

Definice jazyka

- OpenGL Shading language je založen na systému programovacího jazyka ANSI C.

Doplňky k C

- vektorové typy jsou podporovány pro hodnoty float, int a bool

- `vec{2,3,4}`, `ivec{2,3,4}`, `bvec{2,3,4}`,

```
vec4 v=vec4(1.0, 0.5, 0.0, 1.0);  
vec4 u=vec4(5.0, 3.0, 2.0, 4.0);  
vec4 w;  
vec2 w2;  
float a,b;
```

```
w= v+u;  
a= v.x; b= v.z;  
w2= u.zw; (swizzling)
```

```
// x,y,z,w - Treats a vector as a position or direction  
// r,g,b,a - Treats a vector as a color  
// s,t,p,q - Treats a texture coordinate
```

Definice jazyka

- Matice

mat2, mat3, mat4 -matice 2x2, 3x3, 4x4

K matici můžete přistupovat jako k vektoru po sloupcích.

```
vec4 u;  
mat4 m;
```

```
m= ...  
u= m[3];
```

- Samplers

sampler{1,2,3}D

samplerCube

textura mapovaná na kostku.

sampler{1,2}Dshadow

hloubková textura s porovnáním.

Vestavěné funkce – Funkce pro přístup k texturám

- Vestavěné funkce se suffixem “Lod” jsou povoleny pouze ve vertex shaderu. Pro funkce “Lod”, *lod* je přímo použito jako level of detail.
- Vestavěné funkce se suffixem “Proj” mohou být použity pro projektivní texturování. Umožňují textuře projektovat se na objekt obdobně jako jsou promítány slajdy. Může být použito k výpočtu stínových map.

```
vec4 texture1D(sampler1D sampler, float coord [,float bias])  
vec4 texture1DProj(sampler1D sampler, vec2 coord [,float bias])  
vec4 texture1DProj(sampler1D sampler, vec4 coord [,float bias])  
vec4 texture1DLod(sampler1D sampler, float coord [,float bias])  
vec4 texture1DProjLod(sampler1D sampler, vec2 coord [,float bias])  
vec4 texture1DProjLod(sampler1D sampler, vec4 coord [,float bias])
```

- Texturové souřadnice *coord* se používají k vyzvednutí informací z 1D textury aktuálně specifikované *samplerem*. Pro projektivní (Proj) verze, texturová souřadnice *coord.s* je dělena poslední komponentou *coord*. Druhá a třetí komponenta *coord* je pro variantu *vec4 coord* ignorována.

Vestavěné funkce

- Obdobně **texture2D**
- Další položky: **texture3D**, **textureCube**, **shadow1D**, **shadow2D**.

Funkce zpracovávající fragmenty

- Pouze na fragment processoru
- Funkce derivací, **dFdx** a **dFdy** jsou použity k určení rychlosti změn výrazu.

```
float dFdx(float rho), ...
```

Vrátí derivaci v *x* pro vstupní argument *rho*.

```
float dFdy(float rho), ...
```

Vrátí derivaci v *y* pro vstupní argument *rho*.

Šumové funkce ...

Důležité vestavěné proměnné

Vertex Attributes

```
attribute vec4 gl_Color;  
attribute vec4 gl_SecondaryColor;  
attribute vec4 gl_Normal;  
attribute vec4 gl_MultiTexCoord0;  
attribute vec4 gl_MultiTexCoord1;  
attribute vec4 gl_MultiTexCoord2;  
//... up to gl_MultiTexCoordN-1, where N = gl_MaxTextureCoords  
attribute float gl_FogCoord;
```

Uniform Variables

Shadery mohou přistupovat k aktuálnímu stavu OpenGL přes vestavěné proměnné obsahující rezervovaný prefix “gl_”

```
gl_ModelViewMatrix  
...
```

Důležité vestavěné proměnné

Speciální výstupní proměnné

```
vec4 gl_position; // must be written to
```

Proměnná `gl_position` je určena pro zápis pozice vrcholu v clipping coordinates, poté co byly vypočteny ve vertex shaderu.

```
float gl_PointSize; // may be written to
```

```
vec4 gl_ClipVertex; // may be written to
```

Varying Variables

```
varying vec4 gl_FrontColor;
```

```
varying vec4 gl_BackColor;
```

```
varying vec4 gl_FrontSecondaryColor;
```

```
varying vec4 gl_BackSecondaryColor;
```

```
varying vec4 gl_TexCoord[gl_MaxTextureCoords];
```

```
varying float gl_FogFragCoord;
```

Použití textur

1. Vyber konkrétní texturovací jednotku a udělej ji aktivní pomocí funkce **glActiveTexture**
2. Vytvoř texturový objekt a navaž ho na aktivní texturovací jednotku pomocí funkce **glBindTexture**
3. Nastav parametry texturového objektu pomocí funkce (příkazů) **glTexParameter**
4. Definuj texturu funkcí **glTexImage**
5. Případně opakuj pro další texturovací jednotky

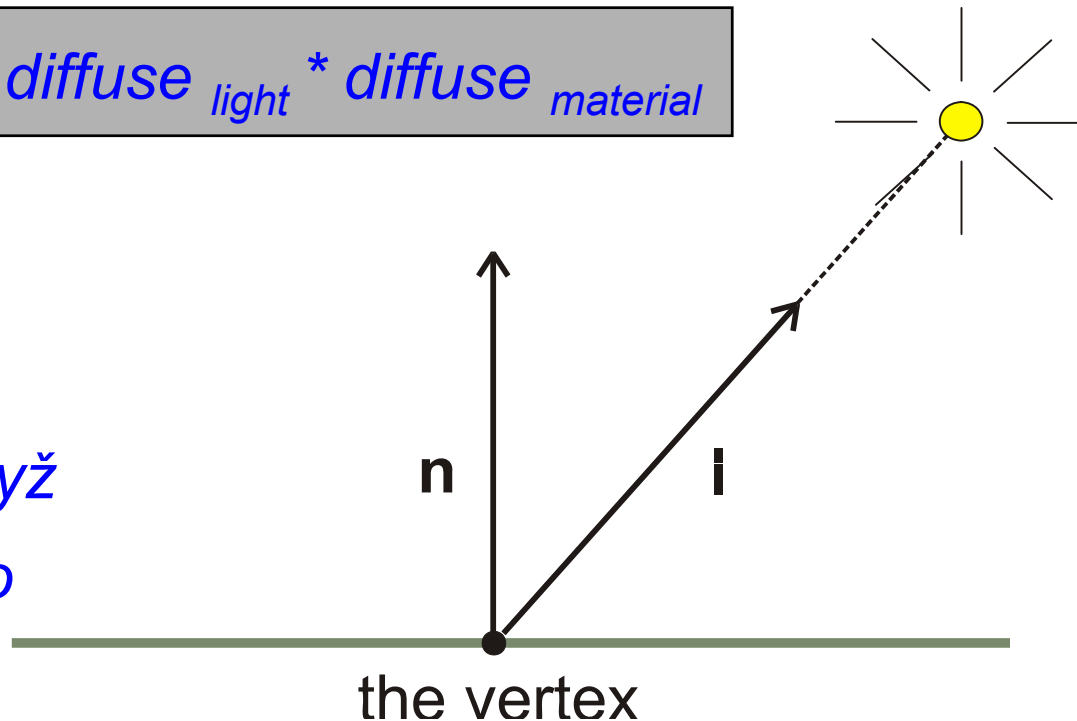
Osvětlení

Jak je počítána barva?

$$\text{ambient} = \text{ambient}_{\text{light}} * \text{ambient}_{\text{material}}$$

$$\text{diffuse} = (\max\{\mathbf{l} \cdot \mathbf{n}, 0\}) \text{diffuse}_{\text{light}} * \text{diffuse}_{\text{material}}$$

- difuze nezáleží na pozici pozorovatele
- difuze je největší když světlo dopadá kolmo na plochu



Osvětlení

Jak je počítána barva?

$$\text{specular} = (\max\{\mathbf{s} \cdot \mathbf{n}, 0\})^{\text{shininess}} \text{specular}_{\text{light}} * \text{specular}_{\text{material}}$$

- *spekulární odraz závisí na
pozici pozorovatele*

