horizonte y no debajo de este.

Rendereado del cielo

OpenGL

Skydomes

- Si el domo se divide en N divisiones y M lados entonces tenemos 2N(M+1) y 2NM triángulos

- Para un domo de alta calidad los valores para N y M son 32 y 48 respectivamente .

- El numero de lados y divisiones puede modificarse para una desempeño óptimo

Rendereado del cielo

OpenGL

Skydomes

- Las texturas para los skydome son difíciles de crear.

- Si se desea crear una textura del domo, hay que ver como tratar la tercera desventaja de los skydomes.

Rendereado del cielo

OpenGL

Skydomes

Hay 2 maneras de tratar este problema:

1.- Cuando trabajamos con un domo con los colores del vértice, se puede obtener el color del horizonte, el cual limpia la pantalla y se convierte al color de la niebla dando la impresión de transición entre el terreno y el cielo.

2.- La segunda opción es añadir un disco como base del domo cerrándolo y colorear las aristas para que coincidan con el horizonte.

Rendereado del cielo

OpenGL

Skydomes – Generación del domo

- Un skydome se describe como una media esfera.

- Para renderear una media esfera necesitamos calcular los puntos usados para crear la media esfera.

- $x^2 + y^2 + z^2 = r^2$

- $f(p) = x^2 + y^2 + z^2 - r^2$

Estas ecuaciones definen una esfera en el centro del sistema con radio r.

Rendereado del cielo

OpenGL

Skydomes – Generación del domo

- En lugar de lo anterior su utilizan coordenadas polares para describir la esfera.

- En coordenadas polares cada punto es calculado especificando 2 ángulos , θ y Φ, los cuales representan las longitudes y latitudes de una esfera.

- Φ es el ángulo entre el vértice y el eje y.

- θ es el ángulos entre la proyección de un vértice en al plano xz y el eje x.

Rendereado del cielo

OpenGL

Skydomes – Calculo de los vértices

El calculo de las coordenadas cartesianas x, y y z a coordenadas polares se logra con:

$x = r\cos(\theta)*\cos(\Phi)$ con θ de 0 a 2π y Φ de 0 a $\pi/2$

$y = r\ \mathrm{sen}(\Phi)$

$z = r\ \mathrm{sen}(\theta) * \cos(\Phi)$

Rendereado del cielo

OpenGL

Skydomes – Calculo de los vértices

- Se debe especificar la precisión o intervalos en los cuales se calculan los puntos.

- Se corta la esfera horizontalmente en partes o divisiones, cada una de las cuales representa un anillo de vértices

- El numero de vértices en cada anillo es especificado por el numero de lados en que el anillo se divide.

Rendereado del cielo

OpenGL

Skydomes – Calculo de los vértices

- El numero de vértices de una parte es lado +1 y no es igual al numero de lados.

- El primer y ultimo punto son el mismo pero se especifican por separado en el rendering.

- El numero de anillos para una media esfera es partes +1, el tope del domo también se renderea como un anillo aunque todos sus vértices son el mismo.

Rendereado del cielo

OpenGL

Skydomes – Calculo de los vertices

- EL numero de vértices que se necesita para el domo es:

NumVertices = (slices+1) * (slices+1)

-Ya que sabemos el numero de partes y lados, podemos calcular los intervalos $\Delta\theta$, $\Delta\Phi$ en los ángulos θ y Φ en los cuales se calculan los puntos:

$\Delta\theta = 2\pi/(slices)$

$\Delta\Phi = (\pi/2)/slices$

Rendereado del cielo

OpenGL

Skydomes – Calculo de los vértices

```
VertexBuffer = new Vector3[(slices+1)*(slices+1)];
//Calculo de Δθ
float polyAng = 2.0f * PI / slices;
for(j=0; j<= slices;j++){
        \\Calculo de ΔΦ
        ang = j * ((PI/2) / slices);
        for(i=0; i<=sides;i++){
                //Calculo de las coordenadas
                vx = cos(i*polyang)* cos(ang);
                vy = sin(i*polyang) * dampeningFactor;
                vy = sin(i*polyang)* cos(ang);
                VertexBuffer[j*(sides+1)+i].x = vx.SkyRadius;
                VertexBuffer[j*(sides+1)+i].y = vy.SkyRadius;
                VertexBuffer[j*(sides+1)+i].z = vz.SkyRadius;
        }
}
```

Rendereado del cielo

OpenGL

Skydomes – Calculo de los vértices

- $\Delta\theta$ y $\Delta\Phi$ se multiplican por i y j para obtener el ángulo final y las coordenadas cartesianas.

-VertexBuffer almacena las coordenadas usando el valor NumVertices para indicar el tamaño.

Skydomes – Calculo de los vértices

- La amortiguación del domo en el eje y produce un domo elíptico que puede ser útil en ciertas situaciones.

- Esta amortiguación se especifica por un factor, dampening factor, el cual se multiplica por la coordenada y.

- El valor del factor esta entre 0 y 1. Un valor de 1 deja todo sin cambios, un valor de 0.5 produce un domo con la mitad de altura.

Skydomes – Creación y rendering de los triángulos

- Los índices son calculados después de las coordenadas.

- La forma de renderear un domo es usando triangle strips.

- Se crean tantos triangles strips como divisiones y cada strip formaran triángulos entre los 2 anillos de vértices

Rendereado del cielo

OpenGL

Skydomes – Calculo de los vértices

```
IndexBuffer = new unsignedshort[slices*(slice+1*)*2];
for(j=1;j<=slices;j++){
        for(i=0;i<=sides;i++){
                IndexBuffer[ind++] = j*(sides+1)+i;
                IndexBuffer[ind++] = (j-1)*(sides+1)+i;
        }
}
```

Rendereado del cielo

OpenGL

Skydomes – Calculo de los vértices

- El rendereado es directo una vez que el arreglo de vértices e índices se pasan en OpenGL

```
glEnableClientState(GL_VERTEX_ARRAY);
glVertexPointer(3,GL_FLOAT,0,&vertexBuffer[0]);
for(i = 0; i< NumSlices;i++)
        glDrawElements( GL_TRAINGLE_STRIP,
                        (NumeSlices+1)*2,
                        GL_UNSIGNED_SHORT,
                         IndexBuffer[i*(NumSides+1)*2]);
glDisableClientState(GL_VERTEX_ARRAY);
```

Rendereado del cielo

OpenGL

Skydomes – Mapeo de una imagen del cielo en el domo

- La forma mas fácil para realizar esto es mapear la textura plana desde el plano xz.

- Este método produce discrepancias y estiramiento en la base del domo.

- En lugar de este método usamos el mapeo de textura esférico

Rendereado del cielo

OpenGL

Skydomes – Mapeo de una imagen del cielo en el domo

- En el mapeo de textura esférico, una textura rectangular se proyecta en la esfera, cubriendo la superficie.

- Las coordenadas se pueden calcular desde los vértices usando las ecuaciones de mapeo esférico

Rendereado del cielo

OpenGL

Skydomes – Mapeo de una imagen del cielo en el domo

$$\theta = \arcsin \frac{z}{\sqrt{x^2 + y^2}} \qquad \varphi = \frac{y}{\sqrt{x^2 + y^2 + z^2}}$$

$$U = 1/2 + \theta/2\pi \qquad V = 1/2 + \varphi/(\pi/2)$$

- Estas ecuaciones no se necesitan evaluar pues ya se ha creado la esfera.

- Ya conocemos los ángulos y podemos calcular las coordenadas de los vértices

Rendereado del cielo

OpenGL

Skydomes – Mapeo de una imagen del cielo en el domo

- La primero es asignar el espacio para las coordenadas de la textura:

TexCoordBuffer = new Vector2[(slices+1)*(slices+1)];

- El numero de las coordenadas es igual al numero de las vértices ya que cada vértices tiene una coordenada de textura.

- Cada vértice de la esfera es mapeado en la imagen de acuerdo a cual lado y división pertenece

Rendereado del cielo

OpenGL

Skydomes – Mapeo de una imagen del cielo en el domo

```
TexCoordBuffer[j*(sides+1)+i].u = (float)(i)/(float)(sides);

TexCoordBuffer[j*(sides+1)+i].v = (float)(j)/(float)(slices);
```

- Estas lineas se agregan al ciclo interno en el cual i y j son divididos por los lados y divisiones para producir loas coordenadas UV en el rango [0,1].

Rendereado del cielo

OpenGL

Skydomes – Rendereado del domo

- Antes de renderearse el domo con un mapeo de textura, TexCoordBuffer[] tiene que darse como un arreglo usando glDrawElements().

-Como con skyplanes y skyboxes, se debe deshabilitar cualquier operación del pipeline innecesaria

Skydomes – Rendereado del domo

```
glEnableClientState(GL_VERTEX_ARRAY);
glEnableClientState(GL_TEXTURE_COORD_ARRAY);
glVertexPointer(3, GL_FLOAT,0, &VertexBuffer[0]);
glTextCoordPointer(2,GL_FLOAT,0,&TexCoordBuffer[0]);
for(i=0;i<NumSlices;i++){
   glDrawElements( GL_TRAINGLE_STRIP,
                           (NumSlices+1)*2,
                           GL_UNSIGNED_SHORT,
                           IndexBuffer[i*(NumSides+1)*2]);
}
glDisableClientState(GL_VERTEX_ARRAY);
glDisableClientState(GL_TEXTURE_COORD_ARRAY);
```

Rendereado del cielo

Skydomes – Texturas para skydomes a base de cubemaps

- Una textura de skydome representa una proyección equidistante en 360 grados del cielo.

- Mapea una superficie esférica que rodea la cámara en un plano.

- Este tipo de proyección muestra gran distorsión en los polos y muy poca en el ecuador.

Rendereado del cielo

OpenGL

Skydomes – Texturas para skydomes a base de cubemaps

- Esta es la que necesita el skydome, ya que en su cima los triángulos que componen las estructuras son muy pequeños y están muy juntos comparados con los del horizonte.

- Usando la proyección equidistante se permite un mapeo de textura muy simple.

- Se asignan las coordenadas de texturas basado en la latitud y longitud en el domo.

Rendereado del cielo

OpenGL

Skydomes – Texturas para skydomes a base de cubemaps

El proceso para crear la textura es:

1.-Se crean 6 texturas que se mapean en un skybox.

2.- Usando una cámara fisheye de 180 grados que mira hacia arriba, rendereamos la imagen.

3.-Desenvolvemos el resultado en una imagen equidistante.

Rendereado del cielo

OpenGL

Rendereado del cielo

OpenGL

Skydomes – Texturas para skydomes a base de cubemaps

Para generar una imagen fisheye pueden construir una escena en el programa POV-Ray, un raytracer libre, que consiste de:

1.- Un skybox mapeado con cubemap. El sky se extiende de (-x,-x,-x) a (x,x,x).

2.- Una cámara fisheye de 180 grados que apunta hacia arriba, localizada en medio del skybox.

Rendereado del cielo

OpenGL

Skydomes

Usando tablas de color para la simulación de skydome.

- Un método para aplicar colores a los vértices del domo es usar una tabla de búsqueda

- Si se tiene una buena tabla de búsqueda, esta es una forma rápida de conseguir una buena apariencia de cielo.

- La tabla se puede leer de un archivo de imagen, donde cada píxel representa el color en un punto especifico de tiempo y latitud en el skydome.

Rendereado del cielo

OpenGL

Skydomes

Usando tablas de color para la simulación de skydome.

- La coordenada x en la imagen representa el tiempo, que va de 0 a 24.

- La coordenada y en la imagen representa la latitud.

- Ya que el numero de lados y divisiones pueden no ser iguales al numero de píxeles en x y y respectivamente, se usa interpolación

Rendereado del cielo

OpenGL

Skydomes

Usando tablas de color para la simulación de skydome.

-Cada vértice se toma de manera individual y, en le primer paso, los 4 valores cercanos en la tabla son determinados.

```
float fX = TimeOfDay / 24.0f * width;
int iX = fX;
int iXn = (iX >=width -1 ? iX : iX+1);
```

iX es el índice del color anterior mas cercano.
iXn es el índice del siguiente color.
fX nos da el valor en el eje x en el que esta el vértice

Rendereado del cielo

OpenGL

Skydomes

Usando tablas de color para la simulación de skydome.

Para el eje y se realiza el mismo proceso(los índices no deben estar fuera de la tabla):

```
float fY = theta / (PI /2) * Height;
int iY = fY;
If( iY >= Height ) iY = Height -1;
int iYn = (iY >= Height -1 ? iY : iY+1);
```

El ángulo contiene la latitud en el domo que va de 0 en la parte inferior a PI/2 en la parte superior.

Rendereado del cielo

OpenGL

Skydomes

Usando tablas de color para la simulación de skydome.

- Conociendo las dimensiones de la tabla podemos obtener los colores usando estos índices

- Estos constituyen un cuadrado y las variables son nombradas con (0,0) a la esquina inferior izquierda y (1,1) la esquina superior derecha.

Rendereado del cielo

OpenGL

Skydomes

Usando tablas de color para la simulación de skydome.

- En ColorTable[] están los datos de la imagen carga desde un archivo.

```
Color4 col00 = ColorTable[iY * width + iX];
Color4 col01 = ColorTable[iY * width + iXn];
Color4 col11 = ColorTable[iYn * width + iXn];
Color4 col10 = ColorTable[iYn * width + iX];
```

Rendereado del cielo

OpenGL

Skydomes

Usando tablas de color para la simulación de skydome.

- Finalmente se puede hacer una interpolación bilineal, col00 con col01 y col10 con col11

```
float hf = fX – iX;
Color4 Horiz0 = col00 * (1.0f – hf) + col01 * hf;
Color4 Horiz1 = col10 * (1.0f – hf) + col11 * hf;
```

Skydomes

Usando tablas de color para la simulación de skydome.

- Los resultados anteriores son interpolados en el eje y, el resultado es almacenado en un apuntador

```
float vf = fY – iY;
*ptr = Horiz0 * (1.0f – vf) + Horiz1 * vf;
```

Rendereado del cielo

OpenGL

Skydomes

Usando tablas de color para la simulación de skydome.

```
Void GetColor( float TimeofDay, float Phi, float Theta,
                                                Color4 *ptr){
    float fx = TimeofDay/24.0f * width;
    int iXn = fx;
    int iXn = (iX >= width -1 ? iX: iX+ 1);
    float fy = Theta / (PI /2) * Height;
    int iY = fY;
    if( iY >= Height ) iY = Height – 1;
    int iYn = (iY >= Height -1 ? iY: iY+ 1);
```

Rendereado del cielo

OpenGL

Skydomes

Usando tablas de color para la simulación de skydome.

```
Color4 col00 = ColorTable[iY * width + iX];
Color4 col01 = ColorTable[iY * width + iXn];
Color4 col11 = ColorTable[iYn * width + iXn];
Color4 col10 = ColorTable[iYn * width + iX];

float hf = fX – iX;
Color4 Horiz0 = col00 * (1.0f – hf) + col01 * hf;
Color4 Horiz1 = col10 * (1.0f – hf) + col11 * hf;

float vf = fY – iY;
*ptr = Horiz0 * (1.0f – vf) + Horiz1 * vf;
}
```

Rendereado del cielo

OpenGL

Skydomes

Usando tablas de color para la simulación de skydome.

- Antes de que el domo sea rendereado, el ColorBuffer[] tiene que ser dado como un arreglo de elementos usando glDrawElements().

- Se tiene que actualizar el color buffer cuando haya cambios de tiempo.

Rendereado del cielo

OpenGL

Skydomes

Usando tablas de color para la simulación de skydome.

```
float halfPi = PI/2;
float SliceAng  = (PI/2)/NumSlices;
float SideAng  = (PI*2)/NumSides;
float Theta,phi;

Color4 *ColorPtr = ColorBuffer;
for(ind=0; ind <=NumSlices; ind++){
        for(num=0; num <= NumSides; num++){
                theta= halfPI-SliceAng *ind;
                phi = num * SideAng;
                GetColor(TimeOfDay, phi, halfPI – theta, ColorPtr);
                ColoPtr++;}}
```

Rendereado del cielo

OpenGL

Skydomes

Usando tablas de color para la simulación de skydome.

```
void CskyDome::RenderDome(){
glEnableClientState(GL_VERTEX_ARRAY);
glEnableClientState(GL_COLOR_ARRAY);
glVertexPointer(3,GL_FLOAT,0,&VertexBuffer[0]);
glColorPointer(4,GL_FLOAT,0,&ColorBuffer[0]);
for(i=0;i < NumSlices;i++){
        glDrawElements(GL_TRIANGLE_STRIP,
                       (NumSides+1)*2,
                       GL_UNSIGNED_SHORT,
                       &IndexBuffer[i*(NumSides+1)*2]);}
glDisableClientState(GL_VERTEX_ARRAY);
glDisableClientState(GL_COLOR_ARRAY);
}
```

Rendereado del cielo

OpenGL