

GRAFIKA KOMPUTEROWA

OpenGL - TEKSTURY

GENEROWANIE OBIEKTU TEKSTURY

Do wygenerowania obiektu tekstury służy funkcja `glGenTextures`.

```
GLuint tex; //identyfikator tekstury
glGenTextures(1, &tex);
```

Aby móc używać tekstury należy ją uczynić aktywną:

```
glBindTexture(GL_TEXTURE_2D, tex); // GL_TEXTURE_2D - gdy mamy do
//czynienia z teksturą dwuwymiarową
```

OBRAZ TEKSTURY

Obraz tekstury można załadować z pliku lub z tablicy pikseli. W przypadku ładowania z pliku należy użyć dodatkowej biblioteki, ponieważ OpenGL nie udostępnia odpowiednich funkcji. Do załadowania pikseli z tablicy służy funkcja `glTexImage2D`.

```
float pixels[] = { // tablica pikseli przedstawiająca
                  // obraz przykładowej tekstury
0.0f, 0.0f, 0.0f, 1.0f, 1.0f, 1.0f,
1.0f, 1.0f, 1.0f, 0.0f, 0.0f, 0.0f
};
int width = 2; height = 2; // szerokość i wysokość obrazu tekstury
glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width, height, 0, GL_RGB, GL_FLOAT, pixels);
```

WSPÓŁRZĘDNE TEKSTURY

Do pikseli tekstury odwołujemy się za pomocą współrzędnych należących do przedziału $\langle 0, 1 \rangle$. Lewy dolny róg tekstury to punkt $(0, 0)$ a prawy górny to $(1, 1)$. Współrzędne są wykorzystywane do pobierania informacji o kolorze (próbkiwanie, ang. *sampling*).

Współrzędne tekstury stanowią atrybuty wierzchołków (obok współrzędnych wierzchołków oraz kolorów).

```
GLfloat tex_coordinates[] = { //przykładowe współrzędne tekstury
    0.0f, 0.0f, //dla czterech wierzchołków
    1.0f, 0.0f,
    1.0f, 1.0f,
    0.0f, 1.0f, };
```

Współrzędne te muszą zostać przesłane do bufora na karcie graficznej:

```
GLuint vbo;
glGenBuffers(1, &vbo); //wygenerowanie bufora
glBindBuffer(GL_ARRAY_BUFFER, vbo); //aktywowanie bufora

glBufferData(GL_ARRAY_BUFFER, sizeof(tex_coordinates), tex_coordinates,
GL_STATIC_DRAW); //przesłanie danych
```

Podobnie jak w przypadku innych atrybutów wierzchołków (np. współrzędnych, kolorów) konieczne jest następnie:

1. pobranie uchwytu do odpowiedniego wektora wejściowego w shaderze wierzchołków (w przypadku naszego programu `texcoord`)

```
GLint texAttrib = glGetUniformLocation(shaderProgram, "texcoord");
```

2. określenie, w jaki sposób dane będą pobierane z tablicy

```
glVertexAttribPointer(texAttrib, 2, GL_FLOAT, GL_FALSE, 0, 0);
```

3. aktywowanie tablicy atrybutów wierzchołków

```
glEnableVertexAttribArray(texAttrib);
```

KONFIGUROWANIE TEKSTURY

Nakładanie

Jeżeli współrzędne tekstury przekraczają zakres $\langle 0, 1 \rangle$, możliwe są następujące sposoby nakładania tekstury:

- `GL_REPEAT` – wzór się powtarza
- `GL_MIRRORED_REPEAT` – wzór się powtarza a kolejne powtórzenia są lustrzanymi odbiciami
- `GL_CLAMP_TO_EDGE` – współrzędnym poza zakresem zostaną przypisane kolory z krawędzi tekstury
- `GL_CLAMP_TO_BORDER` - współrzędnym poza zakresem zostanie przypisany kolor obramowania

Ustawienie parametrów dotyczących nakładania:

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
//GL_TEXTURE_WRAP_S i GL_TEXTURE_WRAP_T określają
//wymiar, wzdłuż którego odbywa się wybrane nakładanie
```

W przypadku nakładania typu `GL_CLAMP_TO_BORDER` można określić kolor ramki w następujący sposób:

```
float color[] = { 1.0f, 0.0f, 0.0f, 1.0f };
glTexParameterfv(GL_TEXTURE_2D, GL_TEXTURE_BORDER_COLOR, color);
```

Filtrowanie

Jeżeli tekstura jest mniejsza lub większa niż obszar, na który jest nakładana, musi zostać odpowiednio rozciągnięta lub ściśnięta. Wtedy próbkowanie koloru może odbywać się na jeden z następujących sposobów:

- `GL_NEAREST` – pobierany jest kolor punktu leżącego najbliżej żądanych współrzędnych
- `GL_LINEAR` – wynikowy kolor jest średnią kolorów czterech punktów leżących wokół żądanych współrzędnych

Ustawienie parametrów tekstury dotyczących filtrowania:

```
glTexParameteri (GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);  
                //gdy obraz jest skalowany w dół  
glTexParameteri (GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);  
                //gdy obraz jest skalowany w górę
```