

Разработка графического приложения на языке программирования C++

Соколова Я.П.

Информатика и робототехника

3 курс, Многопрофильный колледж ФГБОУ ВО «ТГТУ», г. Тамбов, Тамбовской области

Научный руководитель: Мосягина Н.Г., Многопрофильный колледж ФГБОУ ВО «ТГТУ», г. Тамбов, Тамбовской области

ВВЕДЕНИЕ

В современном мире графика играет огромную роль во многих аспектах жизни современного человека: от развлекательных приложений до профессиональных инструментов, графические приложения помогают нам взаимодействовать с информацией и окружающим миром.

Компьютерная графика — область информатики, занимающаяся созданием, хранением и обработкой различных изображений (рисунков, чертежей, мультипликации) на компьютере [1].

Компьютерная графика различается по типу представления графической информации и алгоритмами обработки изображений.

Современная компьютерная графика открывает перед человеком огромные возможности. От реалистичного отображения трехмерных объектов до создания захватывающих анимационных продуктов и обработки изображений, современные графические технологии позволяют воплотить в жизнь самые смелые и, как казалось на первый взгляд, невозможные идеи. Благодаря использованию специализированных библиотек и фреймворков, разработчики могут создавать удивительные визуальные эффекты, которые раньше казались недостижимыми.

Помимо этого, при помощи компьютерной графики моделируются различные программные интерфейсы.

Компьютерная графика является применимой во многих сферах жизнедеятельности человека:

— наука — современная научная компьютерная графика позволяет проводить вычислительные эксперименты с наглядным представлением их

результатов, а также осуществлять автоматизированное проектирование имплантов для костей и суставов;

— архитектура — использование компьютерных технологий в архитектуре привело к появлению нового специального направления в этой области, которое стало называться архитектурной визуализацией, что открыло специалистам более полное представление объекта, уровень качества построек вырос;

— медицина — в настоящее время существуют специально разработанные симуляторы для врачей, которые помогают повышать уровень и качество проводимых врачами операций, примером такого симулятора является Virtual Reality Surgical Simulator — это хирургический VR-симулятор, в котором в том числе опытные врачи-хирурги имеют возможность оттачивать свои навыки в контролируемой среде с использованием реальных трехмерных моделей, имитирующих реальные хирургические операции;

— графический дизайн и дизайн окружающей среды – компьютерная графика породила собой огромное множество современных профессий, пользующихся популярностью, одним из наиболее очевидных представителей таких специалистов является графический дизайнер, создающий визуальный контент: логотипы, баннеры, шрифты, брендбуки и многое другое;

— образование — в настоящее время благодаря внедрению в образовательный процесс возможностей компьютерной графики появилось множество обучающих симуляторов и образовательные версии популярных игр, которые способствуют более наглядному представлению информации и лучшему усваиванию;

— киноиндустрия — компьютерная графика достигла огромного уровня развития, из-за чего получила широкое использование в кинематографе, она позволяет сделать кино гораздо зрелищнее и реалистичнее;

Компьютерная графика является неотъемлемой частью искусства, которое в том числе открывает новые горизонты перед современными художниками, позволяя

им создавать абсолютно новый уровень визуализации художественных произведений.

Цель работы: изучить возможности языка C++ для создание графических приложений, осуществить проектирование собственного графического приложения.

Задачи исследования:

- изучить возможности современной компьютерной графики
- охарактеризовать наиболее распространенные средства разработки графических приложений
- изучить функции языка C++ для создания графических приложений
- разработать графическое приложение с использованием функций языка C++ для создания графических объектов.

Методы исследования. Проект выполнен с использованием методов объектно-ориентированного программирования. В качестве средств проектирования используется инструментальная среда разработки MS Visual Studio 2019, графические библиотеки C++[2].

Основная часть.

Язык программирования C++ является мощным средством для разработки как простых программ, так и для создания графических приложений различного функционала и назначения. Первоначально он был разработан для системного программирования, однако он также обрел широкое применение в создании профессиональных графических приложений благодаря таким своим преимуществам, как эффективность, скорость и масштабируемость. C++ имеет средства для создания графических приложений, в том числе специальные графические библиотеки, такие как SFML, wxWidgets, OpenGL, DirectX и другие. Графические библиотеки и фреймворки – программные платформы, упрощающие разработку программного продукта, предоставляют разработчикам доступ к мощным функциям и инструментам для работы с графикой. Они упрощают процесс создания визуальных эффектов, обеспечивая высокую производительность и качество отображения.

SFML (Simple and Fast Multimedia Library) — одна из самых удобных и быстрых графических библиотек для C++. Одним из важнейших и неоспоримых ее преимуществ выделяют минимальные требования к уровню знаний языка и лёгкость освоения. Также важным преимуществом данного средства разработки является кроссплатформенность, то есть SFML работает под такие операционные системы, как Windows, Linux и Mac OS.

SFML популярен среди небольших команд благодаря тому, что разработка графической части программы не требует написания больших объёмов кода [2].

wxWidgets — кроссплатформенная библиотека инструментов с открытым исходным кодом для разработки кроссплатформенных приложений. Это наиболее известная графическая библиотека, которая используется для построения графического интерфейса пользователя, который так же известен как GUI.

Основной плюс данного средства разработки заключается в высокой производительности. Для каждой системы в wxWidgets используется свой код, благодаря этому все элементы, такие как кнопки, полосы прокрутки и прочие элементы отображаются похожими на системный стиль.

Особенность, которая отличает wxWidgets от других фреймворков, таких как MFC, — многоплатформенная природа. wxWidgets имеет программный интерфейс, который одинаков на всех поддерживаемых платформах.

OpenGL (Open Graphics Library) — это спецификация, определяющая платформонезависимый программный интерфейс для написания приложений, использующих двумерную и трёхмерную компьютерную графику [3].

OpenGL дает возможность рисования сложных трехмерных сцен из привычных всем и простых графических примитивов. OpenGL имеет более 300 функций для рисования.

Данная библиотека часто применяется при создании компьютерных игр, САПР, а также визуализации в научных проектах и исследованиях.

MFC (Microsoft Foundation Classes) — библиотека классов языка C++, предназначенная для создания визуальных приложений при помощи встроенных

средств. MFC поддерживает современные технологии программирования, благодаря чему программы, написанные с ее помощью, будут обладать необходимой функциональностью и надежностью [3].

MFC имеет ряд процедур для создания графических примитивов: эллипс; линия; прямоугольник; прямоугольник со скругленными краями; сектор круга; дуга; полигон и др.

Для того, чтобы нарисовать любой объект в библиотеке MFC на языке C++ необходимо создать *перо*, которым будет выполняться обводка контура графического объекта. Перо имеет три основных параметра: стиль, ширина и цвет. Синтаксис команды создания пера выглядит следующим образом: `HPEN CreatePen(int fnPenStyle, int nWidth, COLORREF crColor);`

Пример создания пера: `CPen newPen(PS_SOLID, 3, RGB(0, 0, 0));`

`CPen* oldPen = pDC->SelectObject(&newPen);`

`pDC->SelectObject(oldPen);`

В данном примере реализуется создание черного сплошного пера с толщиной равной 3 пикселя.

Для заливки цветом замкнутых графических примитивов в MFC существует *кисть*. В зависимости от параметров кисти в MFC имеют несколько типов: `CBrush(int nIndex, COLORREF crColor)` – кисть со штриховкой; `CBrush(COLORREF crColor)` – кисть сплошного цвета.

Параметры кисти: `crColor` – определяет цвет кисти; `nIndex` – определяет стиль штриховки.

— `HS_FDIAGONAL` – 45-градусная штриховка слева направо и снизу вверх;

— `HS_CROSS` – горизонтальная и вертикальная штриховка крест-накрест;

— `HS_DIAGCROSS` – 45-градусная штриховка крест-накрест [4].

Цвет кисти также как и пера задается при помощи цветовой модели RGB.

Пример создания сплошной кисти черного цвета представлен ниже:

`CBrush newBrush(RGB(0, 0, 0));`

```
CBrush* oldBrush = pDC->SelectObject(&newBrush);
```

```
pDC->SelectObject(oldBrush);
```

Приведем примеры функций для создания графических примитивов.

Создание эллипса при помощи MFC

Процедура создания эллипса в библиотеке MFC имеет следующий вид: `BOOL Ellipse (int x1, int y1, int x2, int y2)`.

Функция `Ellipse` имеет несколько параметров: `x1` – определяет координату `x` левого верхнего угла ограничительного прямоугольника эллипса; `x2` – нижнего правого угла ограничительного прямоугольника эллипса; `y1` – левого верхнего угла ограничительного прямоугольника эллипса; `y2` – нижнего правого угла ограничительного прямоугольника эллипса.

Создание линии при помощи MFC

Процедура создания линии имеет следующий общий вид: `BOOL LineTo (int x, int y)`.

Данная функция имеет два параметра:

— `x` – `x`-координата конца линии;

— `y` – `y`-координата конца линии.

Функция `LineTo (int x, int y)` рисует линию текущим выбранным цветом пера из текущей позиции в точку с координатами `x`, `y`.

Текущую позицию можно изменить при помощи использования функции `MoveTo (int x, int y)`. Данная функция изменяет текущую позицию в точку с координатами `x` и `y`, указываемыми в параметрах функции `MoveTo`.

Создание прямоугольника при помощи MFC

Процедура создания прямоугольника при помощи MFC имеет следующий вид: `BOOL Rectangle (int x1, int y1, int x2, int y2)`.

Данная функция выводит прямоугольник, использующий текущее перо. Цвет прямоугольника зависит от текущей используемой кисти.

Функция `Rectangle` имеет несколько параметров: `x1` – определяет координату `x` левого верхнего угла прямоугольника; `x2` – определяет координату

x нижнего правого угла прямоугольника; y1 – определяет координату у левого верхнего угла прямоугольника; y2 – нижнего правого угла прямоугольника.

Код, представленный ниже, реализует рисование таких графических примитивов, как эллипса, прямоугольника, прямоугольника со скругленными краями, линии, дуги окружности, а также сектора окружности.

```
CPen newPen(PS_SOLID, 3, RGB(81, 145, 17));
CPen* oldPen = pDC->SelectObject(&newPen);
pDC->MoveTo(200, 100); pDC->LineTo(300, 150);
CBrush newBrush(RGB(141, 199, 84));
CBrush* oldBrush = pDC->SelectObject(&newBrush);
pDC->Ellipse(180, 180, 285, 260); pDC->Rectangle(50, 300, 150, 400);
pDC->SelectObject(oldBrush); pDC->SelectObject(oldPen);
CBrush newBrush1(RGB(13, 189, 148));
CBrush* oldBrush1 = pDC->SelectObject(&newBrush1);
CPen newPen1(PS_SOLID, 3, RGB(3, 145, 112));
CPen* oldPen1 = pDC->SelectObject(&newPen1);
pDC->RoundRect(20, 50, 110, 150, 10, 10);
pDC->Pie(300, 50, 400, 150, 300, 50, 300, 100);
pDC->Arc(100, 200, 200, 150, 175, 175, 150, 150);
pDC->SelectObject(oldBrush1); pDC->SelectObject(oldPen1);}
```

Результат выполнения программы изображен на рисунке 1.

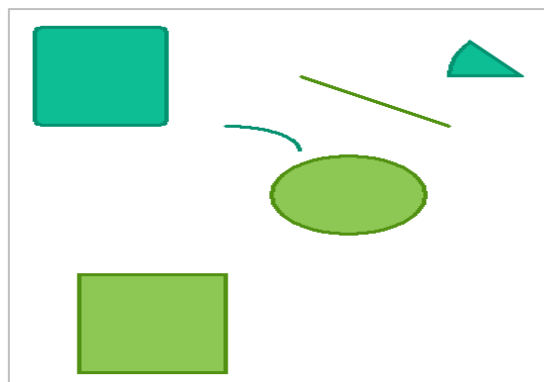


Рисунок 1 – Изображение графических примитивов в MFC

Мною также был выполнен рисунок в стиле пиксель арт – изображение, нарисованное пикселями (квадратами).

Для оптимизации кода и уменьшения его громоздкости были использованы циклы for с заданным числом повторений. Данный цикл используется в случае необходимости организации известного числа повторений оправленного действия. Программный код также содержит создание нескольких кистей различных по цвету.

Фрагмент кода данной программы представлен ниже.

```
// MFCApplication2View.cpp: реализация класса CMFCApplication2View
```

```
void CMFCApplication2View::OnDraw(CDC* pDC)
```

```
{
```

```
CMFCApplication2Doc* pDoc = GetDocument();
```

```
ASSERT_VALID(pDoc);
```

```
if (!pDoc)
```

```
return;
```

```
CPen newPen0(PS_SOLID, 1, RGB(249, 247, 242)); //создание кисти для  
заливки изображения
```

```
CPen* oldPen0 = pDC->SelectObject(&newPen0);
```

```
CBrush newBrush0(RGB(249, 247, 242));
```

```
CBrush* oldBrush0 = pDC->SelectObject(&newBrush0);
```

```
pDC->Rectangle(10, 10, 500, 450);
```

```
pDC->SelectObject(&oldBrush0);
```

```
pDC->SelectObject(&oldPen0);
```

```
CPen newPen(RGB(98, 195, 214)); //создание кисти для  
заливки фона
```

```
CPen* oldPen = pDC->SelectObject(&newPen);
```

```
CBrush newBrush(RGB(98, 195, 214));
```

```
CBrush* oldBrush = pDC->SelectObject(&newBrush);
```

```
for (int i = 0; i < 10; i++)
```

```
{ pDC->Rectangle(10+i*10, 10, 20+i * 10, 20); }
```

```
for (int i = 0; i < 38; i++)
```

```
{ pDC->Rectangle(130 + i * 10, 10, 140 + i * 10, 20); }
```



```
for (int i = 0; i < 10; i++)
{pDC->Rectangle(10 + i * 10, 20, 20 + i * 10, 30);}
for (int i = 0; i < 37; i++)
{pDC->Rectangle(140 + i * 10, 20, 150 + i * 10, 30);}
for (int i = 0; i < 10; i++)
{pDC->Rectangle(10 + i * 10, 30, 20 + i * 10, 40);}
for (int i = 0; i < 35; i++)
{pDC->Rectangle(160 + i * 10, 30, 170 + i * 10, 40);}
for (int i = 0; i < 11; i++)
{pDC->Rectangle(10 + i * 10, 40, 20 + i * 10, 50);}
for (int i = 0; i < 33; i++)
{pDC->Rectangle(180 + i * 10, 40, 190 + i * 10, 50);}
for (int i = 0; i < 11; i++)
{pDC->Rectangle(10 + i * 10, 50, 20 + i * 10, 60);}
...
for (int i = 0; i < 4; i++)
{pDC->Rectangle(330 + i * 10, 70, 340 + i * 10, 80);}
```

Результат запуска программы представлен на рисунке 2.



Рисунок 2 – Изображение в стиле пиксель арт выполненное на языке C++

ЗАКЛЮЧЕНИЕ

Разработка графических приложений на языке программирования С++ требует от разработчиков знания особенностей и специфики языка и основ компьютерной графики. Однако благодаря мощным средствам языка и выбору специализированных графических библиотек и фреймворков, создание сложных визуальных приложений становится доступным более широкому количеству специалистов и любителей программирования.

При должном опыте работы с той или иной графической библиотекой языка С++ графические приложения могут быть не только функциональными, но и визуально привлекательными, тем самым открывая новые возможности для творчества, самовыражения и инноваций в мире компьютерной графики.

Список литературы

1. Национальный открытый университет. Основы работы в графическом редакторе GIMP [Электронный ресурс]. – Режим доступа: <https://intuit.ru/studies/courses/20282/1318/lecture/31039> (дата обращения 09.04.2024).
2. Обзор графических библиотек С++ [Электронный ресурс]. – Режим доступа: <https://tproger.ru/digest/cpp-best-gui?ysclid=lte2jbh0ve727902399> (дата обращения 09.04.2024).
3. Kvodo.ru Введение в Visual C++. MFC. [Электронный ресурс]. – Режим доступа: https://kvodo.ru/visual-c-mfc-vvedenie.html?utm_referrer=https%3A%2F%2Fyandex.ru%2F (дата обращения 09.04.2024).
4. Stud Files. Рисование графических примитивов. [Электронный ресурс]. – Режим доступа: <https://studfile.net/preview/1850055/page:9/> (дата обращения 09.04.2024).