МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Вычислительной техники

**ОТЧЁТ ПО РАСЧЕТНО-ГРАФИЧЕСКОЙ РАБОТЕ**

**ПО ДИСЦИПЛИНЕ «ИНФОРМАТИКА»**

**«Использование графической библиотеки glut.h»**

Факультет: АВТ Преподаватель: Копылова О.А.

Группа: ДТ-460а

Студент: Дроздов И.С.

Содержание

[Задание 3](#_Toc1)

[Идеи и методы решений 3](#_Toc2)

[Линейное движение 3](#_Toc3)

[Функциональное описание программы 3](#_Toc4)

[Перемещение рыб 4](#_Toc5)

[Использование клавиатуры 4](#_Toc6)

[Инициализирование окна 4](#_Toc7)

[Пример работы программы 5](#_Toc8)

[Ошибки и неточности 6](#_Toc9)

[Вывод 6](#_Toc10)

[Список литературы 7](#_Toc12)

[Приложение А. Исходный код программы 8](#_Toc11)

# Задание

Вариант 20. «Корабль на волнах». Корабль плавает по волнам. Волны поднимаются и опускаются, создавая эффект движения воды.

## Идеи и методы решений

Корабль должен появляться после того как уплыл за границу экрана.

В программе используется прямоугольная система координат с центром посередине экрана.

## Линейное движение

Т.к. корабль двумерный, то движение может производиться вверх, вниз, влево, вправо.

Для решения проблемы повторного появления корабля расстояние экрана увеличено в несколько раз. Когда корабль проплывает расстояние экрана, он перемещается в другой конец и плывет заново.

## Функциональное описание программы

**Переменные**

waveHeight, waveLength, waveSpeed, waveOffset — параметры волн

shipX, shipY, shipWidth, shipHeight, shipSpeed, shipAngle — параметры корабля

waterColor, shipColor, sailColor — параметры цветов для объектов

a, d — смена скорости с которой плывёт корабль.

« » - полная остановка корабля

esс — выход из приложения

**Функции рисования**

1. void drawWave() - рисование волн
2. void drawShip() - рисование корабля
3. void display() - отрисовка волн и корабля

**Функции инициализации и управления**

1. void display() - инициализация рисунков в окне
2. void keyboard(unsigned char key, int x, int y) - задание клавиш управления
3. int main(int argc, char\*\* argv) - главный цикл, в котором устанавливаются новые значения переменных

## Перемещение корабля

Чтобы корабль появлялся снова после проплывания экрана, увеличиваем размер экрана в 2 раза. Когда корабль проплывет это расстояние, он появится в начале экрана.

if (shipX > windowWidth + shipWidth) {

shipX = -shipWidth;

}

## Использование клавиатуры

Чтобы перемещать рыб по вертикали, будем использовать клавиатуру.

switch (key) {

case 27: // ESC

exit(0);

break;

case 'a':

shipSpeed -= 0.5f;

break;

case 'd':

shipSpeed += 0.5f;

break;

}

## Инициализирование окна

glutInitWindowSize(windowWidth, windowHeight); - задание размеров окна

glutCreateWindow("Корабль на волнах"); - название окна

glClearColor(0.53f, 0.81f, 0.92f, 1.0f); - задание цвета неба

GLfloat waterColor[] = {0.0f, 0.3f, 0.7f, 1.0f}; - цвет воды

GLfloat shipColor[] = {0.5f, 0.3f, 0.1f, 1.0f}; - цвет корабля

GLfloat sailColor[] = {0.9f, 0.9f, 0.9f, 1.0f}; - цвет паруса

glutDisplayFunc(display); - функция для прорисовки объектов в окне

glutKeyboardFunc(keyboard); - функция для управления с клавиатуры

glutReshapeFunc(reshape); - функция для прорисовки содержимого при смене размеров окна

## Пример работы программы

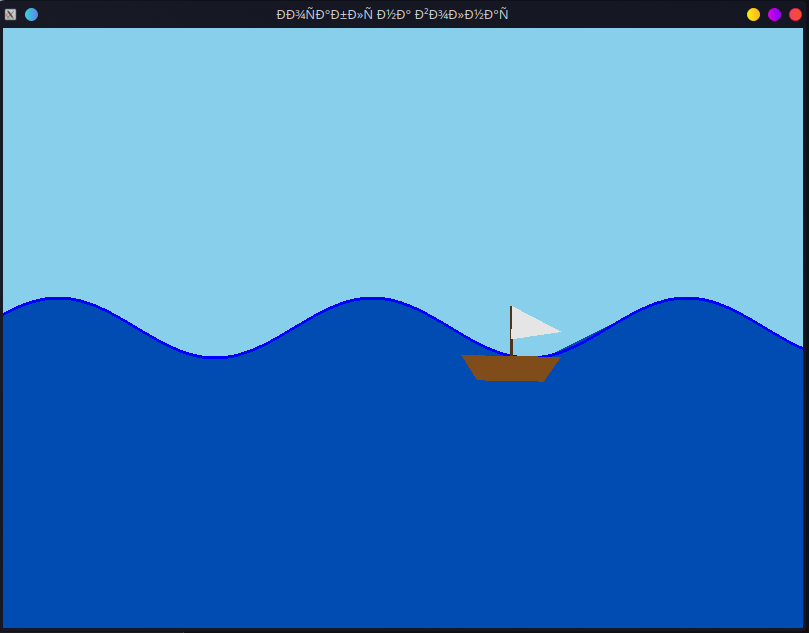


Рисунок 1 - Пример работы программы

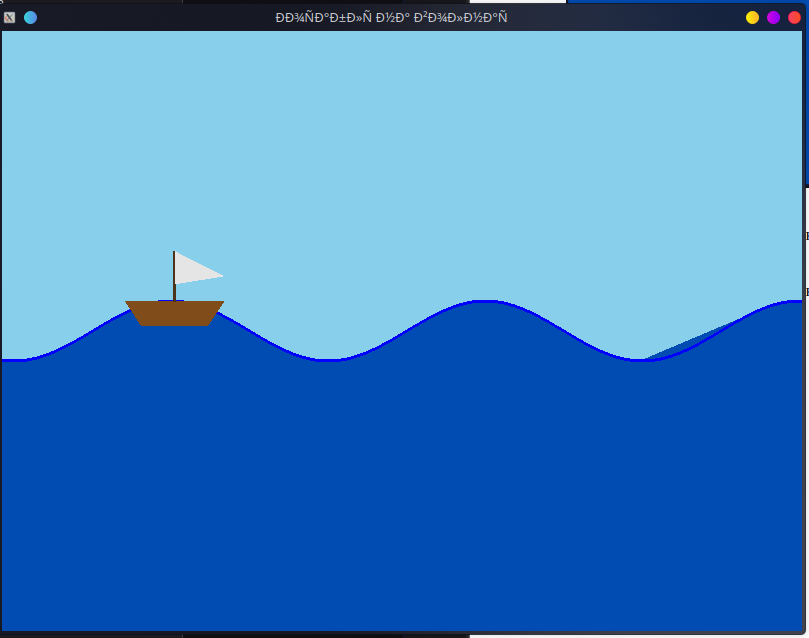


Рисунок 2 - Пример работы программы

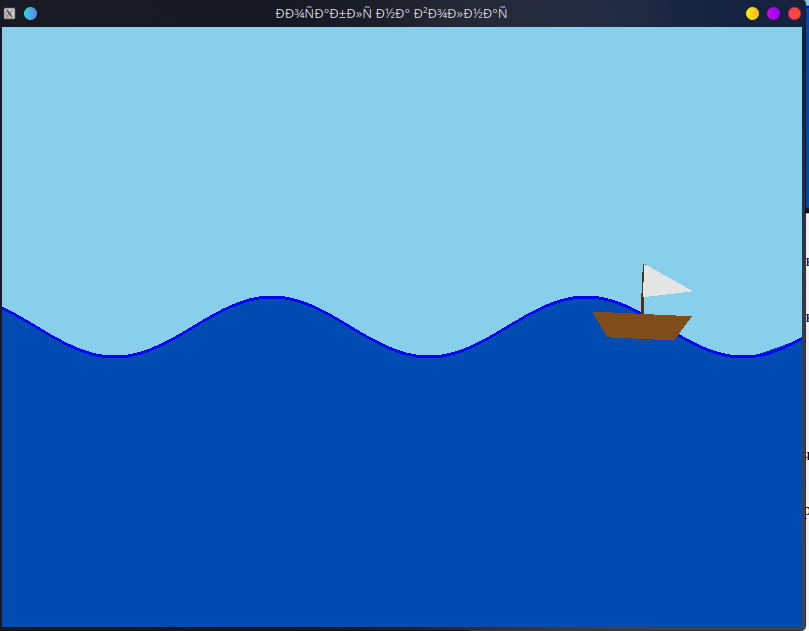


Рисунок 3 - Пример работы программы

## Ошибки и неточности

При задании кораблю отрицательной скорости и его проплывании за левую границу, он пропадёт. Так же окно приложения имеет некорректное название



## Вывод

В ходе выполнения расчетно-графической работы мной была написана программа по рисованию плывущего по волнам корабля. Были получены навыки работы с графической библиотекой glut и OpenGL.

# Список литературы

1.Подбельский В.В., Фомин С.С. Программирование на языке Си: Учеб.пособие.2-е доп.Изд. – М.: Финансы и статистика, 2004 – 600 с.

2.Романов Е. Л. Си/Си++. От дилетанта до профессионала. Электронное учебное "Программирование", "Технология программирования" для студентов 1–2 курсов направления 230100 : учеб. пособие / Е. Л. Романов. – Новосибирский государственный технический университет, № гос регистрации 0321000528, 2010 - 581 с.

3.Си/Си++ от дилетанта до профессионала [Электронный ресурс].

URL: http://ermak.cs.nstu.ru/cprog/HTML/index.htm

## Приложение А. Исходный код программы

#include <GL/glut.h>

#include <math.h>

#include <stdio.h>

// Размеры окна

int windowWidth = 800;

int windowHeight = 600;

// Параметры волн

float waveHeight = 30.0f;

float waveLength = 0.02f;

float waveSpeed = 0.05f;

float waveOffset = 0.0f;

// Параметры корабля

float shipX = 400.0f;

float shipY = 300.0f;

float shipWidth = 100.0f;

float shipHeight = 50.0f;

float shipSpeed = 2.0f;

float shipAngle = 0.0f;

// Цвета

GLfloat waterColor[] = {0.0f, 0.3f, 0.7f, 1.0f};

GLfloat shipColor[] = {0.5f, 0.3f, 0.1f, 1.0f};

GLfloat sailColor[] = {0.9f, 0.9f, 0.9f, 1.0f};

// Функция для отрисовки волны

void drawWave() {

glBegin(GL\_LINE\_STRIP);

glColor3f(0.0f, 0.0f, 1.0f);

for (int x = 0; x <= windowWidth; x += 5) {

float y = waveHeight \* sin(waveLength \* x + waveOffset) + windowHeight / 2;

glVertex2f(x, y);

}

glEnd();

}

// Функция для отрисовки корабля

void drawShip() {

// Корпус корабля

glBegin(GL\_POLYGON);

glColor3fv(shipColor);

glVertex2f(shipX - shipWidth/2, shipY);

glVertex2f(shipX + shipWidth/2, shipY);

glVertex2f(shipX + shipWidth/3, shipY - shipHeight/2);

glVertex2f(shipX - shipWidth/3, shipY - shipHeight/2);

glEnd();

// Мачта

glLineWidth(3.0f);

glBegin(GL\_LINES);

glColor3f(0.3f, 0.2f, 0.1f);

glVertex2f(shipX, shipY);

glVertex2f(shipX, shipY + shipHeight);

glEnd();

// Парус

glBegin(GL\_TRIANGLES);

glColor3fv(sailColor);

glVertex2f(shipX, shipY + shipHeight/3);

glVertex2f(shipX, shipY + shipHeight);

glVertex2f(shipX + shipWidth/2, shipY + shipHeight/2);

glEnd();

}

// Функция отрисовки сцены

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

// Рисуем море (заполненная волна)

glBegin(GL\_POLYGON);

glColor3fv(waterColor);

glVertex2f(0, 0);

glVertex2f(windowWidth, 0);

for (int x = windowWidth; x >= 0; x -= 5) {

float y = waveHeight \* sin(waveLength \* x + waveOffset) + windowHeight / 2;

glVertex2f(x, y);

}

glEnd();

// Рисуем волну (контур)

drawWave();

// Рисуем корабль

drawShip();

glutSwapBuffers();

}

// Функция обновления анимации

void update(int value) {

// Обновляем смещение волны для анимации

waveOffset += waveSpeed;

// Двигаем корабль

shipX += shipSpeed;

if (shipX > windowWidth + shipWidth) {

shipX = -shipWidth;

}

// Качаем корабль на волнах

float waveAtShip = waveHeight \* sin(waveLength \* shipX + waveOffset);

shipY = windowHeight / 2 + waveAtShip;

shipAngle = waveHeight \* waveLength \* cos(waveLength \* shipX + waveOffset) \* 5.0f;

glutPostRedisplay();

glutTimerFunc(16, update, 0); // ~60 FPS

}

// Функция обработки изменения размеров окна

void reshape(int w, int h) {

windowWidth = w;

windowHeight = h;

glViewport(0, 0, w, h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0, w, 0, h);

glMatrixMode(GL\_MODELVIEW);

}

// Функция обработки клавиатуры

void keyboard(unsigned char key, int x, int y) {

switch (key) {

case 27: // ESC

exit(0);

break;

case 'a':

shipSpeed -= 0.5f;

break;

case 'd':

shipSpeed += 0.5f;

break;

}

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(windowWidth, windowHeight);

glutCreateWindow("Корабль на волнах");

glClearColor(0.53f, 0.81f, 0.92f, 1.0f); // Цвет неба

glutDisplayFunc(display);

glutReshapeFunc(reshape);

glutKeyboardFunc(keyboard);

glutTimerFunc(0, update, 0);

glutMainLoop();

return 0;

}