

LMS Moodle VŠB-TUO: 460-2021/03 Základy počítačové grafiky (2024/2025 ZS): Sekce: Cvičení 4

1. Přidejte prosím třídu Camera (bude reprezentovat pohled reálnou kamerou ve scéně).

Ta bude mít pohledovou a projekční matici.

2. Vertex shader rozšířte o pohledovou a projekční matici (na začátku je můžete nastavit na jednotkovou matici).

```
uniform mat4 modelMatrix;  
uniform mat4 viewMatrix;  
uniform mat4 projectionMatrix;  
  
X' = P * V * M * X;
```

3. Pro pohledovou matici můžete použít funkci `glm::LookAt(eye, look, up)`; viz přednáška

```
glm::mat4 Camera::getCamera(void){  
    return glm::lookAt(eye, eye+target, up);  
}
```

4. Pro výpočet vektoru pohledu kamery můžete použít (viz přednáška). Nezapomeňte vhodně omezit úhly obou proměnných a spojit s pohybem myši, abychom se mohli rozhlížet kolem sebe.

```
target.x=sin(alpha)cos(fi);  
target.z=sin(alpha)sin(fi);  
target.y=cos(alpha);
```

5. Nastavte projekční matici na perspektivní promítání (vyzkoušet můžete i rovnoběžné promítání), poměr okna by jste měli upravovat podle rozlišení okna, ve kterém se obsah zobrazuje. `Z_near` nastavte na hodnotu 0.1, nepoužívejte hodnotu 0.

```
//Projection matrix :  
//60° Field of View, 4:3 ratio, display range : 0.1 unit <-> 100 units  
mat4 projection=glm::perspective(60.0f, 4.0f / 3.0f, 0.1f, 100.0f);
```

6. Zaměřte se opět na OOP (zakomponujte vhodně věci jako: Singleton, Observer, Factory, Composite atd.).

7. Křížový odkaz, v případě kamery se nám bude hodit, aby kamera pracovala s shaderem a shader mohl pracovat s kamerou. Dopředná deklarace (Forward declaration) odstraňuje problémy z důvodu křížového odkazu.

```
//----- Shader.h -----
class Camera; // dopredna deklarace (z duvodu krizoveho odkazu)
class Shader{
private:
    Camera* m_camera;
public:
    Shader(Camera* camera);
    void metoda();
};

//----- Shader.cpp -----
#include "Shader.h"
#include "Camera.h"
Shader::Shader(){
    ...
}
```

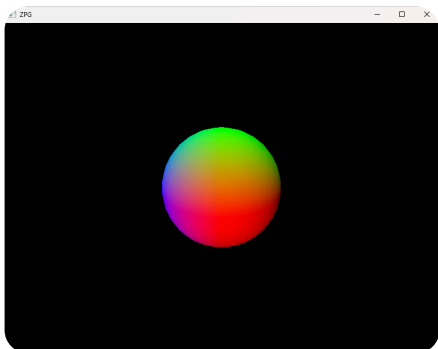
```
//----- Camera.h -----
class Shader; // dopredna deklarace (z duvodu krizoveho odkazu)
class Camera{
private:
    Shader* m_shader;
public:
    Camera(Shader* shader);
    void metoda();
};
```

```
//----- Camera.cpp -----
#include "Camera.h"
#include "Shader.h"
Camera::Camera(){
    ...
}
```

8. Pro propojení třídy Camera a Shader můžete využít návrhový vzor Observer (proč?).

9. Použijte některý ze složitějších modelů ([Models](#)) (koule, krychle, suzi ...), normálu použijte jako barvu.

Pokud byste narazili na problém, tak se můžete podívat na jednoduchou ukázkou.



[main.cpp](#)

[sphere.h](#)

Čeština (cs) ▾

© 2012 - 2025 [VŠB-TUO](#)

[Kontaktovat technickou podporu](#)

Běží na technologii [Moodle Pty Ltd](#)