

Instrukcja laboratoryjna 4	Programowanie grafiki komputerowej
	Temat: Rozszerzenia OpenGL, multiteksturowanie
	Przygotował: dr inż. Tomasz Michno, dr. Inż. Grzegorz Łukawski

1 Wstęp teoretyczny

1.1 Rozszerzenia OpenGL

Rozszerzenia OpenGL pozwalają na korzystanie z nowych funkcji, które nie są częścią głównego standardu OpenGL lub jeszcze do niego nie weszły. Możemy wyróżnić następujące grupy rozszerzeń:

- ARB – oficjalnie zaakceptowane przez organizację ARB,
- EXT – zaakceptowane wspólnie przez producentów sprzętu,
- HP – Hewlett-Packard
- IBM – International Business Machines
- INTEL – Intel
- NV – NVIDIA Corporation
- ATI – ATI Technologies
- MESA – otwarta implementacja OpenGL (Mesa)
- SGIS – Silicon Graphics
- SUN – Sun Microsystems
- WIN – Microsoft

Przykładowe nazwy rozszerzeń:

GL_ARB_multitexture, GL_ARB_texture_cube_map, GL_NV_blend_square

W celu łatwego korzystania z rozszerzeń można skorzystać z jednej z bibliotek: GLEW lub GLEE.

Korzystanie z biblioteki GLEW:

- należy dołączyć plik nagłówkowy
`#include <gl/glew.h>`
- następnie inicjalizujemy bibliotekę oraz sprawdzamy czy funkcja ta zwróciła GL_OK:
`glewInit()`
- sprawdzamy, czy OpenGL jest w minimalnej, obsługiwanej wersji przez wybrane przez nas rozszerzenie:
`if(GLEW_VERSION_1_3){ ... }`
- sprawdzamy, czy rozszerzenie z którego chcemy skorzystać jest dostępne:
`if(rozszerzenie) { ... }`

Korzystanie z biblioteki GLEE:

- należy dołączyć plik nagłówkowy
`#include <gl/glew.h>`
- sprawdzamy, czy OpenGL jest w minimalnej, obsługiwanej wersji przez wybrane przez nas rozszerzenie:
`if(GLEE_VERSION_1_3){ ... }`
- sprawdzamy, czy rozszerzenie z którego chcemy skorzystać jest dostępne:

```
if(rozszerzenie) { ... }
```

1.2 Multiteksturowanie

Multiteksturowanie pozwala na użycie i połączenie ze sobą kilka tekstur na jednym obiekcie. Sprawdzamy, czy obsługiwany jest OpenGL w wersji min. 1.3 (GLEE_VERSION_1_3) oraz dostępne jest rozszerzenie GLEE_ARB_multitexture. Następnie sprawdzamy ile jednostek teksturowania jest obsługiwane (ile maksymalnie tekstur możemy użyć):

```
int maxUnits;  
glGetIntegerv(GL_MAX_TEXTURE_UNITS, &maxUnits);
```

Kolejnym krokiem jest włączenie n jednostek teksturowania:

```
glActiveTexture(GL_TEXTURE0); glEnable(GL_TEXTURE_2D);  
glActiveTexture(GL_TEXTURE1); glEnable(GL_TEXTURE_2D);  
...  
glActiveTexture(GL_TEXTUREn); glEnable(GL_TEXTURE_2D);
```

Następnie ustawiamy sposób łączenia tekstur:

```
glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, tryb);  
gdzie tryb to jedna z wartości:
```

- GL_MODULATE – mnożenie,
- GL_DECAL – nakładanie zgodnie z wartościami kanału alfa,
- GL_BLEND – odwracanie kolorów
- GL_REPLACE – zastępowanie.

Kolejnym krokiem jest załadowanie w standardowy sposób tekstur. W celu wyświetlenia tekstur należy zbindować tekstury i ustawić dla nich odpowiednie współrzędne, np.:

```
glMultiTexCoord2f(GL_TEXTURE0, 0, 0);  
glMultiTexCoord2f(GL_TEXTURE1, 0, 0);  
glMultiTexCoord2f(GL_TEXTURE2, 1, 0);
```

2 Zadania

1. Utwórz animację obracającej się skrzynki, która będzie miała nałożoną teksturę drewnianych desek (GL_TEXTURE0) oraz cieni (GL_TEXTURE1).