

## PROYECTO V – TEXTURAS

El proyecto consiste en iluminar una escena, sin necesidad de crear ninguna luz usando texturas y shaders.

Al correr la plantilla (**Plantilla\_Proyecto\_Texture\_Shader01.rar**) tendrán un escenario con una textura de color plano.



Tienen 4 texturas:

baked\_flat : color plano uniforme.

baked\_fill01: Iluminación de relleno 01

baked\_fill02: Iluminación de relleno 02

baked\_keyrabbitt: Iluminación central al conejo.

baked\_checker: máscara para crear el patrón del piso.

Usando estas texturas deben dar una sensación de iluminación en la escena. Entiéndase, que **NO** deben aplicar la fórmula para el componente difuso o especular vista en clase, el

componente difuso se encuentra ya contenido en las texturas baked\_fill01, baked\_fill02 y baked\_keyrabbit.

Para crear el efecto deberán usar un shader, ya que no es posible crear el efecto aplicando texturas directamente en OpenGL.

Teclas:

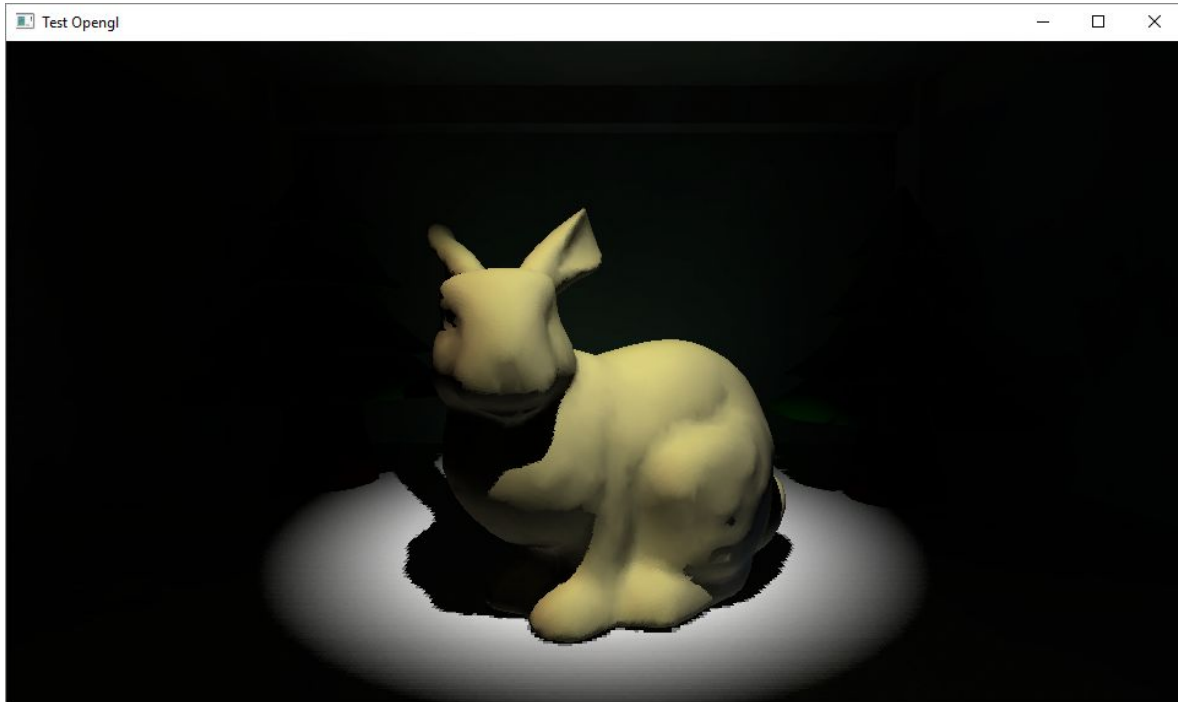
- 1: incrementa la intensidad de la luz ambiental en 0.05 (baked\_flat)
- 2: reduce la intensidad de la luz ambiental en 0.05 (baked\_flat)
- q: incrementa la intensidad de la luz de relleno 01 en 0.05 (baked\_fill01)
- w: reduce la intensidad de la luz de relleno 01 en 0.05 (baked\_fill01)
- a: incrementa la intensidad de la luz de relleno 02 en 0.05 (baked\_fill01)
- s: reduce la intensidad de la luz de relleno 02 en 0.05 (baked\_fill01)
- z: incrementa la intensidad de la luz central del conejo en 0.05 (baked\_keyrabbit)
- x: reduce la intensidad de la luz central del conejo en 0.05 (baked\_keyrabbit)
- e,r,t : incremente el componente RGB en 0.05 del color en la luz de relleno 01
- y,u,i : reduce el componente RGB en 0.05 del color en la luz de relleno 01
- d,f,g : incremente el componente RGB en 0.05 del color en la luz de relleno 02
- h,j,k : reduce el componente RGB en 0.05 del color en la luz de relleno 02
- c,v,b : incremente el componente RGB en 0.05 del color en la luz central del conejo.
- n,m, , : reduce el componente RGB en 0.05 del color en la luz central del conejo.
- o: activa el filtro bilineal
- p: desactiva el filtro bilineal.
- 3, 4, 5, 6: activa 4 colores distintos para el patrón del piso, dejo a ustedes libre la elección de los 4 colores.

Para el **filtro bilineal**, utilizar el algoritmo que se presenta en la siguiente página:

[http://www.ozone3d.net/tutorials/glsl\\_texturing\\_p02.php#part\\_2](http://www.ozone3d.net/tutorials/glsl_texturing_p02.php#part_2)

Tengas cuidado con la intensidades de las luces y los componentes RGB, **NO** pueden tomar valores negativos.

Cuando arranque el programa por primera vez debe verse como se muestra a continuación:



#### TIPS:

1. No deben usar en ningún momento la fórmula del componente difuso y especular vista en clase para crear el efecto, todo se logra usando las texturas.
2. Para el piso, no consideren `baked_checker` como una textura visualmente hablando, piensen en esta textura como una máscara. Como pista solo puedo decirles función GLSL : “mix” .
3. **Importante:** si cuando compilan y se ejecuta el programa le lanza un error de excepción en la función `SetTexture`, es porque seguro tienen un error en el fragment o vertex shader, verifiquen bien las variables y algún error en el código.
4. Implementen poco a poco, primero hagan una luz programen y luego vayan a la siguiente, sino podrá ser frustrante por el punto 3.
5. Vean el video que está con este documento en el aula virtual para que vean cómo debe comportarse la aplicación.

#### Especificaciones del proyecto:

1. El proyecto es en pareja..
2. El proyecto tiene un 12% de nota.
3. La fecha de entrega es hasta el Domingo 12/06, 11:59pm.
4. Deben enviarme por correo (comprimido en NombreEstudianteYNombreEstudiante.rar) el código fuente (solo los archivos .cpp, no es necesario comprimir todo el proyecto), el ejecutable.
5. **IMPORTANTE:** el asunto del correo debe estar escrito con el siguiente formato:  
"Graficas I 2016 - Proyecto V - Opengl - NombreEstudianteYNombreEstudiante".