Лабораторная работа №1

Уткин А.В 4210 вариант №2

Задание:  
  
1. Создайте программу, рисующую на экране график функции:  
  
y(t) = sin(ω1t)cos(ω2t), где ω1, ω2 – const  
  
2. Создайте программу, которая покрывает экранное окно window мозаикой по примеру, разобранному выше. Рисунок для элемента мозаики придумайте сами. Поэкспериментируйте с размерами элементов мозаики и их количеством.

Листинг программы №1:

#include <iostream>

#define GL\_SILENCE\_DEPRECATION

#include <GLUT/glut.h>

#include <cmath>

// Константы m1 и m2

**const** **double** m1 = 1.0; // Устанавливаем константы для m1

**const** **double** m2 = 1.0; // Устанавливаем константы для m2

**const** GLfloat red = 1.0f; // Красный цвет

**const** GLfloat green = 0.97f; // Зеленый цвет

**const** GLfloat blue = 0.86f; // Синий цвет

// Функция y(t)

**double** y(**double** t) {

**return** std::sin(m1 \* t) \* std::cos(m2 \* t);

}

// Функция отображения

**void** display() {

glClear(GL\_COLOR\_BUFFER\_BIT); // Очищаем буфер кадра (framebuffer) | GL\_COLOR\_BUFFER\_BIT - очищаем только цветовой буфер кадра

// Задаем цвет осей координат

glColor3f(0.0, 0.0, 0.0);

// Рисуем ось x

glLineWidth(2.0);

glBegin(GL\_LINES);

glVertex2d(0.0, 0.0);

glVertex2d(10.0, 0.0);

glEnd();

// Рисуем ось y

glLineWidth(2.0);

glBegin(GL\_LINES);

glVertex2d(0.0, -1.0);

glVertex2d(0.0, 1.0);

glEnd();

// Рисуем вертикальную ось x

glLineWidth(2.0);

glBegin(GL\_LINES);

glVertex2d(0.0, 0.0);

glVertex2d(0.0, -1.0);

glEnd();

// Рисуем вертикальную ось y

glLineWidth(2.0);

glBegin(GL\_LINES);

glVertex2d(0.0, 0.0);

glVertex2d(0.0, 1.0);

glEnd();

// Задаем цвет графика функции

glColor3f(220/255.0, 20/255.0, 60/255.0);

// Рисуем график функции

glLineWidth(1.0);

glBegin(GL\_LINE\_STRIP);

**for** (**double** t = 0.0; t <= 100.0; t += 0.01) {

glVertex2d(t, y(t));

}

glEnd();

glFlush();

}

**int** main(**int** argc, **char**\*\* argv) {

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutCreateWindow("Graph of y(t)"); // Имя окна

glClearColor(red, green, blue, 1.0f); // фон окна

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

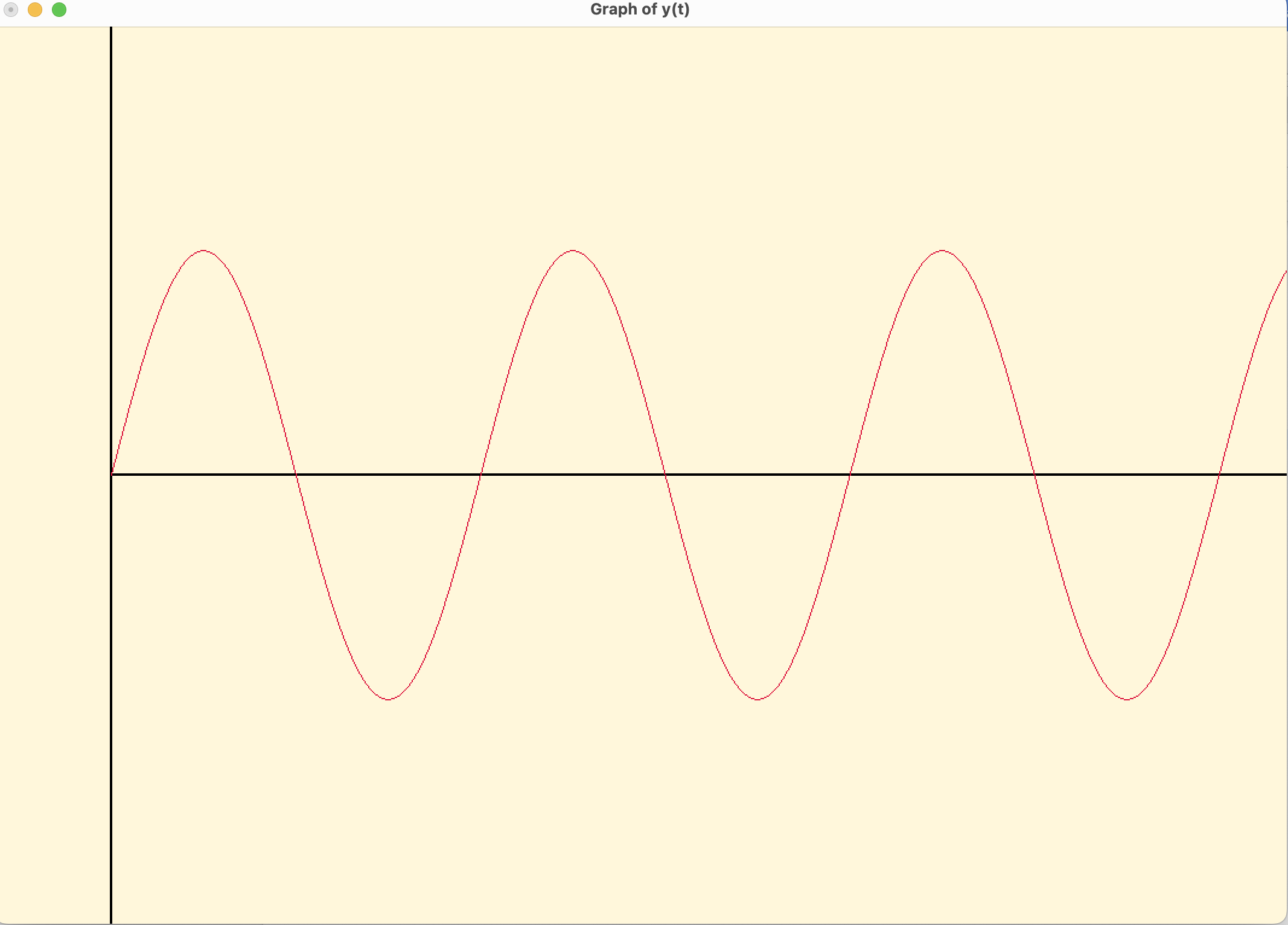
gluOrtho2D(-1.0, 10.0, -1.0, 1.0); // МСК

glutDisplayFunc(display); // Отоброжаем функцию "display"

glutMainLoop(); // Передаем управление внутреннему циклу GLUT

**return** 0;

}



Листинг программы №2

#include <iostream>

#include <stdio.h>

#define GL\_SILENCE\_DEPRECATION

#include <GLUT/glut.h>

#include <cmath>

**void** init()

{

glClearColor(0.9, 0.9, 0.9, 0.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-8.0, 8.0, -2.0, 2.0);

};

**void** draw()

{

GLfloat r = 0.0, g = 0.0, b = 0.0;

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0, 0.0, 0.0);

glViewport(0, 0, 800, 600);

GLint x, y, h = 60, w = 60;

**for**(x = 0; x <= 800; x += w)

{

**for**(y = 0; y <= 600; y += h)

{

glViewport(x, y, w, h);

glColor3f(1.0, 0.0, 0.0);

glBegin(GL\_TRIANGLE\_FAN);

glVertex2f(0, 0);

**for** (**int** i = 0; i <= 360; i++) {

glVertex2f(5\*cos(i) + 5, 5\*sin(i) + 5);

}

glEnd();

}

}

glFlush( );

};

**int** main(**int** argc, **char** \*\*argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(800, 600);

glutInitWindowPosition(250, 250);

glutCreateWindow("Lab 1. Task 1.1");

init();

glutDisplayFunc(draw);

glutMainLoop();

}

