



POLITECHNIKA WROCŁAWSKA
Instytut Informatyki, Automatyki i Robotyki
Zakład Systemów Komputerowych

**Grafika komputerowa i komunikacja
człowiek - komputer**

Kurs: INEK00012L

Sprawozdanie z ćwiczenia nr 6

TEMAT ĆWICZENIA
OpenGL – teksturowanie powierzchni obiektów

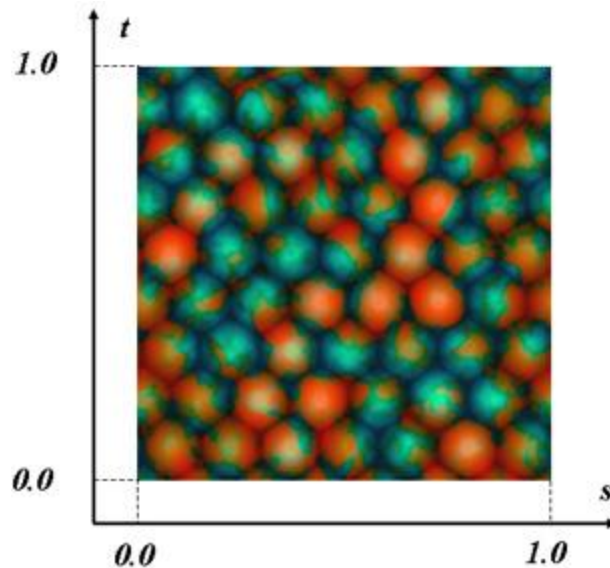
Wykonał:	Karol Pastewski 252798
Termin:	WT/TP 7.30-10.30
Data wykonania ćwiczenia:	07.12.2021r.
Data oddania sprawozdania:	14.12.2021r.
Ocena:	

Uwagi prowadzącego:

1. Wstęp teoretyczny

1.1. Teksturowanie

Teksturowanie polega na nanoszeniu na powierzchnię elementów zadanych w postaci map bitowych. Tekstury najczęściej są umieszczane w osobnym pliku (w naszym przypadku w pliku w formacie .tga). Współrzędne tekstury zostają rzutowane na kwadrat o boku długości 1. Następnie tekstura nakładana jest na obiekt w podobny sposób jak wyznaczanie punktów jajka – mapa bitowa jest dzielona na trójkąty i nakładana na obiekt 3D.



Rysunek 1 Wzorec w układzie współrzędnych tekstury

2. Nowe polecenia OpenGL

- **glEnable(GL_CULL_FACE);**
Funkcja pozwalająca ustawić teksturowanie tylko na jednej stronie obiektu.
- **glCullFace(GL_FRONT); glCullFace(GL_BACK);**
Po włączeniu GL_CULL_FACE możemy określić, która strona ma być widoczna. Domyślnie jest to wartość GL_BACK, ale można ustawić wartość przeciwną GL_FRONT, albo nawet GL_FRONT_AND_BACK, która pokazuje obie strony.
- **glEnable(GL_TEXTURE_2D);**
Włącza mechanizm teksturowania 2D.
- **glTexEnvf(GLenum target, GLenum pname, GLint param);**
Ustawia parametry środowiskowe tekstury, w naszym przypadku ustawia tryb teksturowania. Parametry oznaczają odpowiednio środowisko tekstury (musi być GL_TEXTURE_ENV), nazwę parametru (musi być GL_TEXTURE_ENV_MODE) oraz jej wartość (mogą być wartości GL_MODULATE, GL_DECAL lub GL_BLEND). Domyślna wartość to GL_MODULATE i z własnych testów wynika, że w trybie GL_DECAL na obiekcie nie widać efektu oświetlenia, a w GL_BLEND kolory są odwrócone.
- **glTexParameterf(GLenum target, GLenum pname, GLint param);**
Ustawia parametry tekstury. Parametry oznaczają typ tekstury (może być GL_TEXTURE_1D albo GL_TEXTURE_2D (używana w programie)), nazwę parametru oraz jej wartość.
- **glTexImage2D(**
 GLenum target,
 GLint level,
 GLenum internalformat,
 GLsizei width,
 GLsizei height,
 GLenum border,
 GLenum format,
 GLenum type,
 const GLvoid *pixels);
Specyfikuje dwuwymiarowy obraz tekstury. Kolejne argumenty oznaczają rodzaj tekstury (zawsze GL_TEXTURE_2D), poziom szczegółowości, ilości składowych koloru, szerokości obrazu, wysokości obrazu, szerokości ramki, format danych pikseli, typ danych pikseli i wskaźnik na teksturę w pamięci.
- **glTexCoord2f(GLfloat s, GLfloat t);**
Specyfikuje dwuwymiarowe (mogą być jedno- dwu- trój- albo czterowymiarowe) współrzędne tekstury.

3. Rozwiązanie zadania

```
584 //      —«TEKSTUROWANIE»—
585
586 // Teksturowanie będzie prowadzone tylko po jednej stronie ściany
587 | glEnable(GL_CULL_FACE);
588
589 // Włączenie mechanizmu teksturowania
590 | glEnable(GL_TEXTURE_2D);
591
592 // Ustalenie trybu teksturowania
593 | glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
594
595 // Określenie sposobu nakładania tekstur
596 | glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
597 | glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
```

Powyższy kod pochodzi z funkcji `myInit()` i pokazuje wszystkie funkcje potrzebne do zainicjalizowania teksturowania.

```
269 // Przeczytanie obrazu tekstury z pliku o nazwie tekstura2.tga
270 pBytes = textures::LoadTGAImage("tekstura2.tga", &ImWidth,
271 |                               &ImHeight, &ImComponents, &ImFormat);
272
273 //Zdefiniowanie tekstury 2-D
274 | glTexImage2D(GL_TEXTURE_2D, 0, ImComponents, ImWidth, ImHeight,
275 |             0, ImFormat, GL_UNSIGNED_BYTE, pBytes);
276
277 | free(pBytes);
```

W linijce 270 i 271 mamy wywołanie funkcji odpowiadającej za załadowanie pliku o nazwie `tekstura2.tga` (czyli naszego pliku z teksturą) do pamięci programu. Funkcja `LoadTGAImage()` została zaczerpnięta z instrukcji laboratoryjnej.

```
101 | glBegin(GL_TRIANGLES);
102
103 | glNormal3fv(ARRAY[i][j].normalized);
104 | glTexCoord2fv(ARRAY[i][j].texture);
105 | glVertex3fv(ARRAY[i][j].point3);
106
107 | glNormal3fv(ARRAY[i + 1][j + 1].normalized);
108 | glTexCoord2fv(ARRAY[i + 1][j + 1].texture);
109 | glVertex3fv(ARRAY[i + 1][j + 1].point3);
110
111 | glNormal3fv(ARRAY[i][j + 1].normalized);
112 | glTexCoord2fv(ARRAY[i][j + 1].texture);
113 | glVertex3fv(ARRAY[i][j + 1].point3);
114
115 | glEnd();
```

Zamieszczony powyżej fragment kodu odpowiada za opisanie jednej części trójkątów, z których składa się jajko. Za teksturowanie odpowiada funkcja `glTexCoord2fv()`, która przyjmuje tablicę współrzędnych do wycięcia fragmentu tekstury. Funkcje `glTexCoord2fv()` i `glNormal3fv()` były opisane w poprzednich sprawozdaniach.