МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 4

з навчальної дисципліни

# “ ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ ”

Поліморфізм. Обробка винятків

ВИКОНАВ

студент академічної групи КБ-23

Сергій Козирь

ПЕРЕВІРИВ

Викладач

Козірова Н.Л.

Кропивницький – 2024

ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Лабораторна робота №4

Поліморфізм. Обробка винятків

Мета: Ознайомитись з поняттям поліморфізму у мові C++ та навчитись використовувати віртуальні функції для досягнення поліморфізму. Також вивчити принципи обробки винятків у мові C++.

Завдання 1

Примітка Для виконання цього завдання вам потрібно середовище розробки Qt Creator, якщо Ви раніше не працювали з ним то встановіть, та ознайомтесь. Наступним кроком створіть новий проект Qt: Використовуйте Qt Creator для створення нового проекту типу "Qt Widgets Application". При створені проекту ви отримаєте всі файли, які потрібні і навіть більше, отже вітаю, з цим «більше» вам і прийдеться працювати, а якщо точніше то до цього завдання вам треба зробити інтерфейс. Тему для завдання можна обрати на вибір

1. Напишіть гру «Піймай муху», на формі знаходиться муха, яка має тікати від курсора, також на формі має знаходитися пастка, коли муха попадеться в пастку гра закінчується. В реалізації програми має бути похідний клас, що унаслідується від базового класу «QMainWindow», в похідному класі, перевизначить функцію «event»

Mainwindow.h

#include <QLabel>

#include <QMouseEvent>

#include <QPixmap>

namespace Ui {

class MainWindow;

}

class MainWindow : public QMainWindow {

Q\_OBJECT

public:

explicit MainWindow(QWidget \*parent = nullptr);

~MainWindow();

protected:

void mouseMoveEvent(QMouseEvent \*event) override; // Для обробки подій руху миші

private:

Ui::MainWindow \*ui;

QLabel \*fly; // Муха

QLabel \*swatter; // Мухобійка

QLabel \*gameOverLabel; // Повідомлення про завершення гри

bool gameOver = false; // Прапорець для завершення гри

void escapeFly(const QPoint &cursorPos); // Метод для втечі мухи

};

#endif // MAINWINDOW\_H

Mainwindow.cpp

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QRandomGenerator> // Додаємо для генерації випадкових чисел

#include <QPixmap>

MainWindow::MainWindow(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow) {

// Ініціалізація інтерфейсу

ui->setupUi(this);

// Створення мухи з використанням картинки

fly = new QLabel(this);

// Збільшуємо розмір мухи

QPixmap flyImage(":/images/fly.png");

int flyWidth = 100; // Новий розмір по ширині

int flyHeight = 100; // Новий розмір по висоті

fly->setPixmap(flyImage.scaled(flyWidth, flyHeight)); // Масштабуємо зображення

fly->setGeometry(50, 50, flyWidth, flyHeight); // Задаємо нові розміри та початкове положення

// Створення мухобійки з використанням картинки

swatter = new QLabel(this);

// Збільшуємо розмір мухобійки

QPixmap swatterImage(":/images/swatter.png");

int swatterWidth = 200; // Новий розмір по ширині

int swatterHeight = 200; // Новий розмір по висоті

swatter->setPixmap(swatterImage.scaled(swatterWidth, swatterHeight)); // Масштабуємо зображення

swatter->setGeometry(200, 200, swatterWidth, swatterHeight); // Задаємо нові розміри та початкове положення

// Повідомлення про завершення гри (спочатку приховане)

gameOverLabel = new QLabel(this);

gameOverLabel->setText("Game Over");

gameOverLabel->setStyleSheet("QLabel { color : red; font-size: 30px; }");

gameOverLabel->setGeometry(this->width()/2 - 100, this->height()/2 - 30, 200, 60); // Центр екрану

gameOverLabel->setVisible(false); // Спочатку приховане

}

MainWindow::~MainWindow() {

delete ui;

}

void MainWindow::mouseMoveEvent(QMouseEvent \*event) {

// Якщо гра завершена, ігноруємо події

if (gameOver) {

return;

}

// Отримуємо координати курсора

QPoint cursorPos = event->pos();

// Отримуємо позиції мухи та мухобійки

QRect flyRect = fly->geometry();

QRect swatterRect = swatter->geometry();

// Якщо курсор знаходиться на муху, вона повинна тікати

if (flyRect.contains(cursorPos)) {

escapeFly(cursorPos); // Муха тікає від курсора

}

// Перевірка, чи муха потрапила на мухобійку

if (flyRect.intersects(swatterRect)) {

fly->setVisible(false); // Ховаємо муху

gameOverLabel->setVisible(true); // Виводимо повідомлення "Game Over"

gameOver = true; // Завершуємо гру

}

}

void MainWindow::escapeFly(const QPoint &cursorPos) {

// Рух мухи випадковим чином по екрану

int flyX = fly->x();

int flyY = fly->y();

// Генеруємо випадкові зміщення

int offsetX = QRandomGenerator::global()->bounded(-500, 500); // Випадкове зміщення по X

int offsetY = QRandomGenerator::global()->bounded(-500, 500); // Випадкове зміщення по Y

int newX = flyX + offsetX;

int newY = flyY + offsetY;

// Перевірка, щоб муха не виходила за межі вікна

if (newX < 0) newX = 0;

if (newY < 0) newY = 0;

if (newX > this->width() - fly->width()) newX = this->width() - fly->width();

if (newY > this->height() - fly->height()) newY = this->height() - fly->height();

// Переміщуємо муху у нове випадкове місце

fly->move(newX, newY);

}

Images.qrc

<RCC>

<qresource prefix="/images">

<file>fly.png</file>

<file>swatter.png</file>

</qresource>

</RCC>



