Proyecto #1
Esquema de monoambiente

Fernando Braña

- Objetivos
- Proceso de desarrollo
- Controles
- Detalles técnicos de interés
- Dificultades
- Conclusión

Objetivo

Proceso de desarrollo

Desarrollo iterativo

Proceso de desarrollo

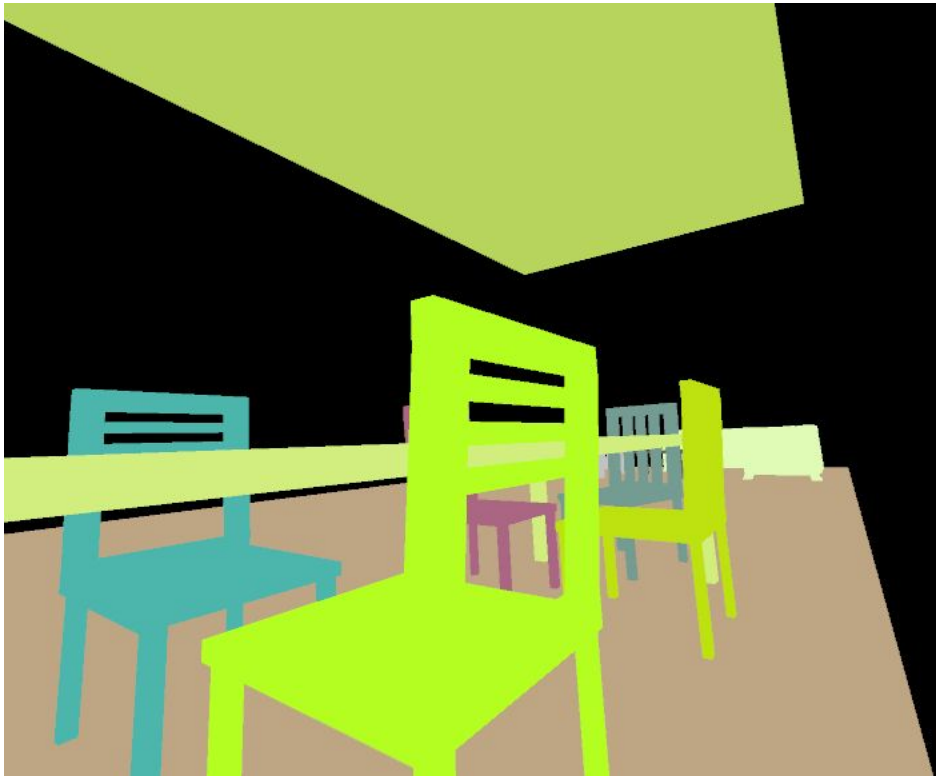
Desarrollo iterativo:

Cargador de archivos .obj

Proceso de desarrollo

Desarrollo iterativo:

Cargador de archivos .obj



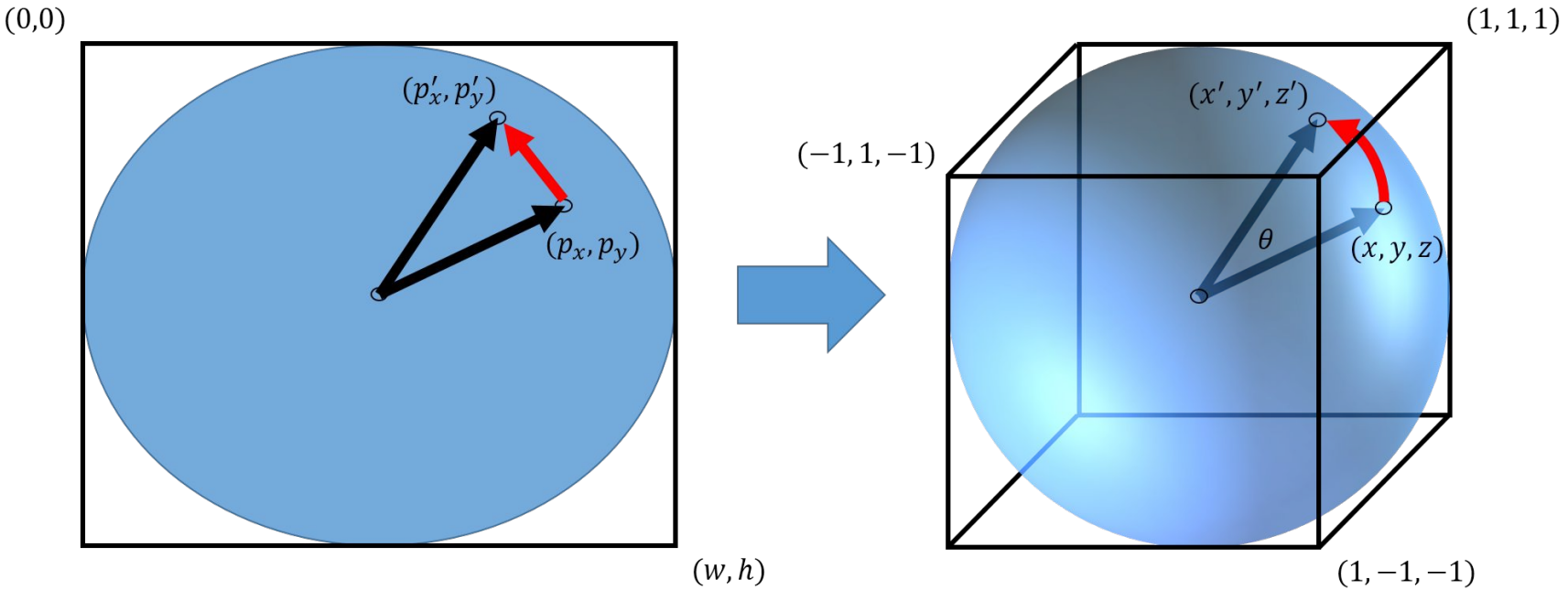
Proceso de desarrollo

Desarrollo iterativo:

Cargador de archivos .obj

Camaras

Proceso de desarrollo



Proceso de desarrollo

Desarrollo iterativo:

Cargador de archivos .obj

Camaras

Creación de cada cuarto

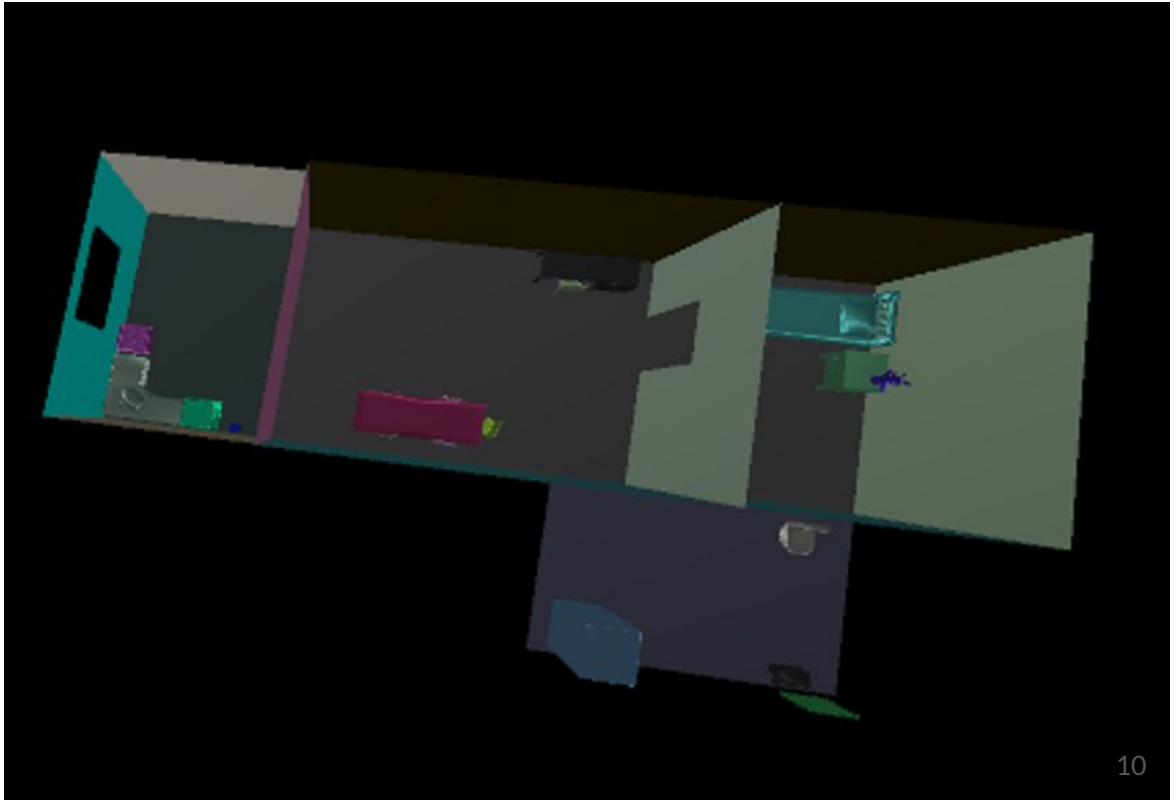
Proceso de desarrollo

Desarrollo iterativo:

Cargador de archivos .obj

Camaras

Creación de cada cuarto



Controles

Cambiar de camara

Colorear a todos los objetos aleatoriamente

Mostrar/ocultar los techos

Mostrar/ocultar las paredes

Resetear la cámara orbital

Controles


unity

Camara orbital

Rotación de un vector sobre un eje arbitrario

Camara orbital

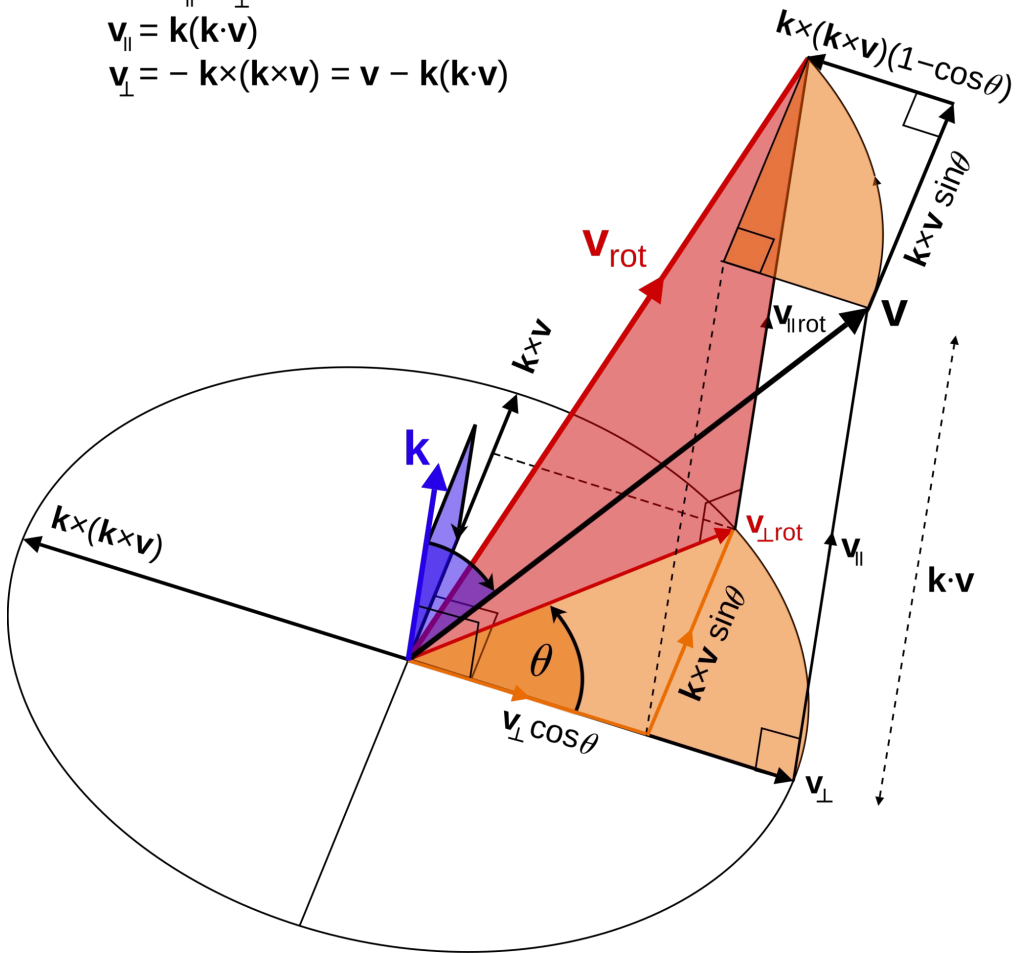
Rotación de un vector sobre un eje arbitrario

```
private Vector3 rotateAround(Vector3 V, float angleInRadians, Vector3 rotationAxis)
{
    Vector3 K = rotationAxis;
    Vector3 rotatedV = V * Mathf.Cos(angleInRadians)
                + (Vector3.Cross(K, V) * Mathf.Sin(angleInRadians))
                + K * (Vector3.Dot(K, V) * (1 - Mathf.Cos(angleInRadians)));
    return rotatedV;
}
```

$\mathbf{v} = \mathbf{v}_{\parallel} + \mathbf{v}_{\perp}$

$\mathbf{v}_{\parallel} = \mathbf{k}(\mathbf{k}\cdot\mathbf{v})$

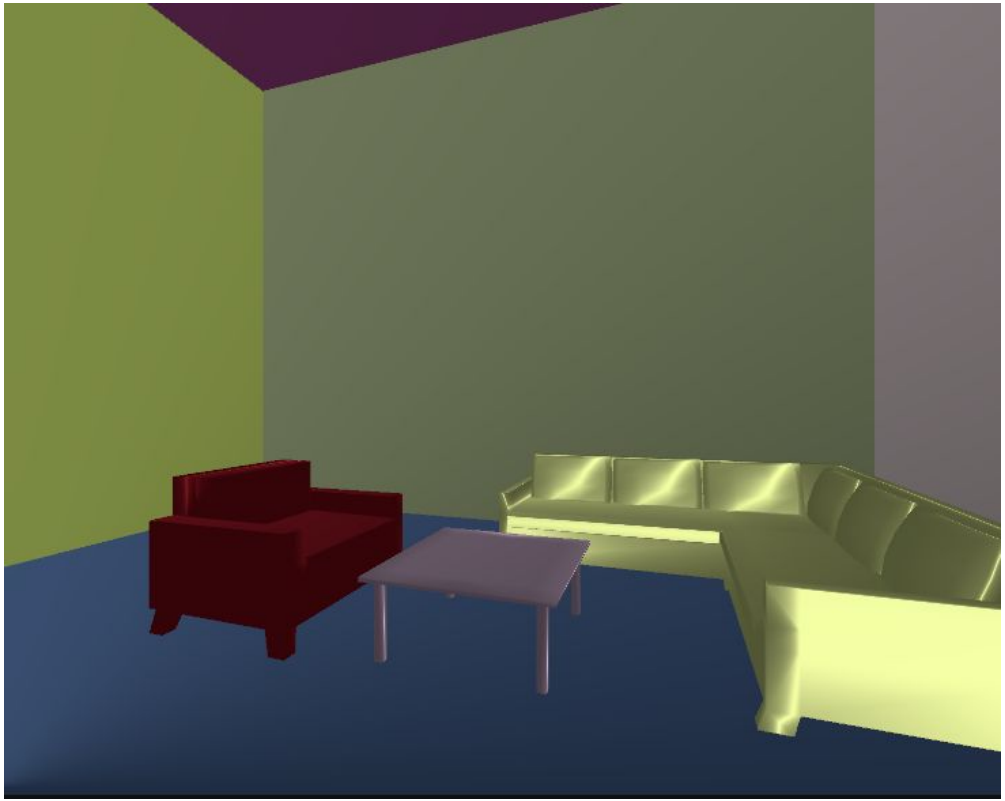
$\mathbf{v}_{\perp} = -\,\mathbf{k}\times(\mathbf{k}\times\mathbf{v}) = \mathbf{v} - \mathbf{k}(\mathbf{k}\cdot\mathbf{v})$



Sistema simple de iluminación

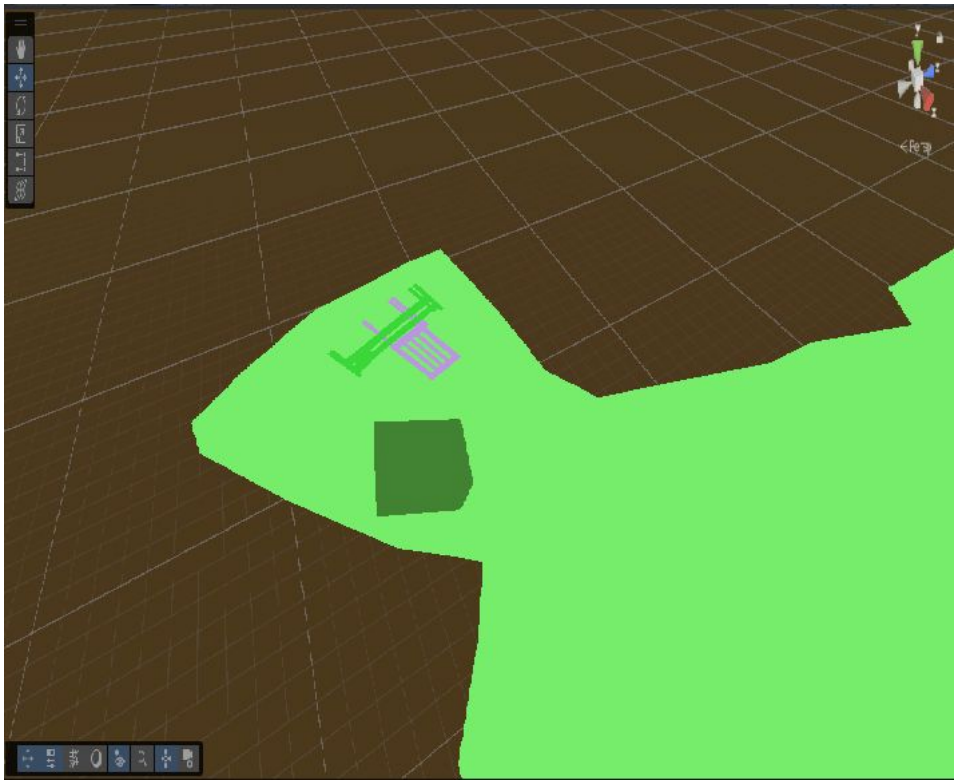
```
fixed4 frag(v2f i): SV_Target {
    float ambientStrength = 0.1;
    float3 ambient = ambientStrength * _LightColor;

    float3 normalizedNormal = -normalize(i.normal);
    float3 lightDir = normalize(_LightPos - i.vertex);
    fixed diffuseFactor = max( dot(normalizedNormal, lightDir), 0.0);
    float3 diffuse = diffuseFactor * _LightColor;
    float3 finalColor = (ambient+diffuse) * _color;
    return half4(finalColor, 1.0);
}
```

Sistema simple de iluminación



Dificultades

Profundidades invertidas



Profundidades invertidas

Solución?

Cambiar a OpenGL


60
55
5
50
10
45
15
40
20
35
30
25

Conclusion