

Nama: Tsabita Syahida Khafid

NIM: 24060123130071

Lab : GKV E1

Praktikum 7 Tekstur dengan Gambar

1. main.cpp

Tujuan program ini yaitu menampilkan sebuah kubus 3D bertekstur dan berwarna yang dapat berputar secara otomatis menggunakan OpenGL dengan library GLFW untuk membuat window dan GLEW untuk mengakses ekstensi OpenGL modern. Dua sisi kubus diberi tekstur, sementara empat sisi lainnya diberi warna polos.

- handleKeypress

Fungsi callback untuk menangani input keyboard.

Jika tombol ESC ditekan, maka `glfwSetWindowShouldClose()` dipanggil untuk menutup aplikasi.

- loadTexture

Digunakan untuk membuat tekstur OpenGL dari objek Image*.

Langkah-langkah:

1. Generate ID tekstur.
2. Bind tekstur ke `GL_TEXTURE_2D`.
3. Isi data tekstur (`glTexImage2D`) dari pixel gambar.
4. Atur parameter filter (linear).
5. Return ID tekstur.

- initRendering

Inisialisasi fitur OpenGL yang dibutuhkan:

- `GL_DEPTH_TEST`: untuk depth-buffer (3D rendering).
- `GL_LIGHTING`, `GL_LIGHT0`: untuk pencahayaan
- `GL_NORMALIZE`: menormalkan vektor normal.
- `GL_COLOR_MATERIAL`: agar warna objek mengikuti `glColor`.

Load gambar BMP lalu ubah ke tekstur OpenGL menggunakan `loadTexture`.

- drawScene

Fungsi utama untuk menggambar objek ke layar setiap frame.

Proses:

1. Clear buffer warna dan depth.
2. Reset ModelView matrix, lalu geser kamera ke belakang.
3. Atur pencahayaan (ambient dan diffuse).
4. Rotasi objek terhadap angle.
5. Gambar kubus:
 - 4 sisi menggunakan warna polos (`glColor3f` dan `glNormal3f`).
 - 2 sisi (depan dan belakang) menggunakan tekstur.
6. Aktifkan dan nonaktifkan `GL_TEXTURE_2D` hanya saat menggambar sisi yang diberi tekstur.

- main

Alur utama program:

1. Inisialisasi GLFW dan GLEW:
GLFW diinisialisasi.
Buat window 800x600.
Set context dan inisialisasi GLEW.
2. Pengaturan callback input:
Tombol keyboard dikontrol dengan handleKeypress.
3. Set proyeksi perspektif:
Gunakan gluPerspective untuk tampilan 3D.
4. Inisialisasi OpenGL:
Panggil initRendering.
5. Main Loop:
 - Selama window tidak ditutup:
Tambahkan rotasi angle.
Panggil drawScene.
Tukar buffer (swapBuffers).
Poll event input.
 - Cleanup:
Hancurkan window dan akhiri GLFW.

2. Imageloader.cpp

Tujuan kode ini untuk membaca file gambar BMP (khusus 24-bit) dan memuat data pixel-nya ke dalam objek Image, yang nantinya akan dipakai sebagai tekstur dalam OpenGL.

Proses:

1. Buka file BMP dalam mode biner.
2. Validasi file:
Periksa apakah file diawali dengan karakter 'B' dan 'M'.
Pastikan gambar bertipe 24-bit (RGB).
3. Baca metadata dari header:
Ukuran gambar (width, height).
Offset awal data piksel (dataOffset).
4. Baca data piksel mentah dari file dimulai dari offset.
5. Konversi format warna dari BMP (BGR) ke RGB.
6. Simpan data piksel ke dalam objek Image.
7. Kembalikan objek Image untuk digunakan dalam OpenGL sebagai tekstur.

3. Imageloader.h

Header file imageloader.h ini berfungsi sebagai deklarasi antarmuka (interface) untuk kelas dan fungsi yang digunakan dalam proses pemuatan gambar BMP ke dalam memori.

Proses:

1. Menyediakan blueprint (kerangka) kelas Image yang berisi:
Ukuran gambar (width, height).
Pointer ke array data pixel (pixels) dalam format RGB.
2. Mendeklarasikan konstruktor dan destruktur:
Konstruktor: mengalokasikan memori untuk data pixel.
Destruktor: membersihkan memori saat objek dihapus.
3. Mendeklarasikan fungsi loadBMP(const char*):
Akan membaca file BMP 24-bit.
Mengisi objek Image dengan data RGB dari file tersebut.
Mengembalikan pointer ke objek Image.
4. Menggunakan header guard (#ifndef ... #endif):
Mencegah file ini dibaca lebih dari satu kali saat di-include.

Praktikum 8 Bayangan

1. main.cpp

Program ini menampilkan sebuah kerucut 3D dengan pencahayaan. Bayangan dinamis dari kerucut ke lantai menggunakan proyeksi matriks. Interaksi pengguna untuk mengatur posisi cahaya secara real-time. Demonstrasi penting konsep shadow projection dalam grafika komputer menggunakan OpenGL fixed pipeline.

• help

Menampilkan informasi kontrol program di terminal:

W/S/A/D/Q/E: Gerakkan posisi cahaya.

ESC: Keluar dari program.

• draw

Menggambarkan objek utama — sebuah kerucut menggunakan gluCylinder.

Menggunakan gluNewQuadric() untuk menggambarkan permukaan silindris.

• glShadowProjection

Membuat matriks proyeksi bayangan:

Berdasarkan posisi cahaya l, bidang e dan n.

Menghitung proyeksi geometris bayangan dengan rumus khusus dan memanggil glMultMatrixf(mat) untuk mengalikan transformasi matriks OpenGL aktif.

• render()

Fungsi inti untuk menggambarkan seluruh adegan:

Clear screen dan set latar belakang.

Gambar titik sumber cahaya (kuning).

Gambar bidang lantai tempat bayangan akan jatuh.

Gambar objek asli (kerucut, berwarna biru).

Hitung proyeksi bayangan dan gambar bayangannya di lantai (berwarna abu-abu gelap).

Efek pencahayaan (glEnable(GL_LIGHTING)) diatur aktif untuk objek asli, tapi dimatikan saat menggambarkan bayangan agar terlihat gelap.

• keypress

Menangani input keyboard:

ESC: keluar.

W/S/A/D/Q/E: mengatur posisi cahaya.

H: tampilkan bantuan.

- idle

Memutar objek secara perlahan.

Memanggil render() untuk menggambar ulang.

glfwSwapBuffers(): menampilkan hasil render ke layar.

- resize

Mengatur ulang viewport dan perspektif saat ukuran jendela berubah.

gluPerspective(...): mengatur sudut pandang kamera.

- main

Fungsi utama program:

1. Inisialisasi GLFW dan GLEW.

2. Buat window berukuran 400x400.

3. Atur callback:

Keyboard → keypress

Ukuran jendela → resize

4. Aktifkan fitur OpenGL:

Pencahayaannya (GL_LIGHTING, GL_LIGHT0)

Normalisasi, depth test, dll.

5. Atur pencahayaan ambient dan diffuse.

6. Set transformasi awal (kamera menjauh ke $z = -150$).

7. Tampilkan bantuan (help()).

8. Masuk ke loop utama:

Selama jendela belum ditutup → panggil idle() dan polling event.

9. Setelah loop selesai → tutup window dan terminasi GLFW.

Praktikum 9 Interaksi Antar Objek

1. main.cpp (menendang bola)

Tujuan program ini Mensimulasikan tendangan bola oleh kaki dalam bentuk animasi.

Interaktif: bisa menggerakkan kamera, memutar scene, dan mengontrol aksi tendangan.

Menampilkan pencahayaan, grid lantai, objek balok (kaki) dan bola (dengan pergerakan dan rotasi).

- init

Inisialisasi awal OpenGL:

Mengaktifkan depth test (untuk tampilan 3D yang benar).

Mengatur GL_FILL agar objek ditampilkan penuh (bukan wireframe).

Membuat objek quadric (untuk bola/sphere) dan mengatur normal serta teksturnya.

- reshape

Mengatur ulang viewport dan kamera saat jendela diubah ukurannya:

Menentukan rasio aspek (ratio) dan area tampilan (glViewport).

Mengatur perspektif kamera (gluPerspective).

Memasang kamera dengan gluLookAt.

- orientMe

Memutar arah pandang kamera horizontal:

Mengubah vektor arah lx dan lz berdasarkan sudut.

Mengatur ulang kamera menggunakan gluLookAt.

- moveMeFlat

Menggerakkan kamera maju/mundur:

Kamera berpindah posisi dalam arah vektor lx, lz.

Diperbarui dengan gluLookAt.

- keyCallback

Menangani input keyboard utama:

Rotasi objek (rotAngleX/Y/Z).

Gerak horizontal kaki dan bola (O, P).

Animasi tendangan (K).

Reset seluruh posisi (SPACE).

Keluar program (ESCAPE).

- specialKeyCallback

Menangani tombol panah keyboard (arah kamera):

Tombol panah mengubah deltaMove (maju/mundur) dan deltaAngle (putar kamera).

Tombol ini dikendalikan di loop utama juga.

- lighting

Mengatur pencahayaan dalam scene:

Aktifkan pencahayaan dan normalisasi.

Mengatur sumber cahaya dan material objek (ambient, diffuse, specular).

Membuat efek 3D realistis pada objek.

- Grid

Menggambarkan grid lantai:

Menggambarkan garis-garis sumbu X dan Z sebagai referensi tanah/grid.

- Balok

Membuat balok (kaki):

Menggambarkan 6 sisi balok menggunakan GL_QUADS.

Digunakan untuk menggambarkan kaki penendang.

- pergerakanKaki

Animasi kaki saat menendang:

kick == 1: kaki berayun ke atas.

kick == 2: kaki menendang ke bawah, dan jika menyentuh bola, bola mulai bergulir.

kick == 3: kaki kembali ke posisi semula.

Mengontrol sudut rotKaki.

- pergerakanBola

Animasi bola setelah ditendang:

Jika $\text{roll} == 1$, bola bergeser ke kiri secara perlahan sejauh jarak.

- Object

Menggambar objek utama (kaki + bola):

Kaki:

Translasi ke posisi posXKaki , posYKaki .

Rotasi animasi ditentukan oleh rotKaki .

Bola:

Bergerak tergantung status roll .

Digambar dengan gluSphere .

- display

Fungsi render utama (dipanggil setiap frame):

Membersihkan buffer.

Gerakkan kamera jika tombol panah ditekan ($\text{moveMeFlat/orientMe}$).

Terapkan rotasi global (rotAngleX/Y/Z) ke seluruh scene.

Gambar grid dan objek (kaki + bola).

Tukar buffer tampilan.

- main

Fungsi utama program:

1. Inisialisasi GLFW dan buat window.
2. Inisialisasi GLEW
3. Atur callback:
Keyboard (keyCallback)
Resize (reshape)
4. Inisialisasi pencahayaan (lighting()) dan properti awal (init()).
5. Setup kamera awal (reshape).
6. Loop utama:
Panggil display() setiap frame.
Polling event GLFW.
Tangani input kamera (arrow key).
7. Setelah keluar dari loop, hancurkan window dan tutup program.

2. main.cpp (untuk memegang objek)

Tujuan dari program ini adalah mensimulasikan interaksi antara sebuah robot (atau karakter berbentuk balok) dan sebuah kotak dalam lingkungan 3D menggunakan OpenGL, di mana pengguna dapat menggerakkan karakter, memanipulasi tangan robot untuk mengambil dan menjatuhkan kotak.

- init

Inisialisasi pengaturan awal OpenGL:

Mengaktifkan GL_DEPTH_TEST untuk kedalaman (3D).

Atur polygon mode ke fill (bukan wireframe).

Inisialisasi quadric object (untuk objek 3D seperti bola, tabung jika dipakai nanti).

Konfigurasi quadric: normal smooth dan bisa di-texture

- reshape

Menyesuaikan viewport dan perspektif saat jendela diubah ukurannya:

Hitung rasio aspek.

Atur proyeksi perspektif dengan `gluPerspective`.

Posisikan kamera dengan `gluLookAt`

- `orientMe`

Mengubah arah pandang kamera (horizontal):

Hitung arah kamera (lx , lz) berdasarkan angle.

Update posisi kamera dengan `gluLookAt`.

- `moveMeFlat`

Menggerakkan kamera maju/mundur berdasarkan arah pandang:

Update posisi x dan z dengan arah lx dan lz .

Update kamera dengan `gluLookAt`.

- `keyCallback`

Menangani input keyboard biasa (untuk rotasi dan kontrol objek):

Tombol WASD + QE: rotasi objek.

Tombol O/P: gerakkan "badan" ke kiri/kanan.

Tombol G: mulai gerakan tangan untuk grab.

Tombol T: drop kotak.

Tombol SPACE: reset semua posisi dan status.

Tombol ESCAPE: keluar dari aplikasi.

- `specialKeyCallback`

Menangani input keyboard panah untuk menggerakkan kamera:

Ubah nilai `deltaMove` (maju/mundur).

Ubah nilai `deltaAngle` (rotasi kamera).

- `lighting`

Mengatur pencahayaan OpenGL:

Aktifkan lighting, normalisasi, dan material.

Set parameter cahaya: ambient, diffuse, specular, posisi.

- `Grid dan Grid2`

`Grid()`: Membuat garis-garis grid di tanah (garis bantu).

`Grid2()`: Membuat lantai putih sebagai dasar datar.

- `Balok`

Menggambar balok 3D:

Gambar 6 sisi kubus dengan `glBegin(GL_QUADS)`.

- `perubahKotak`

Menurunkan kotak secara bertahap saat di-drop:

Kurangi `posYKotak` jika sedang drop dan grab.

Matikan status bring, hit, dan grab setelah kotak selesai turun.

- `pengubahTangan`

Animasi gerakan tangan robot:

Cek apakah posisi badan mendekati kotak \rightarrow hit.

Jika hit dan grab, maka `bring = 1`.

Tangan bergerak:

gerakTangan = 1: rotasi tangan ke bawah (ambil).

gerakTangan = 2: tangan memutar ke dalam (menggenggam).

gerakTangan = 3: lepas genggaman.

gerakTangan = 4: tangan kembali ke posisi awal.

- Object

Menggambar semua objek dalam scene:

Objek tetap (meja/latar) → warna biru tua.

Kotak merah → bisa digerakkan dan dijatuhkan.

Badan biru + tangan hijau → bergerak dan mengangkat/menggenggam.

- display

Fungsi render utama, menggambar frame setiap saat:

Clear buffer.

Update kamera jika bergerak.

Rotasi seluruh scene dengan rotAngleX/Y/Z.

Gambar grid, lantai, dan objek.

Swap buffer untuk menampilkan hasil.

- main

Alur utama program:

1. Inisialisasi GLFW dan buat jendela.
2. Set callback untuk input keyboard dan resize window.
3. Inisialisasi GLEW.
4. Atur pencahayaan dan inisialisasi OpenGL.
5. Loop utama (while !windowShouldClose):
 - Tampilkan (display()).
 - Ambil input (pollEvents).
 - Update variabel input jika tombol panah ditekan/dilepas.
6. Setelah keluar dari loop:
 - Hancurkan window dan akhiri GLFW.