

ДОДАТОК А

Апробація результатів роботи

УДК 004.896

АВТОМАТИЗОВАНИЙ МОДУЛЬ КЕРУВАННЯ СКЛАДСЬКИМ РОБОТОМ-КОНТРОЛЕРОМ

Шевченко А.А., Хрустальова С.В.

e-mail: anna.shevchenko1@nure.ua, sofiia.khrustalova@nure.ua

Харківський національний університет радіоелектроніки, каф. КІТАР
м. Харків, Україна

This article presents an analysis and idea for improving the automated control module of a warehouse robot-controller to meet the needs of medium-sized Internet businesses selling household goods online. An overview of existing warehouse automation systems from various companies is presented, their advantages and disadvantages are identified, and conclusions are drawn. An idea for improving the identified disadvantages and methods for its implementation are considered, which could increase the efficiency of the warehouse and improve the business.

В умовах постійного розвитку онлайн-продажів актуальність автоматизації складських приміщень викликає все більше уваги у суб'єктів господарювання, підприємців, громадських та міжнародних організацій. Багато підприємств готові впроваджувати новітні технології, адже це значно підвищить їх конкурентоздатність, допоможе збільшити продуктивність та знизити ризики помилок. [1] Як показує статистика, деякі склади здатні обробляти більш ніж 10 000 товарних одиниць за день, та в середньому працівники проходять від 12 до 20 кілометрів між полицями за одну зміну. Тому не є дивно, що в таких умовах важко швидко та правильно обробляти замовлення і вірогідність помилок, пов'язаних із людським фактором, зростає. Отже, як закордонні, так і вітчизняні компанії повинні чітко розуміти методи, шляхи та способи досягнення найвищих конкурентних позицій на основі впровадження новітніх технологій, для покращення роботи всієї галузі.

Автоматизація потребує застосування індивідуального підходу, оскільки кожен склад чи складський комплекс має свої особливості навіть у межах однієї галузі. Якщо розглядається середній бізнес із продажу побутових товарів в онлайн-магазині, то габарити такого складського приміщення можуть становити від 100 до 500 м².

Для оптимізації робочого простору необхідно розуміти як будуть побудовані слоти для зберігання. Є декілька прикладів, які можуть бути розглянуті для аналізу розподілу складського простору. Одним з таких є центр дистрибуції в Андовері «Ocado». Особливістю цієї автоматичної системи є те, що роботи пересуваються рейками зверху над осередками з товаром уздовж спеціальної розмітки у вигляді сітки. [2] Таке споживання простору є не дуже раціональним

з точки зору об'єму площі складу, оскільки за відсутності можливості використання великих приміщень, такий спосіб розміщення комірок з товаром не буде ефективним.

Тож можна зробити висновок, що в даному випадку треба орієнтуватись на роботів, які переміщуються підлогою та працюють із вертикальним розміщенням товару по складському приміщенню. У такому випадку можна розглянути AGV (Automated Guided Vehicle) – автоматизований керований транспортний засіб, що використовується на складах, у виробничих лініях та логістичних центрах. Такі роботи зазвичай використовують трекові системи, зокрема магнітну стрічку або навігацію за QR-кодами. [3] Вони прості у налаштуванні та експлуатації, дешевші, ніж деякі інші подібні роботи, та мають високу точність руху по заданих маршрутах. Але AGV мають і деякі недоліки, до яких можна віднести те, що вони є негнучкими в умовах змін на складі бо вони фізично залежать від інфраструктури складу. Також вони мають труднощі з об'їздом перешкод на своєму шляху через використання спрощених сенсорів (ультразвукові та ін.), які лише зупиняють AGV та відсутність алгоритмів динамічного ухилення від перешкод. Також, трекові системи, які прокладені для руху цих роботів не завжди є зручними для використання, оскільки, наприклад, якщо це кабель канал на підлозі, то він може перешкоджати руху інших транспортних засобів.

Для вирішення вище перелічених недоліків пропонуються наступні зміни: впровадити гібридну навігацію шляхом додавання LiDAR (Light Detection and Ranging – це технологія дистанційного зондування, яка використовує лазерне випромінювання для вимірювання відстаней до об'єктів, система складається з лазера, датчика для реєстрації відбитого світла та системи обробки даних) [4] та камери, щоб AGV могли розпізнавати середовище, використати SLAM (Simultaneous Localization and Mapping. Така концепція об'єднує два взаємопов'язані процеси – навігацію та створення карт – в єдиний обчислювальний цикл, що дозволяє інтелектуальним роботам збирати інформацію про навколишнє середовище, будувати карту, визначати своє місцезнаходження та орієнтуватися в просторі) [5] для створення карти складу, а також розробити алгоритми автоматичної корекції маршруту при виявленні перешкод на шляху AGV.

Таким чином, проаналізовано існуючі підходи щодо керування складським роботом-контролером. Запропоновано шляхи вдосконалення модуля керування складським роботом-контролером за рахунок впровадження адаптивної навігації для орієнтування робота в навколишньому середовищі та оптимізації його маршрутів для усунення заторів у складському приміщенні, що підвищить ефективність процесу складання та доставлення товару інтернет-магазину побутових речей до точки сортування та пакування товарів.

Список використаних джерел:

1. Podra O., Homza K. Modern technologies of automation of warehouse activities of enterprises. Management and Entrepreneurship in Ukraine: the stages of formation and problems of development. 2022. Т. 2022, № 2. С. 70–78.

2. 5 прикладів впровадження роботів на складах: досвід компаній. WareTeKa. URL: <https://wareteka.com.ua/uk/blog/roboti-na-skladah-prikladi-avtomatizaciyi/> (дата звернення: 20.02.2025)

3. Raevskaya N. AGV - Складські роботи. Інтеграція систем і даних. Впровадження ERP систем. TQM systems. URL: <https://tqm.com.ua/ua/likbez/interview-ua/avg-roboty-na-skladi> (дата звернення: 23.02.2025).

4. Yevsieiev , V., Maksymova , S., Gurin , D., & Alkhalaileh, A. (2024). Data Fusion Research for Collaborative Robots-Manipulators within Industry 5.0. ACUMEN: International Journal of Multidisciplinary Research, 1(4), 125–137.

5. І. Невлюдов, С. Новоселов, К. Сухачов. Метод одночасної локалізації та картографування для побудови 2,5D-карти навколишнього середовища засобами ROS. Сучасний стан наукових досліджень та технологій в промисловості. 2023. №2 (24). С. 145–160.

ДОДАТОК Б

Код програми для технічної моделі складського робота у середовищі Tinkercad

```
#include <Servo.h>

Servo headServo;
int servoPin = 8;

int trigScan = A0;
int echoScan = A1;

float scanDistance;
float scanDuration;

int FirstPlus = 3;
int FirstMinus = 9;
int SecondPlus = 10;
int SecondMinus = 11;
int trigPin1 = 5;
int echoPin1 = 4;
int trigPin2 = 7;
int echoPin2 = 6;
int trigPin3 = 13;
int echoPin3 = 12;
float duration1,duration2,duration3, cm1,cm2,cm3;

void setup()
{
```

```
Serial.begin (9600);
pinMode(FirstPlus, OUTPUT);
pinMode(FirstMinus, OUTPUT);
pinMode(SecondPlus, OUTPUT);
pinMode(SecondMinus, OUTPUT);
pinMode(trigPin1, OUTPUT);
    pinMode(echoPin1, INPUT);
pinMode(trigPin2, OUTPUT);
    pinMode(echoPin2, INPUT);
pinMode(trigPin3, OUTPUT);
    pinMode(echoPin3, INPUT);
headServo.attach(servoPin);
pinMode(trigScan, OUTPUT);
pinMode(echoScan, INPUT);
}
void Ahead()
{
analogWrite(FirstPlus, 255);
digitalWrite(FirstMinus, LOW);
    analogWrite(SecondPlus, 255);
    digitalWrite(SecondMinus, LOW);
delay(2000);
}
void Back()
{
digitalWrite(FirstPlus, LOW);
analogWrite(FirstMinus, 200);
    digitalWrite(SecondPlus, LOW);
    analogWrite(SecondMinus, 200);
delay(2000);
```

```
}  
void Right()  
{  
  analogWrite(FirstPlus, 180);  
  digitalWrite(FirstMinus, LOW);  
  digitalWrite(SecondPlus, LOW);  
  analogWrite(SecondMinus, 180);  
  delay(2000);  
}  
void Left()  
{  
  analogWrite(SecondPlus, 180);  
  digitalWrite(SecondMinus, LOW);  
  digitalWrite(FirstPlus, LOW);  
  analogWrite(FirstMinus, 180);  
  delay(2000);  
}  
void Stop()  
{  
  digitalWrite(SecondPlus, LOW);  
  digitalWrite(SecondMinus, LOW);  
  digitalWrite(FirstPlus, LOW);  
  digitalWrite(FirstMinus, LOW);  
  delay(2000);  
}  
  
void scanSurroundings() {  
  Serial.println("=== SLAM SCAN START ===");  
  for (int angle = 0; angle <= 180; angle += 30) {  
    headServo.write(angle);  
  }  
}
```

```
delay(200);
digitalWrite(trigScan, LOW);
delayMicroseconds(5);
digitalWrite(trigScan, HIGH);
delayMicroseconds(10);
digitalWrite(trigScan, LOW);
scanDuration = pulseIn(echoScan, HIGH);
scanDistance = (scanDuration / 2) / 29.1;

Serial.print("Angle:");
Serial.print(angle);
Serial.print("°, Distance:");
Serial.print(scanDistance);
Serial.println("cm");
}
Serial.println("=== SLAM SCAN END ===");
}

void loop()
{
digitalWrite(trigPin1, LOW);
delayMicroseconds(5);
digitalWrite(trigPin1, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin1, LOW);
duration1 = pulseIn(echoPin1, HIGH);
cm1 = (duration1/2) / 29.1;

digitalWrite(trigPin2, LOW);
delayMicroseconds(5);
```

```

digitalWrite(trigPin2, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin2, LOW);
duration2 = pulseIn(echoPin2, HIGH);
cm2 = (duration2/2) / 29.1;

digitalWrite(trigPin3, LOW);
delayMicroseconds(5);
digitalWrite(trigPin3, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin3, LOW);
duration3 = pulseIn(echoPin3, HIGH);
cm3 = (duration3/2) / 29.1;

if (cm1>=51 && cm1<=335 && cm2>=51 && cm2<=335 && cm3>=51 &&
cm3<=335)
{
  Ahead();
}

if (cm2<=50 && cm2>=30 && cm1>=51 && cm1<=335 && cm3>=51 &&
cm3<=335)
{
  Right();
}

if (cm2<=50 && cm2>=30 && cm1>=51 && cm1<=335 && cm3<=50 &&
cm3>=30)
{
  Right();
}

```

```

}
```

```

    if (cm2>=51 && cm2<=335 && cm1>=51 && cm1<=335 && cm3<=50 &&
cm3>=30)
```

```

    {
        Right();
    }
```

```

    if (cm1<=50 && cm1>=30 && cm2>=51 && cm2<=335 && cm3>=51 &&
cm3<=335)
```

```

    {
        Left();
    }
```

```

    if (cm1<=50 && cm1>=30 && cm2<=50 && cm2>=30 && cm3>=51 &&
cm3<=335)
```

```

    {
        Left();
    }
```

```

    if (cm1<=50 && cm1>=30 && cm2<=50 && cm2>=30 && cm3<=50 &&
cm3>=30)
```

```

    {
        Stop();
        Back();
        Stop();
        Left();
    }
```

```

    if (cm1<=50 && cm1>=30 && cm2>=51 && cm2<=335 && cm3<=50 &&
cm3>=30)
```

```
{
  Stop();
  Back();
  Stop();
  Right();
}

if (cm1<=29 | cm2<=29 | cm3<=29 )
{
  Stop();
  Back();
  Stop();
  Ahead();
}

if (cm1<=29 && cm2<=29 && cm3<=29 )
{
  Stop();
  Back();
  Stop();
  Right();
}
Serial.print("Forward:"); Serial.print(cm2); Serial.print("cm");
Serial.print("Left:"); Serial.print(cm1); Serial.print("cm");
Serial.print("Right:"); Serial.print(cm3); Serial.println("cm");
scanSurroundings();
}
```

ДОДАТОК В

Код програми для технічної моделі складського робота у середовищі Wokwi

```
#include <Servo.h>

const int trigFront = 22, echoFront = 23;
const int trigBack = 24, echoBack = 25;
const int trigLeft = 26, echoLeft = 27;
const int trigRight = 28, echoRight = 29;
const int trigFL = 30, echoFL = 31;
const int trigFR = 32, echoFR = 33;
const int servoPin = 9;
const int trigServo = 34, echoServo = 35;
const int enA = 2;
const int in1 = 3;
const int in2 = 4;
const int enB = 5;
const int in3 = 6;
const int in4 = 7;
const int DIST_THRESHOLD = 20;

Servo servo;

void setup() {

  Serial.begin(9600);

  pinMode(trigFront, OUTPUT); pinMode(echoFront, INPUT);
```

```
pinMode(trigBack, OUTPUT); pinMode(echoBack, INPUT);
pinMode(trigLeft, OUTPUT); pinMode(echoLeft, INPUT);
pinMode(trigRight, OUTPUT); pinMode(echoRight, INPUT);
pinMode(trigFL, OUTPUT); pinMode(echoFL, INPUT);
pinMode(trigFR, OUTPUT); pinMode(echoFR, INPUT);
pinMode(trigServo, OUTPUT); pinMode(echoServo, INPUT);
```

```
pinMode(enA, OUTPUT); pinMode(in1, OUTPUT); pinMode(in2,
OUTPUT);
```

```
pinMode(enB, OUTPUT); pinMode(in3, OUTPUT); pinMode(in4, OUTPUT);
```

```
servo.attach(servoPin);
```

```
servo.write(90);
```

```
delay(500);
```

```
}
```

```
void stopAll() {
```

```
  analogWrite(enA, 0);
```

```
  analogWrite(enB, 0);
```

```
}
```

```
void forward() {
```

```
  digitalWrite(in1, HIGH); digitalWrite(in2, LOW);
```

```
  digitalWrite(in3, HIGH); digitalWrite(in4, LOW);
```

```
  analogWrite(enA, 200);
```

```
  analogWrite(enB, 200);
```

```
}
```

```
void backward() {
```

```
  digitalWrite(in1, LOW); digitalWrite(in2, HIGH);
```

```
digitalWrite(in3, LOW); digitalWrite(in4, HIGH);  
analogWrite(enA, 200);  
analogWrite(enB, 200);  
}
```

```
void turnRight() {  
    digitalWrite(in1, LOW); digitalWrite(in2, HIGH);  
    digitalWrite(in3, HIGH); digitalWrite(in4, LOW);  
    analogWrite(enA, 200);  
    analogWrite(enB, 200);  
}
```

```
void turnLeft() {  
    digitalWrite(in1, HIGH); digitalWrite(in2, LOW);  
    digitalWrite(in3, LOW); digitalWrite(in4, HIGH);  
    analogWrite(enA, 200);  
    analogWrite(enB, 200);  
}
```

```
long readDistance(int trigPin, int echoPin) {  
    digitalWrite(trigPin, LOW);  
    delayMicroseconds(5);  
    digitalWrite(trigPin, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trigPin, LOW);  
  
    long duration = pulseIn(echoPin, HIGH);  
    long distance = duration * 0.0343 / 2;  
    return distance;  
}
```

```
void loop() {  
  forward();  
  
  long distF = readDistance(trigFront, echoFront);  
  long distL = readDistance(trigLeft, echoLeft);  
  long distR = readDistance(trigRight, echoRight);  
  long distFL = readDistance(trigFL, echoFL);  
  long distFR = readDistance(trigFR, echoFR);  
  
  if (distF < DIST_THRESHOLD || distFL < DIST_THRESHOLD || distFR <  
DIST_THRESHOLD) {  
    stopAll();  
    backward();  
    delay(500);  
    turnLeft();  
    delay(300);  
    stopAll();  
  }  
  
  else if (distR < DIST_THRESHOLD) {  
    stopAll();  
    turnLeft();  
    delay(300);  
    stopAll();  
  }  
  
  else if (distL < DIST_THRESHOLD) {  
    stopAll();  
    turnRight();  
  }  
}
```

```
    delay(300);
    stopAll();
}

static int angle = 0;
servo.write(angle);
delay(50);
long distScan = readDistance(trigServo, echoServo);

if (angle >= 0 && angle <= 180 && distScan < DIST_THRESHOLD) {
    stopAll();
    turnRight();
    delay(300);
    stopAll();
}
angle += 30;
if (angle > 180) angle = 0;

if (random(1000) > 990) {
    stopAll();
    backward();
    delay(300);
    turnRight();
    delay(300);
    stopAll();
}

delay(100);
}
```

ДОДАТОК Г

Код головної форми (MyForm.h) віртуального макету системи у Visual Studio

```
#pragma comment(lib,"user32.lib")
#pragma comment(lib,"gdi32.lib")
#pragma comment(lib,"glu32.lib")
#pragma comment(lib,"opengl32.lib")

#include <windows.h>
#include <GL/gl.h>
#include <GL/glu.h>
#include <cmath>
#include "MyForm1.h"
#include <ctime>
#include <WinUser.h>

public ref class Order {
public:
    System::String^ Name;
    System::Collections::Generic::List<int>^ RackNumbers;
};

float zoom = 5.0f;
int currentRackInOrderIndex = -1;
int hspos = 50, vspos = 50;
int delayCounter = 0;
int selectedRackIndex = -1;
int currentRidePointIndex = -1;
```

```
enum BoxState {
    BOX_START,
    BOX_TO_POINT_1,
    BOX_TO_POINT_2,
    BOX_BACK_TO_START
}; BoxState boxState = BOX_START;
enum class RobotState {
    Idle,
    ToRidePoint1,
    ToRidePoint2,
    ToRidePoint3,
    currentRidePointPointReached,
    ToRackApproach,
    ToRack,
    CarryingToRidePoint1,
    CarryingToRidePoint2,
    CarryingToRidePoint3,
    ReturningToRidePoint1,
    ReturningToRidePoint2,
    ReturningToRidePoint3,
    CarryingToHome,
    WaitingForSpace,
    ReturningToRackApproach,
    ReturningToRack,
    GoingToWaitingPointRidePoint1,
    GoingToWaitingPointRidePoint2,
    GoingToWaitingPointRidePoint3,
    GoingToWaitingPointHome,
    OrderComplete
}; RobotState robotState = RobotState::Idle;
```

```

GLuint floorTextureID;
GLfloat box_Translate[] = { 0.4f, 0.65f, 0.0f };
GLfloat robot_Translate[] = { 6.0f, 0.0f, 1.0f };
GLfloat robotTarget[] = { 6.0f, 0.0f, 1.0f };
GLfloat robotHome[] = { 6.0f, 0.0f, 1.0f };
GLfloat rackPositions[25][3] = {
    { -0.5f, 0.0f, 0.0f }, { 0.85f, 0.0f, 0.0f }, { -1.85f, 0.0f, 0.0f },
    { -4.55f, 0.0f, 0.0f }, { -3.2f, 0.0f, 0.0f }, { -0.5f, 0.0f, 2.0f },
    { 0.85f, 0.0f, 2.0f }, { -1.85f, 0.0f, 2.0f }, { -4.55f, 0.0f, 2.0f },
    { -3.2f, 0.0f, 2.0f }, { -0.5f, 0.0f, -2.0f }, { 0.85f, 0.0f, -2.0f },
    { -1.85f, 0.0f, -2.0f }, { -4.55f, 0.0f, -2.0f }, { -3.2f, 0.0f, -2.0f },
    { -0.5f, 0.0f, -4.0f }, { 0.85f, 0.0f, -4.0f }, { -1.85f, 0.0f, -4.0f },
    { -4.55f, 0.0f, -4.0f }, { -3.2f, 0.0f, -4.0f }, { -0.5f, 0.0f, 4.0f },
    { 0.85f, 0.0f, 4.0f }, { -1.85f, 0.0f, 4.0f }, { -4.55f, 0.0f, 4.0f },
    { -3.2f, 0.0f, 4.0f }
};

```

```

bool spaceKeyPressed = false;
bool isMoving = false;
bool carryingRack = false;
bool returningRack = false;
bool waiting = false;
bool returningHome = false;
bool rackAttached = false;
bool goingHome = false;
bool currentRidePointPointReached[3];
const int totalRacks = 25;

```

```

struct RidePoint {
    float x;
    float z;
};

static float targetLineX = 2.0f;
static const RidePoint waitingPoint = { 3.0f, 1.0f };
static const RidePoint ridePoints[] = {
    { 3.0f, 1.0f }, { 2.0f, 1.0f }, { 2.0f, -1.0f }, { 2.0f, -3.0f }, { 2.0f, 3.0f },
    { 0.8f, 1.0f }, { 0.8f, -1.0f },
    { 0.8f, -3.0f }, { 0.8f, 3.0f }, { -0.5f, 1.0f }, { -0.5f, -1.0f }, { -0.5f, -3.0f },
},
    { -0.5f, 3.0f }, { -1.8f, 1.0f }, { -1.8f, -1.0f }, { -1.8f, -3.0f }, { -1.8f, 3.0f },
},
    { -3.1f, 1.0f }, { -3.1f, -1.0f }, { -3.1f, -3.0f }, { -3.1f, 3.0f }, { -4.4f, 1.0f },
},
    { -4.4f, -1.0f }, { -4.4f, -3.0f }, { -4.4f, 3.0f }
};

static const int rackRoutes[totalRacks][4] = {

    { 0, 1, 9, -1 },
    { 0, 1, 5, -1 },
    { 0, 1, 13, -1 },
    { 0, 1, 21, -1 },
    { 0, 1, 17, -1 },

    { 0, 1, 9, -1 },
    { 0, 1, 5, -1 },
    { 0, 1, 13, -1 },
    { 0, 1, 21, -1 },

```

```

    { 0, 1, 17, -1 },

    { 0, 2, 10, -1 },
    { 0, 2, 6, -1 },
    { 0, 2, 14, -1 },
    { 0, 2, 22, -1 },
    { 0, 2, 18, -1 },

    { 0, 3, 11, -1 },
    { 0, 3, 7, -1 },
    { 0, 3, 15, -1 },
    { 0, 3, 23, -1 },
    { 0, 3, 19, -1 },

    { 0, 4, 12, -1 },
    { 0, 4, 8, -1 },
    { 0, 4, 16, -1 },
    { 0, 4, 24, -1 },
    { 0, 4, 20, -1 },
};

const int totalRidePoints = sizeof(ridePoints) / sizeof(RidePoint);
bool MoveTo(float& current, float target, float speed) {
    if (abs(current - target) < speed) {
        current = target;
        return true;
    }
    else if (current < target) {
        current += speed;
    }
}

```

```
else {
    current -= speed;
}
return false;
}

namespace diplom {
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    using namespace Microsoft::VisualBasic;

    public ref class MyForm : public System::Windows::Forms::Form
    {
    public:
        MyForm(void)
        {
            InitializeComponent();
        }

    protected:
        ~MyForm()
        {
            if (components)
            {
                delete components;
            }
        }
    }
}
```

```

    }
private:
    Order^ currentActiveOrder;
    HDC hdc;
    HGLRC hglrc;
    Timer^ renderTimer;
    Timer^ animationTimer;
    Timer^ boxTimer;
    VScrollBar^ vScrollBar1;
    HScrollBar^ hScrollBar1;
    System::ComponentModel::Container^ components;

#pragma region Windows Form Designer generated code
    void InitializeComponent(void)
    {
        this->vScrollBar1 = (gcnew
System::Windows::Forms::VScrollBar());
        this->hScrollBar1 = (gcnew
System::Windows::Forms::HScrollBar());
        this->SuspendLayout();
        this->vScrollBar1->Location = System::Drawing::Point(10,
21);

        this->vScrollBar1->Name = L"vScrollBar1";
        this->vScrollBar1->Size = System::Drawing::Size(20, 570);
        this->vScrollBar1->TabIndex = 0;
        this->vScrollBar1->Value = 50;
        this->vScrollBar1->Scroll += gcnew
System::Windows::Forms::ScrollEventHandler(this,
&MyForm::vScrollBar1_Scroll);

```

```

        this->hScrollBar1->Location = System::Drawing::Point(47,
10);

        this->hScrollBar1->Name = L"hScrollBar1";
        this->hScrollBar1->Size = System::Drawing::Size(1196,
22);

        this->hScrollBar1->TabIndex = 1;
        this->hScrollBar1->Value = 50;
        this->hScrollBar1->Scroll += gcnew

System::Windows::Forms::ScrollEventHandler(this,
&MyForm::hScrollBar1_Scroll);

        this->AutoScaleDimensions = System::Drawing::SizeF(8,
16);

        this->AutoScaleMode =

System::Windows::Forms::AutoScaleMode::Font;

        this->ClientSize = System::Drawing::Size(1252, 600);
        this->Controls->Add(this->hScrollBar1);
        this->Controls->Add(this->vScrollBar1);
        this->Name = L"MyForm";
        this->Text = L"Storage";
        this->FormClosing += gcnew

System::Windows::Forms::FormClosingEventHandler(this,
&MyForm::MyForm_Closing);

        this->Load += gcnew System::EventHandler(this,
&MyForm::MyForm_Load);

        this->Resize += gcnew System::EventHandler(this,
&MyForm::MyForm_Resize);

        this->MouseWheel += gcnew MouseEventArgs(this,
&MyForm::MyForm_MouseWheel);

        animationTimer = gcnew Timer();
        animationTimer->Interval = 30;

```

```

        animationTimer->Tick += gcnew EventHandler(this,
&MyForm::AnimationTimer_Tick);

        boxTimer = gcnew Timer();
        boxTimer->Interval = 30;
        boxTimer->Tick += gcnew EventHandler(this,
&MyForm::BoxTimer_Tick);
        this->KeyPreview = true;
        this->KeyDown += gcnew KeyEventHandler(this,
&MyForm::MyForm_KeyDown);
        this->ResumeLayout(false);
    }
    void vScrollBar1_Scroll(System::Object^ sender,
System::Windows::Forms::ScrollEventArgs^ e) {
        vspos = e->NewValue;
        this->RenderScene(nullptr, nullptr);
    }
    void hScrollBar1_Scroll(System::Object^ sender,
System::Windows::Forms::ScrollEventArgs^ e) {
        hspos = e->NewValue;
        this->RenderScene(nullptr, nullptr);
    }
    void MyForm_MouseWheel(System::Object^ sender,
System::Windows::Forms::MouseEventArgs^ e)
    {
        if (e->Delta > 0)
            zoom -= 0.5f;
        else
            zoom += 0.5f;
        if (zoom < 1.0f) zoom = 1.0f;
    }

```

```

        if (zoom > 50.0f) zoom = 50.0f;
        this->RenderScene(nullptr, nullptr);
    }
void InitializeOpenGL() {
    hdc = GetDC((HWND)this->Handle.ToPointer());
    PIXELFORMATDESCRIPTOR pfd;
    ZeroMemory(&pfd, sizeof(pfd));
    pfd.nSize = sizeof(pfd);
    pfd.nVersion = 1;
    pfd.dwFlags = PFD_DRAW_TO_WINDOW |
PFD_SUPPORT_OPENGL | PFD_DOUBLEBUFFER;
    pfd.iPixelFormat = PFD_TYPE_RGBA;
    pfd.cColorBits = 32;
    pfd.cDepthBits = 24;
    pfd.iLayerType = PFD_MAIN_PLANE;
    int pixelFormat = ChoosePixelFormat(hdc, &pfd);
    SetPixelFormat(hdc, pixelFormat, &pfd);
    hglrc = wglCreateContext(hdc);
    wglMakeCurrent(hdc, hglrc);
    glEnable(GL_DEPTH_TEST);
    renderTimer = gcnew Timer();
    renderTimer->Interval = 16;
    renderTimer->Tick += gcnew EventHandler(this,
&MyForm::RenderScene);
    renderTimer->Start();
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glEnable(GL_COLOR_MATERIAL);
    glColorMaterial(GL_FRONT,
GL_AMBIENT_AND_DIFFUSE);

```

```

    GLfloat lightAmbient[] = { 0.2f, 0.2f, 0.2f, 1.0f };
    GLfloat lightDiffuse[] = { 0.8f, 0.8f, 0.8f, 1.0f };
    GLfloat lightSpecular[] = { 1.0f, 1.0f, 1.0f, 1.0f };
    GLfloat lightPosition[] = { 5.0f, 5.0f, 5.0f, 1.0f };
    glLightfv(GL_LIGHT0, GL_AMBIENT, lightAmbient);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, lightDiffuse);
    glLightfv(GL_LIGHT0, GL_SPECULAR, lightSpecular);
    glLightfv(GL_LIGHT0, GL_POSITION, lightPosition);
    glEnable(GL_NORMALIZE);
}
void SetupProjection(int width, int height)
{
    if (height == 0) height = 1;
    float aspect = (float)width / (float)height;
    glViewport(0, 0, width, height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(50.0, aspect, 0.1, 50.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}
void RenderScene(Object^ sender, EventArgs^ e) {
    glClear(GL_COLOR_BUFFER_BIT |
GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    float cameraX = (float)hspos / 5.0f - 5.0f;
    float cameraY = (float)vspos / 5.0f - 5.0f;
    gluLookAt(cameraX, cameraY, zoom,
        0.0, 0.0, 0.0,
        0.0, 1.0, 0.0);
}

```

```

Floor();
Conveyor();
Table();
Chair();
Robot();
for (int i = 0; i < totalRacks; i++) {
    if (!(carryingRack && i == selectedRackIndex)) {
        glPushMatrix();
        glTranslatef(rackPositions[i][0] - 0.5f, 0.0f,
rackPositions[i][2]);

        DrawRack(i);
        glPopMatrix();
    }
}
glFinish();
SwapBuffers(hdc);
}
void Cube(float size)
{
    float half = size / 2.0f;
    glBegin(GL_QUADS);
    glNormal3f(0.0f, 1.0f, 0.0f);
    glVertex3f(-half, half, -half);
    glVertex3f(half, half, -half);
    glVertex3f(half, half, half);
    glVertex3f(-half, half, half);

    glNormal3f(0.0f, -1.0f, 0.0f);
    glVertex3f(-half, -half, -half);
    glVertex3f(half, -half, -half);

```

```
glVertex3f(half, -half, half);
glVertex3f(-half, -half, half);
```

```
glNormal3f(0.0f, 0.0f, 1.0f);
glVertex3f(-half, -half, half);
glVertex3f(half, -half, half);
glVertex3f(half, half, half);
glVertex3f(-half, half, half);
```

```
glNormal3f(0.0f, 0.0f, -1.0f);
glVertex3f(-half, -half, -half);
glVertex3f(half, -half, -half);
glVertex3f(half, half, -half);
glVertex3f(-half, half, -half);
```

```
glNormal3f(-1.0f, 0.0f, 0.0f);
glVertex3f(-half, -half, -half);
glVertex3f(-half, half, -half);
glVertex3f(-half, half, half);
glVertex3f(-half, -half, half);
```

```
glNormal3f(1.0f, 0.0f, 0.0f);
glVertex3f(half, -half, -half);
glVertex3f(half, half, -half);
glVertex3f(half, half, half);
glVertex3f(half, -half, half);
glEnd();
```

```
}
```

```
void Celinder(float baseRadius, float topRadius, float height, int
slices, int stacks) {
```

```

        GLUquadric* quadric = gluNewQuadric();
        gluQuadricTexture(quadric, GL_TRUE);
        glPushMatrix();
        glTranslatef(0.0f, 0.0f, 0.0f);
        gluDisk(quadric, 0.0f, baseRadius, slices, 1);
        glPopMatrix();
        glPushMatrix();
        glTranslatef(0.0f, 0.0f, height);
        gluDisk(quadric, 0.0f, topRadius, slices, 1);
        glPopMatrix();
        gluCylinder(quadric, baseRadius, topRadius, height, slices,
stacks);

        gluDeleteQuadric(quadric);
    }
void Floor() {
    glPushMatrix();
    glTranslatef(0.0f, -0.2f, 0.0f);
    glScalef(15.0f, 0.2f, 12.0f);
    glColor3f(0.3f, 0.3f, 0.3f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
}
void Conveyor() {
    glPushMatrix();
    glTranslatef(7.0f, 0.0f, 0.0f);

```

```
glPushMatrix();
glTranslatef(0.25f, 0.25f, -4.5f);
glScalef(0.05f, 0.5f, 0.05f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(-0.25f, 0.25f, -4.5f);
glScalef(0.05f, 0.5f, 0.05f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.25f, 0.25f, 4.5f);
glScalef(0.05f, 0.5f, 0.05f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
```

```
glPushMatrix();
glTranslatef(-0.25f, 0.25f, 4.5f);
glScalef(0.05f, 0.5f, 0.05f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.48f, 0.0f);
glScalef(0.50f, 0.02f, 9.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.375f, 0.0f);
glScalef(0.5f, 0.02f, 9.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(-0.15, 0.425f, 4.5f);
glRotatef(90.0f, 0.0f, 1.0f, 0.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.06f, 0.06f, 0.4f, 32, 32);
glPopMatrix();
```

```

    glPushMatrix();
    glTranslatef(-0.15, 0.425f, -4.5f);
    glRotatef(90.0f, 0.0f, 1.0f, 0.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.06f, 0.06f, 0.4f, 32, 32);
    glPopMatrix();
    glPopMatrix();
}
void Table() {
    glPushMatrix();
    glTranslatef(6.0f, 0.0f, -1.5f);
    glPushMatrix();
    glTranslatef(0.0f, 0.5f, 0.0f);
    glScalef(1.5f, 0.05f, 0.75f);
    glColor3f(0.5f, 0.5f, 0.5f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.725f, 0.25f, 0.36f);
    glScalef(0.05f, 0.5f, 0.05f);
    glColor3f(0.0f, 1.0f, 0.0f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);

```

```
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.725f, 0.25f, -0.36f);
glScalef(0.05f, 0.5f, 0.05f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(-0.725f, 0.25f, 0.36f);
glScalef(0.05f, 0.5f, 0.05f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(-0.725f, 0.25f, -0.36f);
glScalef(0.05f, 0.5f, 0.05f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
```

```
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(-0.4f, 0.55f, 0.2f);
glScalef(0.35f, 0.05f, 0.2f);
glColor3f(0.0f, 0.0f, 0.0f);
Cube(1.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(-0.4f, 0.55f, -0.2f);
glScalef(0.15f, 0.05f, 0.15f);
glColor3f(0.0f, 0.0f, 0.0f);
Cube(1.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(-0.4f, 0.6f, -0.2f);
glScalef(0.05f, 0.1f, 0.05f);
glColor3f(0.0f, 0.0f, 0.0f);
Cube(1.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(-0.4f, 0.75f, -0.2f);
glScalef(0.35f, 0.25f, 0.05f);
glColor3f(0.0f, 0.0f, 0.0f);
Cube(1.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(-0.4f, 0.75f, -0.198f);
glScalef(0.335f, 0.235f, 0.05f);
glColor3f(1.0f, 1.0f, 1.0f);
```

```

Cube(1.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(box_Translate[0], box_Translate[1],
box_Translate[2]);

glScalef(0.4f, 0.3f, 0.5f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPopMatrix();
}

void Chair() {
    glPushMatrix();
    glTranslatef(6.0f, 0.0f, -1.0f);
    glPushMatrix();
    glTranslatef(0.0f, 0.3f, 0.0f);
    glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
    glColor3f(0.5f, 0.5f, 0.6f);
    Celinder(0.2f, 0.2f, 0.1f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Celinder(0.2f, 0.2f, 0.1f, 8, 8);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.15f, 0.0f);

```

```
glScalef(0.05f, 0.3f, 0.05f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.01f, 0.0f);
glScalef(0.25f, 0.05f, 0.05f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.01f, 0.0f);
glScalef(0.05f, 0.05f, 0.25f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, -0.05f, 0.15f);
```

```

        glScalef(0.05f, 0.05f, 0.05f);
        glColor3f(0.0f, 0.0f, 0.0f);
        Cube(1.0f);
        glPopMatrix();
        glPushMatrix();
        glTranslatef(0.0f, -0.05f, -0.15f);
        glScalef(0.05f, 0.05f, 0.05f);
        glColor3f(0.0f, 0.0f, 0.0f);
        Cube(1.0f);
        glPopMatrix();
        glPushMatrix();
        glTranslatef(0.15f, -0.05f, 0.0f);
        glScalef(0.05f, 0.05f, 0.05f);
        glColor3f(0.0f, 0.0f, 0.0f);
        Cube(1.0f);
        glPopMatrix();
        glPushMatrix();
        glTranslatef(-0.15f, -0.05f, 0.0f);
        glScalef(0.05f, 0.05f, 0.05f);
        glColor3f(0.0f, 0.0f, 0.0f);
        Cube(1.0f);
        glPopMatrix();
        glPopMatrix();
    }

    void Rack_1() {
        glPushMatrix();
        glPushMatrix();
        glTranslatef(0.0f, 0.0f, -0.02f);
        glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
        glColor3f(0.0f, 0.0f, 0.0f);

```

```
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
```

```
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
```

```
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
```

```
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 1.1f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
```

```
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}
void Rack_2() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(1.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
```

```
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
```

```
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
```

```
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
```

```

    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.3f, 1.1f, 0.15f);
    glScalef(0.3f, 0.2f, 0.3f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}

void Rack_3() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);

```

```
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
```

```
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
```

```
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
```

```
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 1.1f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPopMatrix();
}
```

```
void Rack_4() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(1.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(1.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
}
```

```
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
```

```
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
```

```
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
```

```

    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.3f, 1.1f, 0.15f);
    glScalef(0.3f, 0.2f, 0.3f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}

void Rack_5() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
}

```

```
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
```

```
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
```

```
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
```

```

    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.75f, 1.15f, 0.15f);
    glScalef(0.4f, 0.3f, 0.4f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.3f, 1.1f, 0.15f);
    glScalef(0.3f, 0.2f, 0.3f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}

void Rack_6() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);

```

```
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
```

```
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
```

```
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
```

```
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 1.1f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
```

```

    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}
void Rack_7() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(1.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
}

```

```
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
```

```
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
```

```
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
```

```

    glScalef(0.4f, 0.3f, 0.4f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.3f, 1.1f, 0.15f);
    glScalef(0.3f, 0.2f, 0.3f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}

void Rack_8() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();

```

```
glTranslatef(0.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
```

```
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
```

```
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
```

```
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 1.1f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
```

```
        glPopMatrix();
        glPopMatrix();
    }
void Rack_9() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(1.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(1.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
```

```
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
```

```
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
```

```
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
```

```

    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.3f, 1.1f, 0.15f);
    glScalef(0.3f, 0.2f, 0.3f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}

void Rack_10() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
}

```

```
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
```

```
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
```

```
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
```

```

    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.75f, 1.15f, 0.15f);
    glScalef(0.4f, 0.3f, 0.4f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.3f, 1.1f, 0.15f);
    glScalef(0.3f, 0.2f, 0.3f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}
void Rack_11() {
    glPushMatrix();

```

```
glPushMatrix();
glTranslatef(0.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
```

```
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
```

```
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
```

```
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 1.1f, 0.15f);
```

```
    glScalef(0.3f, 0.2f, 0.3f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}
void Rack_12() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(1.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
```

```
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
```

```
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
```

```
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
```

```

    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.75f, 1.15f, 0.15f);
    glScalef(0.4f, 0.3f, 0.4f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.3f, 1.1f, 0.15f);
    glScalef(0.3f, 0.2f, 0.3f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}

void Rack_13() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);

```

```
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
```

```
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
```

```
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
```

```
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 1.1f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
```

```
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPopMatrix();
}
void Rack_14() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(1.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(1.0f, 0.0f, 0.28f);
```

```
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
```

```
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
```

```
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
```

```

    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.3f, 1.1f, 0.15f);
    glScalef(0.3f, 0.2f, 0.3f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}

void Rack_15() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);

```

```
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
```

```
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
```

```
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
```

```
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.75f, 1.15f, 0.15f);
    glScalef(0.4f, 0.3f, 0.4f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.3f, 1.1f, 0.15f);
    glScalef(0.3f, 0.2f, 0.3f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}
void Rack_16() {
```

```
glPushMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Celinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Celinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Celinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Celinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
```

```
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
```

```
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
```

```
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
```

```

    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.3f, 1.1f, 0.15f);
    glScalef(0.3f, 0.2f, 0.3f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}

void Rack_17() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
}

```

```
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
```

```
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
```

```
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
```

```

    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.75f, 1.15f, 0.15f);
    glScalef(0.4f, 0.3f, 0.4f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.3f, 1.1f, 0.15f);
    glScalef(0.3f, 0.2f, 0.3f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}

void Rack_18() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);

```

```
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
```

```
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
```

```
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
```

```
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 1.1f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
```

```
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPopMatrix();
}
void Rack_19() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(1.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
```

```
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
```

```
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
```

```
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
```

```

    glScalef(0.4f, 0.3f, 0.4f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.3f, 1.1f, 0.15f);
    glScalef(0.3f, 0.2f, 0.3f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}

void Rack_20() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
}

```

```
glPushMatrix();
glTranslatef(0.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
```

```
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
```

```
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
```

```
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 1.1f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
```

```
        glPopMatrix();
        glPopMatrix();
    }
void Rack_21() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(1.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(1.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
```

```
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
```

```
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
```

```
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
```

```

Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 1.1f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPopMatrix();
}
void Rack_22() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);

```

```
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
```

```
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
```

```
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
```

```

    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.75f, 1.15f, 0.15f);
    glScalef(0.4f, 0.3f, 0.4f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.3f, 1.1f, 0.15f);
    glScalef(0.3f, 0.2f, 0.3f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}

void Rack_23() {
    glPushMatrix();
    glPushMatrix();

```

```
glTranslatef(0.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
```

```
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
```

```
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
```

```
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 1.1f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
```

```
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}
void Rack_24() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, 0.28f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(1.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);
```

```
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
```

```
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
```

```
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
```

```

    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.75f, 1.15f, 0.15f);
    glScalef(0.4f, 0.3f, 0.4f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.3f, 1.1f, 0.15f);
    glScalef(0.3f, 0.2f, 0.3f);
    glColor3f(0.76f, 0.60f, 0.42f);
    Cube(1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Cube(1.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPopMatrix();
}

void Rack_25() {
    glPushMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.0f, -0.02f);
    glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
    glColor3f(0.0f, 0.0f, 0.0f);
    Celinder(0.1f, 0.1f, 0.03f, 32, 32);

```

```
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, -0.02f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.0f, 0.28f);
glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Cylinder(0.1f, 0.1f, 0.03f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
```

```
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.0f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(1.0f, 0.7f, 0.3f);
glScalef(0.1f, 1.5f, 0.1f);
glColor3f(0.0f, 1.0f, 0.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
```

```
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 1.0f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.5f, 0.3f, 0.15f);
glScalef(1.15f, 0.05f, 0.45f);
glColor3f(0.5f, 0.5f, 0.5f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 0.45f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
```

```
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 0.4f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.75f, 1.15f, 0.15f);
glScalef(0.4f, 0.3f, 0.4f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.3f, 1.1f, 0.15f);
glScalef(0.3f, 0.2f, 0.3f);
glColor3f(0.76f, 0.60f, 0.42f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
```

```
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPopMatrix();
}
void Robot() {
    glPushMatrix();
    glTranslatef(robot_Translate[0], robot_Translate[1],
robot_Translate[2]);

    glPushMatrix();
    glTranslatef(0.0f, 0.09f, 0.0f);
    glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
    glColor3f(0.5f, 0.4f, 0.5f);
    Celinder(0.2f, 0.2f, 0.115f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Celinder(0.2f, 0.2f, 0.115f, 8, 8);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0.0f, 0.15f, 0.0f);
    glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
    glColor3f(1.0f, 1.0f, 1.0f);
    Celinder(0.185f, 0.2f, 0.05f, 32, 32);
    glColor3f(0.0f, 0.0f, 0.0f);
    glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
    Celinder(0.185f, 0.2f, 0.075f, 8, 8);
    glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
    glPopMatrix();
```

```
glPushMatrix();
glTranslatef(0.0f, 0.16f, 0.0f);
glScalef(0.15f, 0.1f, 0.15f);
glColor3f(0.0f, 0.0f, 1.0f);
Cube(1.0f);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cube(1.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(0.04f, 0.23f, 0.0f);
glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glColor3f(0.5f, 0.4f, 0.5f);
Cylinder(0.025f, 0.025f, 0.175f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cylinder(0.025f, 0.025f, 0.175f, 8, 8);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
glPushMatrix();
glTranslatef(-0.04f, 0.23f, 0.0f);
glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
glColor3f(0.5f, 0.4f, 0.5f);
Cylinder(0.025f, 0.025f, 0.175f, 32, 32);
glColor3f(0.0f, 0.0f, 0.0f);
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
Cylinder(0.025f, 0.025f, 0.175f, 8, 8);
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
glPopMatrix();
```

```
glPushMatrix();
glTranslatef(-0.09f, 0.0f, 0.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Celinder(0.05f, 0.05f, 0.05f, 32, 32);
glPopMatrix();
glPushMatrix();
glTranslatef(0.09f, 0.0f, 0.0f);
glColor3f(0.0f, 0.0f, 0.0f);
Celinder(0.05f, 0.05f, 0.05f, 32, 32);
glPopMatrix();
glPushMatrix();
glTranslatef(0.1f, -0.04f, 0.1f);
glColor3f(0.0f, 0.0f, 0.0f);
Celinder(0.015f, 0.015f, 0.05f, 32, 32);
glPopMatrix();
glPushMatrix();
glTranslatef(0.1f, -0.04f, -0.1f);
glColor3f(0.0f, 0.0f, 0.0f);
Celinder(0.015f, 0.015f, 0.05f, 32, 32);
glPopMatrix();
glPushMatrix();
glTranslatef(-0.1f, -0.04f, 0.1f);
glColor3f(0.0f, 0.0f, 0.0f);
Celinder(0.015f, 0.015f, 0.05f, 32, 32);
glPopMatrix();
glPushMatrix();
glTranslatef(-0.1f, -0.04f, -0.1f);
glColor3f(0.0f, 0.0f, 0.0f);
Celinder(0.015f, 0.015f, 0.05f, 32, 32);
glPopMatrix();
```

```
    if (carryingRack && selectedRackIndex >= 0) {
        glPushMatrix();
        glTranslatef(-0.5f, 0.0f, -0.15f);
        DrawRack(selectedRackIndex);
        glPopMatrix();
    }
    glPopMatrix();
}

void DrawRack(int index) {
    switch (index) {
        case 0: Rack_1(); break;
        case 1: Rack_2(); break;
        case 2: Rack_3(); break;
        case 3: Rack_4(); break;
        case 4: Rack_5(); break;
        case 5: Rack_6(); break;
        case 6: Rack_7(); break;
        case 7: Rack_8(); break;
        case 8: Rack_9(); break;
        case 9: Rack_10(); break;
        case 10: Rack_11(); break;
        case 11: Rack_12(); break;
        case 12: Rack_13(); break;
        case 13: Rack_14(); break;
        case 14: Rack_15(); break;
        case 15: Rack_16(); break;
        case 16: Rack_17(); break;
        case 17: Rack_18(); break;
        case 18: Rack_19(); break;
        case 19: Rack_20(); break;
    }
}
```

```

        case 20: Rack_21(); break;
        case 21: Rack_22(); break;
        case 22: Rack_23(); break;
        case 23: Rack_24(); break;
        case 24: Rack_25(); break;
        default: break;
    }
}
void StartRobotAnimation(int rackIndex)
{
    selectedRackIndex = rackIndex;
    isMoving = true;
    carryingRack = false;
    rackAttached = false;
    waiting = false;
    returningRack = false;
    goingHome = false;
    spaceKeyPressed = false;
    robotState = RobotState::ToRidePoint1;
    currentRidePointIndex = 0;
    animationTimer->Start();
}
void AnimationTimer_Tick(System::Object^ sender,
System::EventArgs^ e)
{
    if (!isMoving && robotState !=
RobotState::WaitingForSpace) {
        if (robotState == RobotState::Idle &&
currentActiveOrder == nullptr) {
            if (boxState == BOX_START) {

```

```

        boxTimer->Start();
    }
}
this->RenderScene(nullptr, nullptr);
return;
}
float speed = 0.05f;
switch (robotState) {
case RobotState::Idle:
    isMoving = false;
    break;
case RobotState::ToRidePoint1: {
    if (selectedRackIndex == -1 || selectedRackIndex >=
totalRacks) {
        robotState = RobotState::Idle;
        isMoving = false;
        MessageBox::Show("Помилковий номер
стелажа в замовленні.", "Помилка");
        currentActiveOrder = nullptr;
        currentRackInOrderIndex = -1;
        break;
    }
    int ridePointIndex =
rackRoutes[selectedRackIndex][0];
    if (MoveTo(robot_Translate[0],
ridePoints[ridePointIndex].x, speed) &&
        MoveTo(robot_Translate[2],
ridePoints[ridePointIndex].z, speed)) {

```

```

        robotState =
(rackRoutes[selectedRackIndex][1] != -1) ? RobotState::ToRidePoint2 :
RobotState::ToRackApproach;
        currentRidePointPointReached[0] = true;
        currentRidePointIndex = 1;
    }
    break;
}
case RobotState::ToRidePoint2: {
    int ridePointIndex =
rackRoutes[selectedRackIndex][1];
        if (MoveTo(robot_Translate[0],
ridePoints[ridePointIndex].x, speed) &&
            MoveTo(robot_Translate[2],
ridePoints[ridePointIndex].z, speed)) {
            robotState =
(rackRoutes[selectedRackIndex][2] != -1) ? RobotState::ToRidePoint3 :
RobotState::ToRackApproach;
            currentRidePointPointReached[1] = true;
            currentRidePointIndex = 2;
        }
        break;
}
case RobotState::ToRidePoint3: {
    int ridePointIndex =
rackRoutes[selectedRackIndex][2];
        if (MoveTo(robot_Translate[0],
ridePoints[ridePointIndex].x, speed) &&
            MoveTo(robot_Translate[2],
ridePoints[ridePointIndex].z, speed)) {

```

```

        robotState = RobotState::ToRackApproach;
        currentRidePointReached[2] = true;
        currentRidePointIndex = 3;
    }
    break;
}
case RobotState::ToRackApproach: {
    if (MoveTo(robot_Translate[0],
rackPositions[selectedRackIndex][0], speed) &&
        MoveTo(robot_Translate[2],
rackPositions[selectedRackIndex][2], speed)) {
        robotState = RobotState::ToRack;
    }
    break;
}
case RobotState::ToRack: {
    carryingRack = true;
    rackAttached = true;
    int lastRidePointIndex = -1;
    for (int i = 3; i >= 0; --i) {
        if (rackRoutes[selectedRackIndex][i] != -1) {
            lastRidePointIndex = i;
            break;
        }
    }
    if (lastRidePointIndex == 2) {
        robotState =
RobotState::CarryingToRidePoint3;
        currentRidePointIndex = 2;
    }
}

```

```

else if (lastRidePointIndex == 1) {
    robotState =
RobotState::CarryingToRidePoint2;
    currentRidePointIndex = 1;
}
else if (lastRidePointIndex == 0) {
    robotState =
RobotState::CarryingToRidePoint1;
    currentRidePointIndex = 0;
}
else {
    robotState = RobotState::CarryingToHome;
}
break;
}
case RobotState::CarryingToRidePoint3: {
    int ridePointIndex =
rackRoutes[selectedRackIndex][2];
    if (MoveTo(robot_Translate[0],
ridePoints[ridePointIndex].x, speed) &&
        MoveTo(robot_Translate[2],
ridePoints[ridePointIndex].z, speed)) {
        robotState =
RobotState::CarryingToRidePoint2;
        currentRidePointIndex = 1;
    }
    break;
}
case RobotState::CarryingToRidePoint2: {

```

```

        int ridePointIndex =
rackRoutes[selectedRackIndex][1];
        if (MoveTo(robot_Translate[0],
ridePoints[ridePointIndex].x, speed) &&
            MoveTo(robot_Translate[2],
ridePoints[ridePointIndex].z, speed)) {
            robotState =
RobotState::CarryingToRidePoint1;
            currentRidePointIndex = 0;
        }
        break;
    }
    case RobotState::CarryingToRidePoint1: {
        int ridePointIndex =
rackRoutes[selectedRackIndex][0];
        if (MoveTo(robot_Translate[0],
ridePoints[ridePointIndex].x, speed) &&
            MoveTo(robot_Translate[2],
ridePoints[ridePointIndex].z, speed)) {
            robotState = RobotState::CarryingToHome;
        }
        break;
    }
    case RobotState::CarryingToHome: {
        if (MoveTo(robot_Translate[0], robotHome[0],
speed) &&
            MoveTo(robot_Translate[2], robotHome[2],
speed)) {
            robotState = RobotState::WaitingForSpace;
            carryingRack = true;

```

```

    }
    break;
}
case RobotState::WaitingForSpace: {
    if (spaceKeyPressed) {
        spaceKeyPressed = false;
        for (int i = 0; i < 3; ++i) {
            currentRidePointPointReached[i] = false;
        }
        int firstRidePointIndex = -1;
        for (int i = 0; i < 3; ++i) {
            if (rackRoutes[selectedRackIndex][i] != -
1) {
                firstRidePointIndex = i;
                break;
            }
        }
        if (firstRidePointIndex == 0) {
            robotState =
RobotState::ReturningToRidePoint1;
            currentRidePointIndex = 0;
        }
        else if (firstRidePointIndex == 1) {
            robotState =
RobotState::ReturningToRidePoint2;
            currentRidePointIndex = 1;
        }
        else if (firstRidePointIndex == 2) {
            robotState =
RobotState::ReturningToRidePoint3;

```

```

        currentRidePointIndex = 2;
    }
    else {
        robotState =
RobotState::ReturningToRackApproach;
    }
}
this->RenderScene(nullptr, nullptr);
return;
}
case RobotState::ReturningToRidePoint1: {
    int ridePointIndex =
rackRoutes[selectedRackIndex][0];
    if (MoveTo(robot_Translate[0],
ridePoints[ridePointIndex].x, speed) &&
        MoveTo(robot_Translate[2],
ridePoints[ridePointIndex].z, speed)) {
        robotState =
(rackRoutes[selectedRackIndex][1] != -1) ? RobotState::ReturningToRidePoint2 :
RobotState::ReturningToRackApproach;
        currentRidePointIndex = 1;
    }
    break;
}
case RobotState::ReturningToRidePoint2: {
    int ridePointIndex =
rackRoutes[selectedRackIndex][1];
    if (MoveTo(robot_Translate[0],
ridePoints[ridePointIndex].x, speed) &&

```

```

        MoveTo(robot_Translate[2],
ridePoints[ridePointIndex].z, speed)) {
            robotState =
(rackRoutes[selectedRackIndex][2] != -1) ? RobotState::ReturningToRidePoint3 :
RobotState::ReturningToRackApproach;
            currentRidePointIndex = 2;
        }
        break;
    }
    case RobotState::ReturningToRidePoint3: {
        int ridePointIndex =
rackRoutes[selectedRackIndex][2];
        if (MoveTo(robot_Translate[0],
ridePoints[ridePointIndex].x, speed) &&
            MoveTo(robot_Translate[2],
ridePoints[ridePointIndex].z, speed)) {
            robotState =
RobotState::ReturningToRackApproach;
            currentRidePointIndex = 3;
        }
        break;
    }
    case RobotState::ReturningToRackApproach: {
        if (MoveTo(robot_Translate[0],
rackPositions[selectedRackIndex][0], speed) &&
            MoveTo(robot_Translate[2],
rackPositions[selectedRackIndex][2], speed)) {
            robotState = RobotState::ReturningToRack;
        }
        break;
    }

```

```

}
case RobotState::ReturningToRack: {
    carryingRack = false;
    rackAttached = false;

    int lastRidePointIndex = -1;
    for (int i = 3; i >= 0; --i) {
        if (rackRoutes[selectedRackIndex][i] != -1) {
            lastRidePointIndex = i;
            break;
        }
    }
    if (lastRidePointIndex == 2) {
        robotState =
RobotState::GoingToWaitingPointRidePoint3;
        currentRidePointIndex = 2;
    }
    else if (lastRidePointIndex == 1) {
        robotState =
RobotState::GoingToWaitingPointRidePoint2;
        currentRidePointIndex = 1;
    }
    else if (lastRidePointIndex == 0) {
        robotState =
RobotState::GoingToWaitingPointRidePoint1;
        currentRidePointIndex = 0;
    }
    else {
        robotState =
RobotState::GoingToWaitingPointHome;

```

```

        }
        break;
    }
    case RobotState::GoingToWaitingPointRidePoint3: {
        int ridePointIndex =
rackRoutes[selectedRackIndex][2];
        if (MoveTo(robot_Translate[0],
ridePoints[ridePointIndex].x, speed) &&
            MoveTo(robot_Translate[2],
ridePoints[ridePointIndex].z, speed)) {
            robotState =
RobotState::GoingToWaitingPointRidePoint2;
            currentRidePointIndex = 1;
        }
        break;
    }
    case RobotState::GoingToWaitingPointRidePoint2: {
        int ridePointIndex =
rackRoutes[selectedRackIndex][1];
        if (MoveTo(robot_Translate[0],
ridePoints[ridePointIndex].x, speed) &&
            MoveTo(robot_Translate[2],
ridePoints[ridePointIndex].z, speed)) {
            robotState =
RobotState::GoingToWaitingPointRidePoint1;
            currentRidePointIndex = 0;
        }
        break;
    }
    case RobotState::GoingToWaitingPointRidePoint1: {

```

```

        int ridePointIndex =
rackRoutes[selectedRackIndex][0];
        if (MoveTo(robot_Translate[0],
ridePoints[ridePointIndex].x, speed) &&
            MoveTo(robot_Translate[2],
ridePoints[ridePointIndex].z, speed)) {
            robotState =
RobotState::GoingToWaitingPointHome;
        }
        break;
    }
    case RobotState::GoingToWaitingPointHome: {
        if (MoveTo(robot_Translate[0], waitingPoint.x,
speed) &&
            MoveTo(robot_Translate[2], waitingPoint.z,
speed)) {
            if (currentActiveOrder != nullptr &&
currentRackInOrderIndex < currentActiveOrder->RackNumbers->Count - 1) {
                currentRackInOrderIndex++;
                selectedRackIndex =
currentActiveOrder->RackNumbers[currentRackInOrderIndex];
                robotState = RobotState::ToRidePoint1;
                isMoving = true;
            }
            else {
                robotState = RobotState::Idle;
                isMoving = false;
                currentActiveOrder = nullptr;
                currentRackInOrderIndex = -1;
                selectedRackIndex = -1;
            }
        }
    }
}

```

```

        boxState = BOX_TO_POINT_1;
        boxTimer->Start();
        MessageBox::Show("Замовлення
повністю виконано!", "Інформація");
    }
}
break;
}
default:
    break;
}
this->RenderScene(nullptr, nullptr);
}
void BoxTimer_Tick(System::Object^ sender,
System::EventArgs^ e)
{
    static float speed = 0.06f;
    bool reachedTarget = false;

    switch (boxState) {
    case BOX_START:
        break;
    case BOX_TO_POINT_1:
        reachedTarget = MoveTo(box_Translate[0], 1.0f,
speed) &&
            MoveTo(box_Translate[2], 0.0f, speed);
        if (reachedTarget) {
            boxState = BOX_TO_POINT_2;
        }
        break;

```

```

case BOX_TO_POINT_2:
    reachedTarget = MoveTo(box_Translate[0], 1.0f,
speed) &&
        MoveTo(box_Translate[2], 5.5f, speed);
    if (reachedTarget) {
        boxState = BOX_BACK_TO_START;
    }
    break;
case BOX_BACK_TO_START:
    box_Translate[0] = { 0.4f };
    box_Translate[2] = { 0.0f };
    boxState = BOX_START;
    break;
}

this->RenderScene(nullptr, nullptr);
}
void MyForm_KeyDown(System::Object^ sender,
System::Windows::Forms::KeyEventArgs^ e)
{
    if (e->KeyCode == Keys::Enter && !isMoving &&
robotState == RobotState::Idle)
    {
        diplom::OrderSelectionForm^ orderForm = gcnw
diplom::OrderSelectionForm();
        if (orderForm->ShowDialog() ==
System::Windows::Forms::DialogResult::OK)
        {
            currentActiveOrder = orderForm-
>SelectedOrder;

```

```

        if (currentActiveOrder != nullptr &&
currentActiveOrder->RackNumbers->Count > 0)
        {
            currentRackInOrderIndex = 0;
            selectedRackIndex =
currentActiveOrder->RackNumbers[currentRackInOrderIndex];
            isMoving = true;
            robotState = RobotState::ToRidePoint1;
            boxTimer->Stop();
            boxState = BOX_START;
            animationTimer->Start();
        }
        else
        {
            MessageBox::Show("Замовлення не
вибрано або воно порожнє.", "Помилка", MessageBoxButtons::OK,
MessageBoxIcon::Warning);

            isMoving = false;
            robotState = RobotState::Idle;
        }
    }
}
else if (e->KeyCode == Keys::Space && robotState ==
RobotState::WaitingForSpace)
{
    spaceKeyPressed = true;
}
this->RenderScene(nullptr, nullptr);
}

```

```

void MyForm_Load(System::Object^ sender, System::EventArgs^
e)
{
    InitializeOpenGL();
    SetupProjection(this->ClientSize.Width, this-
>ClientSize.Height);
    glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
}
void MyForm_Resize(System::Object^ sender,
System::EventArgs^ e)
{
    SetupProjection(this->ClientSize.Width, this-
>ClientSize.Height);
}
void MyForm_Closing(System::Object^ sender,
System::Windows::Forms::FormClosingEventArgs^ e)
{
    renderTimer->Stop();
    wglMakeCurrent(NULL, NULL);
    wglDeleteContext(hglrc);
    ReleaseDC((HWND)this->Handle.ToPointer(), hdc);
}
#pragma endregion
};
}

```

ДОДАТОК Д

Код додаткової форми (MyForm1.h)

```

#pragma once
#include <vector>
#include <random>
#include <ctime>
#include <msclr/marshal_cppstd.h>
namespace diplom {
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    public ref class Order {
    public:
        System::String^ Name;
        System::Collections::Generic::List<int>^ RackNumbers;
        Order() {
            RackNumbers = gcnew System::Collections::Generic::List<int>();
        }
    };
    const int totalRacks = 25;
    public ref class OrderSelectionForm : public
System::Windows::Forms::Form
    {
    public:
        System::Collections::Generic::List<Order^>^ AllGeneratedOrders;

```

```

property Order^ SelectedOrder{
    Order ^ get() {
        if (lbOrders->SelectedIndex != -1) {
            return AllGeneratedOrders[lbOrders->SelectedIndex];
        }
        return nullptr;
    }
}
OrderSelectionForm(void)
{
    InitializeComponent();
    AllGeneratedOrders = gnew
System::Collections::Generic::List<Order^>();
    srand(time(0));
    GenerateOrders();
}
protected:
~OrderSelectionForm()
{
    if (components)
    {
        delete components;
    }
}
private: System::Windows::Forms::Button^ btnGenerateOrders;
private: System::Windows::Forms::ListBox^ lbOrders;
private: System::Windows::Forms::Button^ btnSelectOrder;
private:
    System::ComponentModel::Container^ components;

```

#pragma region Windows Form Designer generated code

```

void InitializeComponent(void)
{
    this->btnGenerateOrders = (gcnew
System::Windows::Forms::Button());
    this->lbOrders = (gcnew System::Windows::Forms::ListBox());
    this->btnSelectOrder = (gcnew System::Windows::Forms::Button());
    this->SuspendLayout();

    this->btnGenerateOrders->Location = System::Drawing::Point(12, 12);
    this->btnGenerateOrders->Name = L"btnGenerateOrders";
    this->btnGenerateOrders->Size = System::Drawing::Size(260, 40);
    this->btnGenerateOrders->TabIndex = 0;
    this->btnGenerateOrders->Text = L"Нові Замовлення";
    this->btnGenerateOrders->UseVisualStyleBackColor = true;
    this->btnGenerateOrders->Click += gcnew System::EventHandler(this,
&OrderSelectionForm::btnGenerateOrders_Click);

    this->lbOrders->FormattingEnabled = true;
    this->lbOrders->Location = System::Drawing::Point(12, 60);
    this->lbOrders->Name = L"lbOrders";
    this->lbOrders->Size = System::Drawing::Size(260, 160);
    this->lbOrders->TabIndex = 1;

    this->btnSelectOrder->Location = System::Drawing::Point(12, 226);
    this->btnSelectOrder->Name = L"btnSelectOrder";
    this->btnSelectOrder->Size = System::Drawing::Size(260, 40);
    this->btnSelectOrder->TabIndex = 2;
    this->btnSelectOrder->Text = L"Вибрати Замовлення";

```

```

this->btnSelectOrder->UseVisualStyleBackColor = true;
this->btnSelectOrder->Click += gcnew System::EventHandler(this,
&OrderSelectionForm::btnSelectOrder_Click);

this->AutoScaleDimensions = System::Drawing::SizeF(6.0F, 13.0F);
this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(284, 278);
this->Controls->Add(this->btnSelectOrder);
this->Controls->Add(this->lbOrders);
this->Controls->Add(this->btnGenerateOrders);
this->Name = L"OrderSelectionForm";
this->Text = L"Вибір ЗАМОВЛЕННЯ";
this->ResumeLayout(false);
}
#pragma endregion

private: System::Void btnGenerateOrders_Click(System::Object^ sender,
System::EventArgs^ e) {
    GenerateOrders();
}

private: System::Void GenerateOrders() {
    AllGeneratedOrders->Clear();
    lbOrders->Items->Clear();
    System::Random^ rand = gcnew System::Random();
    for (int i = 0; i < 10; ++i) {
        Order^ newOrder = gcnew Order();
        newOrder->Name = "ЗАМОВЛЕННЯ " + (i + 1).ToString();
        int numRacks = rand->Next(1, 6);
        for (int j = 0; j < numRacks; ++j) {
            int rackNum = rand->Next(0, totalRacks);

```

```

        if (!newOrder->RackNumbers->Contains(rackNum)) {
            newOrder->RackNumbers->Add(rackNum);
        }
        else {
            j--;
        }
    }
    AllGeneratedOrders->Add(newOrder);
    System::String^ rackListString = "";
    for each(int rackNum in newOrder->RackNumbers) {
        rackListString += rackNum.ToString() + " ";
    }
    lbOrders->Items->Add(newOrder->Name + " (Стелажі: " +
rackListString->Trim() + ")");
    }
}
private: System::Void btnSelectOrder_Click(System::Object^ sender,
System::EventArgs^ e) {
    if (lbOrders->SelectedIndex != -1) {
        this->DialogResult = System::Windows::Forms::DialogResult::OK;
        this->Close();
    }
    else {
        MessageBox::Show("Будь ласка, виберіть замовлення зі списку.",
"Помилка вибору", MessageBoxButtons::OK, MessageBoxIcon::Warning);
    }
}
};
}

```

ДОДАТОК Е

Демонстраційний матеріал

