

Atributs :

layout (location = 0) in vec3 vertex;
layout (location = 1) in vec3 normal;
layout (location = 2) in vec3 color;
layout (location = 3) in vec2 texCoord;

Uniforms :

uniform mat4 modelMatrix;
uniform mat4 viewMatrix;
uniform mat4 projectionMatrix;
uniform mat4 modelViewMatrix;
uniform mat4 modelViewProjectionMatrix;
uniform mat4 modelMatrixInverse;
uniform mat4 viewMatrixInverse;
uniform mat4 projectionMatrixInverse;
uniform mat4 modelViewMatrixInverse;
uniform mat4 modelViewProjectionMatrixInverse;
uniform mat3 normalMatrix;
uniform vec4 lightAmbient;
uniform vec4 lightDiffuse;
uniform vec4 lightSpecular;
uniform vec4 lightPosition;
uniform vec4 matAmbient;
uniform vec4 matDiffuse;
uniform vec4 matSpecular;
uniform float matShininess;
uniform vec3 boundingBoxMin;
uniform vec3 boundingBoxMax;
uniform vec2 mousePosition;

Funcions importants:

float pow(float x, float y);
float sqrt(float x);
float abs(float x); → es pot utilitzar directament en vectors
float floor(float x) → retorna la part entera per sota, ex 0.5 = 0 i -0.5 = -1
float fract(float x) → retorna la part fraccional, ex 2.5 = 0.5
float mod(float x, float y) → fa el modul de x mod y
float clamp(float x, float minVal, float maxVal) → si x esta entre min i max retornem la x, si x < min retorna min i si x > max retorna max
vec4 mix(vec4 x, vec4 y, float a) → fa la interpolacio lineal entre x i y en funcio de a
float step(float edge, float x) → retorna 0 si la x és més petita que la edge, retorna 1 en cas contrari

float smoothstep(float edge0, float edge1, float x) → retorna 0 si x és més petit que edge0, retorna 1 si és més gran que edge1 i en un altre cas retorna el valor interpolat entre 0 i 1

float distance(vec3 p0, vec3 p1) → distancia entre dos vectors(dona igual la mida d'aquests), el càlcul és com fer la resta entre els dos i acte seguit el length

float length(vec3 x) → retorna l'arrel quadrada de la suma dels quadrats dels seus components, es a dir, la llargada d'aquest vector

float dot(vec3 x, vec3 y) → retorna el producte escalar

vec3 cross(vec3 x, vec3 y) → retorn el producte vectorial

vec3 normalize(vec3 x) → retorna un vector de length 1

vec4 texture2D(sampler2D sampler, vec2 coord) → on coord son les coordenades de textura

Matrïus de rotació:

A l'eix x:

mat3 rotate = mat3(vec3(1,0,0),vec3(0,cos(angle),sin(angle)),vec3(0,-sin(angle),cos(angle)));

A l'eix y:

mat3 rotate = mat3(vec3(cos(angle),0,-sin(angle)),vec3(0,1,0),vec3(sin(angle),0,cos(angle)));

A l'eix z:

mat3 rotate = mat3(vec3(cos(angle),sin(angle),0),vec3(-sin(angle),cos(angle),0),vec3(0,0,1));

Funció sinusoidal:

$A \cdot \sin(2 \cdot M_PI \cdot freq \cdot time + fase)$

La fase sol ser 0

En el cas de que et demanin un desplaçament segons la sinusoidal el que farem sera sumar el resultat obtingut al vertex que li passem a gl_Position

Si et demanen que vagi en direcció de la normal multiplicarem així:

$normal \cdot A \cdot \sin(2 \cdot M_PI \cdot freq \cdot time + fase)$

Espai de coordenades a cada moment:

Iniciem al VS amb object space.

Multipliquem per la model matrix per tenir en world space

Multipliquem per la view per tenir en eye/camara space

Multipliquem per la projection per tenir coordenades de clipping o de finestra

Passem al FS i es fa la perspective division (dividir entre la component w) i tenim coordenades normalitzades

