**Отчет по Главе 11: Графика (Graphics)**

**1. Введение**

Данная глава посвящена изучению возможностей 2D-графики в .NET MAUI с использованием пространства имен Microsoft.Maui.Graphics. Эта библиотека предоставляет кроссплатформенный API для рисования на холсте (canvas), позволяя создавать кастомные визуальные элементы, диаграммы, графики и другие графические представления непосредственно в приложении. Понимание основ работы с графикой необходимо для реализации нестандартных UI-элементов и визуализации данных, что является важным аспектом при разработке многофункциональных настольных и мобильных приложений.

**2. Основная часть**

**2.1. Основные концепции и принципы**

* **Microsoft.Maui.Graphics**: Пространство имен, содержащее кроссплатформенные типы для работы с 2D-графикой.
* **GraphicsView**: Элемент управления .NET MAUI, который выступает в качестве "холста" для рисования. Он отображает графику, определенную объектом, который реализует интерфейс IDrawable.
* **IDrawable**: Интерфейс с единственным методом Draw(ICanvas canvas, RectF dirtyRect). Любой класс, реализующий этот интерфейс, может быть использован для рисования внутри GraphicsView.
* **ICanvas**: Интерфейс, представляющий собой холст для рисования. Он предоставляет методы для рисования различных примитивов (линий, фигур), установки свойств рисования (цвет, толщина линии, стиль) и выполнения трансформаций (масштабирование, поворот, сдвиг).
* **Основные методы ICanvas:**
  + DrawLine(x1, y1, x2, y2): Рисует линию.
  + DrawRectangle(x, y, width, height): Рисует контур прямоугольника.
  + FillRectangle(x, y, width, height): Рисует закрашенный прямоугольник.
  + DrawRoundedRectangle(...): Рисует прямоугольник с закругленными углами.
  + DrawEllipse(...): Рисует эллипс (или круг, если ширина и высота равны).
  + DrawArc(...): Рисует дугу.
  + DrawPath(PathF path): Рисует сложную фигуру, определенную объектом PathF.
  + DrawString(value, x, y, horizontalAlignment): Рисует текст.
* **Свойства ICanvas для настройки рисования:**
  + StrokeColor, FillColor: Определяют цвет контура и заливки.
  + StrokeSize: Определяет толщину контура.
  + StrokeDashPattern: Задает шаблон для создания пунктирных линий.
  + Font, FontColor, FontSize: Определяют свойства для рисования текста.

**2.2. Примеры кода с пояснениями**

Для демонстрации графических возможностей был создан класс GraphicsDrawable, реализующий IDrawable, и страница GraphicsDemoPage.xaml, на которой размещен GraphicsView.

**2.2.1. GraphicsDrawable.cs**  
Этот класс содержит всю логику рисования. Метод Draw вызывается системой всякий раз, когда GraphicsView необходимо перерисовать свое содержимое.

// Фрагменты из GraphicsDrawable.cs

using Microsoft.Maui.Graphics;

public class GraphicsDrawable : IDrawable

{

public void Draw(ICanvas canvas, RectF dirtyRect)

{

// Устанавливаем цвет и толщину линии

canvas.StrokeColor = Colors.DarkBlue;

canvas.StrokeSize = 4;

canvas.DrawLine(10, 10, 90, 100);

// Рисуем закрашенный прямоугольник

canvas.FillColor = Colors.IndianRed;

canvas.FillRectangle(150, 230, 100, 50);

// Рисуем текст

canvas.FontColor = Colors.Purple;

canvas.FontSize = 18;

canvas.Font = Microsoft.Maui.Graphics.Font.DefaultBold; // Явное указание пространства имен

canvas.DrawString("Hello, Graphics!", 10, 300, HorizontalAlignment.Left);

// Рисуем путь (path) - треугольник

PathF path = new PathF();

path.MoveTo(50, 350);

path.LineTo(150, 350);

path.LineTo(100, 450);

path.Close();

canvas.StrokeColor = Colors.Green;

canvas.DrawPath(path);

}

}

*Пояснение:* В методе Draw последовательно вызываются методы ICanvas для рисования различных фигур. Перед каждым вызовом метода рисования можно установить свойства, такие как StrokeColor, FillColor, StrokeSize, которые будут применяться к последующим операциям.

**2.2.2. GraphicsDemoPage.xaml**  
На XAML-странице размещается GraphicsView, свойству Drawable которого присваивается экземпляр нашего класса GraphicsDrawable.

<!-- Фрагмент GraphicsDemoPage.xaml -->

<ContentPage xmlns:local="clr-namespace:HelloMauiApp" ... >

<!-- ... -->

<Frame Padding="10" BorderColor="LightGray" HeightRequest="500">

<GraphicsView>

<GraphicsView.Drawable>

<!-- Здесь мы используем наш класс GraphicsDrawable, который реализует IDrawable -->

<local:GraphicsDrawable/>

</GraphicsView.Drawable>

</GraphicsView>

</Frame>

<!-- ... -->

</ContentPage>

*Пояснение:* GraphicsView использует экземпляр GraphicsDrawable для получения инструкций по рисованию. xmlns:local используется для ссылки на класс GraphicsDrawable из нашего пространства имен.

**3. Заключение**

API Microsoft.Maui.Graphics предоставляет мощный и гибкий набор инструментов для создания кастомной 2D-графики в приложениях .NET MAUI. Использование GraphicsView в связке с классами, реализующими IDrawable, позволяет инкапсулировать логику рисования и создавать сложные, динамические визуализации. Эти возможности особенно полезны при разработке настольных приложений, где может потребоваться создание кастомных контролов, диаграмм, графиков или специализированных редакторов.

**4. Скриншоты и примеры**

Демонстрация работы с графикой была выполнена на странице GraphicsDemoPage, запущенной на платформе Windows.

*Рис. 11.1. Демонстрация рисования различных графических примитивов с помощью GraphicsView и IDrawable.*

**Процесс выполнения примеров:**

1. Создан класс GraphicsDrawable, реализующий интерфейс IDrawable.
2. В методе Draw класса GraphicsDrawable реализована логика рисования различных фигур: линий, прямоугольников, эллипсов, путей, дуг и текста.
3. Создана страница GraphicsDemoPage.xaml, на которой размещен элемент GraphicsView.
4. В качестве Drawable для GraphicsView был указан экземпляр класса GraphicsDrawable.
5. В процессе разработки была устранена ошибка неоднозначности ссылки на тип Font путем явного указания пространства имен Microsoft.Maui.Graphics.Font.
6. Страница добавлена в AppShell.xaml и протестирована.  
   Все графические элементы были успешно отрисованы на холсте GraphicsView, как и было определено в коде.

**5. Документация**

* **Ссылка на GitHub с исходным кодом:** https://github.com/RaniZee/.NET-MAUI-Showcase-Cross-Platform-Applications
* **Описание выполненных шагов:**
  1. Изучены материалы по теме "Графика" (глава 11) с ресурса [metanit.com/sharp/maui/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fmetanit.com%2Fsharp%2Fmaui%2F).
  2. Создана страница GraphicsDemoPage и класс GraphicsDrawable.
  3. Реализован пример рисования различных примитивов на ICanvas.
  4. Продемонстрировано использование GraphicsView для отображения кастомной графики.
  5. Устранены возникшие в процессе разработки ошибки компиляции.
  6. Проведено тестирование на платформе Windows и подготовлены скриншоты.