

Nama : Radityantari Kusuma Wardani

NIM : 24060123130059

Lab : E1

Tugas Praktikum 6

1. Kode Tekstur dengan Gambar

- Fungsi `handleKeyPress` untuk menutup jendela saat tombol `Escape` ditekan. Fungsi `loadTexture` untuk memuat gambar BMP menjadi tekstur OpenGL. Fungsi `initRendering` untuk mengaktifkan fitur rendering seperti pencahayaan dan `depth`, serta memuat tekstur dari file BMP. Fungsi `drawScene` untuk menggambar kubus 3D berwarna dengan pencahayaan dan sebagian permukaan bertekstur. Fungsi `main` untuk menginisialisasi `GLFW` dan `GLEW`, membuat jendela, mengatur proyeksi 3D, lalu menjalankan loop utama untuk menggambar dan memutar kubus.
- Tekstur yang dibuat yaitu file gambar `852.bmp` dan diterapkan pada 2 sisi kubus.
- Cara kerja tekstur dimulai dari pemuatan file gambar BMP menggunakan fungsi `loadBMP`, yang membaca data piksel dari file dan menyimpannya dalam objek `Image`. Gambar ini kemudian dikonversi menjadi tekstur OpenGL melalui fungsi `loadTexture`, di mana data gambar diunggah ke GPU dan diberi ID tekstur. Setelah itu, saat inisialisasi rendering, tekstur ini diaktifkan dan disiapkan untuk digunakan. Penerapan tekstur terjadi saat proses menggambar kubus, khususnya pada dua sisi kubus (depan dan belakang). Tekstur diterapkan dengan mengaktifkan mode tekstur 2D, mengikat ID tekstur, serta memberikan koordinat tekstur yang sesuai pada setiap vertex sisi kubus. Setelah gambar ditampilkan, mode tekstur dimatikan kembali untuk mencegah penerapan tekstur pada sisi lainnya. Alur ini memungkinkan gambar BMP ditampilkan secara visual pada permukaan kubus dengan tepat.

2. Kode Bayangan

- Fungsi `help` menampilkan panduan kontrol untuk pengguna dalam menjalankan simulasi, termasuk tombol untuk mengatur posisi sumber cahaya. Fungsi `draw` bertugas untuk menggambar objek utama, yaitu sebuah kerucut menggunakan `gluCylinder`. Untuk menghitung dan menerapkan proyeksi bayangan dari objek ke bidang datar, digunakan fungsi `glShadowProjection`, yang membangun dan mengalikan matriks proyeksi bayangan berdasarkan posisi cahaya, bidang, dan normal permukaan. Fungsi `render` bertanggung jawab terhadap proses render keseluruhan frame. Fungsi ini membersihkan layar, menggambar sumber cahaya sebagai titik kuning, menggambar bidang datar berwarna abu-abu muda, dan menggambar objek kerucut biru. Selain itu, objek bayangan juga digambar di posisi proyeksinya menggunakan warna abu-abu tua, dengan sebelumnya menerapkan matriks proyeksi bayangan. Interaksi pengguna dikendalikan oleh fungsi `keypress`, yang merespons input tombol keyboard. Tombol `W`, `S`, `A`, `D`, `Q`, dan `E` mengubah posisi cahaya dalam tiga dimensi, sedangkan tombol `ESC` menutup aplikasi dan tombol `H` menampilkan bantuan. Fungsi `idle` digunakan untuk memperbarui rotasi objek secara perlahan agar animasi berjalan, serta memanggil `render` dan menukar buffer untuk menampilkan hasil render.

- Bayangan dalam program ini dihasilkan dengan teknik *shadow projection*, yaitu memproyeksikan geometri objek (kerucut) ke sebuah bidang datar menggunakan transformasi matriks. Proses ini dilakukan setelah objek utama digambar. Program pertama-tama menyimpan transformasi saat ini dengan `glPushMatrix`, lalu memanggil fungsi `glShadowProjection` untuk membentuk matriks proyeksi berdasarkan posisi cahaya, titik di bidang, dan normal permukaan bidang. Setelah itu, objek digambar kembali (menggunakan fungsi `draw`), tetapi kali ini dengan warna abu-abu gelap dan *lighting* dimatikan agar menyerupai bayangan. Terakhir, transformasi dipulihkan dengan `glPopMatrix` agar tidak memengaruhi elemen lain. Fungsi `resize` menangani penyesuaian viewport dan proyeksi saat ukuran jendela berubah, memastikan proporsi dan perspektif tetap sesuai. Di bagian `main`, program melakukan inisialisasi GLFW dan GLEW, membuat jendela OpenGL, mengatur callback, dan menginisialisasi pencahayaan serta proyeksi awal. Program kemudian memasuki loop utama di mana fungsi `idle` terus dijalankan hingga jendela ditutup. Setelah keluar dari loop, jendela dihancurkan dan GLFW diterminasi.
- Fungsi `glShadowProjection` menghitung matriks proyeksi bayangan berdasarkan posisi cahaya, titik bidang, dan normal bidang. Pertama, dihitung nilai *dot product* antara vektor normal dengan posisi cahaya, dan konstanta `c` yang menyatakan kedalaman relatif antara titik bidang dan cahaya terhadap bidang tersebut. Matriks 4x4 kemudian dibentuk dengan rumus matematis dari *shadow matrix projection*, yang merepresentasikan transformasi proyeksi ke bidang. Terakhir, `glmMultMatrixf` digunakan untuk mengalikan matriks proyeksi tersebut ke *modelview matrix* saat ini, sehingga objek yang digambar setelahnya akan tampak seperti bayangan yang diproyeksikan ke bidang.

3. Kode Interaksi Antar Objek bagian 1 (menendang bola)

Fungsi `init` mengatur mode rendering dasar seperti depth testing, mode polygon fill, dan membuat objek quadric untuk menggambar bentuk seperti bola dengan `gluSphere`. Fungsi `reshape` menangani perubahan ukuran jendela dan mengatur ulang perspektif menggunakan `gluPerspective` serta kamera menggunakan `gluLookAt`. Fungsi `orientMe` mengatur arah pandang kamera berdasarkan sudut rotasi horizontal, sedangkan `moveMeFlat` memungkinkan pergerakan kamera maju-mundur di sumbu x-z. Fungsi `keyCallback` dan `specialKeyCallback` menangani input keyboard. `keyCallback` mengatur rotasi objek, pergerakan kaki dan bola, serta reset dan keluar program. `specialKeyCallback` menangani input panah untuk rotasi kamera dan gerakan maju-mundur. Fungsi `lighting` mengaktifkan pencahayaan OpenGL dan mengatur properti cahaya serta material untuk memberikan efek visual yang lebih realistis. Fungsi `Grid` menggambar garis-garis membentuk grid pada bidang tanah untuk membantu orientasi objek dalam ruang. Fungsi `Balok` menggambar bentuk balok 3D (digunakan sebagai kaki), dengan setiap sisi digambar menggunakan `GL_QUADS`. Fungsi `pergerakanKaki` menganimasikan gerakan kaki saat menendang bola dengan perubahan nilai sudut `rotKaki` secara bertahap, sementara `pergerakanBola` menggerakkan bola jika terkena tendangan. Fungsi `Object` adalah gabungan dari kedua objek, yaitu kaki dan bola, yang masing-masing diberi transformasi dan warna. Fungsi `display` bertanggung jawab atas proses rendering

tiap frame. Ia membersihkan buffer, mengatur ulang posisi kamera jika ada input, memutar objek sesuai input rotasi, menggambar grid dan objek, serta akhirnya menampilkan frame dengan `glfwSwapBuffers`.

4. Kode Interaksi Antar Objek bagian 2 (memegang objek)

Fungsi `init` mengaktifkan depth testing untuk menjaga urutan kedalaman objek dalam tampilan 3D. Kemudian, `gluNewQuadric` membuat objek quadric baru yang digunakan untuk menggambar bentuk silinder/sphere, dan menyetel propertinya agar memiliki permukaan halus dan bisa diberi tekstur. Fungsi `reshape` dipanggil saat ukuran jendela berubah. Ia mengatur viewport dan mengatur proyeksi perspektif agar objek tetap terlihat proporsional. `gluLookAt` digunakan untuk mengatur posisi dan arah pandang kamera berdasarkan koordinat x , y , z , serta arah vektor lx , ly , lz . Fungsi `orientMe` mengubah orientasi kamera berdasarkan sudut `ang`, mengatur vektor arah pandang lx dan lz menggunakan fungsi sinus dan kosinus, lalu memperbarui tampilan kamera menggunakan `gluLookAt`. Fungsi `moveMeFlat` menggerakkan posisi kamera maju atau mundur berdasarkan nilai `i`, dengan arah gerak mengikuti vektor arah pandang kamera, lalu memperbarui pandangan dengan `gluLookAt`. Fungsi `keyCallback` menangani input tombol keyboard biasa. Misalnya, tombol W, A, S, D mengatur rotasi objek di sumbu X dan Y. Tombol Q dan E mengatur rotasi pada sumbu Z. Tombol O dan P digunakan untuk memindahkan badan karakter ke kiri dan kanan, serta jika sedang membawa objek, maka objek ikut bergerak. Tombol G memicu gerakan tangan, dan tombol T untuk melepaskan objek jika posisi badan mencukupi. Tombol SPACE mengatur ulang semua variabel ke kondisi awal, dan ESC untuk menutup jendela. Fungsi `specialKeyCallback` menangani input dari tombol panah (atas, bawah, kiri, kanan). Tombol atas/bawah mengatur pergerakan kamera maju/mundur dengan `deltaMove`, sedangkan kiri/kanan mengatur rotasi kamera dengan `deltaAngle`. Fungsi `lighting` mengatur sistem pencahayaan menggunakan OpenGL. Mengaktifkan pencahayaan, menentukan parameter cahaya (ambient, diffuse, specular, dan posisi), serta menetapkan sifat material seperti kilap dan warna. Fungsi `Grid` menggambar garis-garis horizontal dan vertikal membentuk grid (garis bantu) di lantai.

Fungsi `Grid2` menggambar bidang datar putih sebagai lantai tambahan. Fungsi **Balok** menggambar sebuah balok 3D berbentuk kotak dengan dimensi yang ditentukan. Tiap sisi balok dibuat dengan `glBegin(GL_QUADS)` dan enam sisi digambar satu per satu. Fungsi `perubahKotak` mengatur pergerakan objek "kotak" jika sedang dibawa (`grab == 1`) dan status `drop == 1`. Kotak diturunkan sedikit demi sedikit hingga mencapai ketinggian tertentu, lalu status interaksi dengan kotak dimatikan. Fungsi `pengubahTangan` menangani animasi gerakan tangan karakter dalam tiga tahap. Jika posisi badan sejajar dengan posisi kotak (`posXBadan == 4`), maka status `hit` akan aktif. Jika `hit` dan `grab` aktif, maka karakter mulai "membawa" kotak. Gerakan tangan diatur secara bertahap dari `rotTangan1` hingga `rotTangan2`, dan bila gerakan sudah selesai, maka status `bring` diaktifkan dan rotasi kembali di-reset saat menjatuhkan objek.