

ОБЗОР КРОССПЛАТФОРМЕННЫХ БИБЛИОТЕК ДЛЯ ПОСТРОЕНИЯ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ C#

И.Т. Новиков, В.В. Лебедев

© Новиков И.Т., Лебедев В.В., 2026

***Аннотация.** В статье рассмотрены библиотеки для построения кроссплатформенного графического интерфейса на языке программирования C# в рамках программной платформы .Net.*

***Ключевые слова:** графический интерфейс, C#, библиотеки графического интерфейса, кроссплатформенность.*

Разработка настольных прикладных приложений для компьютера тесно связана с понятием графического интерфейса. Чтобы облегчить разработку программного обеспечения, программисты обращаются к библиотекам графического интерфейса. Для одного и того же языка программирования может существовать большое количество таких библиотек. Однако библиотеки графического интерфейса тесно связаны с операционной системой (ОС), так как они основываются на системных вызовах и работе с графическим драйвером. В наши дни в связи с ростом популярности различных ОС и в связи с импортозамещением в Российской Федерации все больше программистов смотрит в сторону кроссплатформенных решений для разработки одного приложения, которое будет работать на разных системах. В настоящей статье будут рассмотрены основные кроссплатформенные библиотеки для программной платформы .Net и языка программирования C#.

Если обратиться к истории развития языка программирования C# и тесно связанной с ним платформы .Net, можно увидеть, что до современной платформы .Net (бывшая .Net Core) существовала .Net Framework. В состав .Net Framework входили такие библиотеки графического интерфейса, как WinForms и WPF. Несмотря на гибкость, простоту и удобства, у этих библиотек был главный минус: отсутствие кроссплатформенности. WinForms является надстройкой над стандартными библиотеками ОС Windows (в первую очередь Gdi32.dll), а WPF основывалась на технологии DirectX API. Обе технологии работают только на ОС Windows. В связи с развитием других ОС (MacOS, GNU/Linux), а также смартфонов (Android, iOS) стало понятно, что требуются новые библиотеки. Из-за этого сообщество программистов и сами авторы .Net

(корпорация Microsoft) приступили к разработке своих кроссплатформенных библиотек.

Рассмотрим основные кроссплатформенные библиотеки для разработки на языке C# и платформе .Net.

Avalonia

Avalonia – это библиотека с открытым исходным кодом, разрабатываемая энтузиастами [1, 8].

Данная библиотека в первую очередь основана на библиотеке для рисования Skia. Skia представляет собой библиотеку с открытым исходным кодом для работы с 2D-графикой. Она разработана компанией Google и написана на языке C++. Для работы со Skia на платформе .Net есть библиотека SkiaSharp, которая позволяет вызывать функции библиотеки Skia.

Для вызова системно зависимых функций в Avalonia используется специальный «низкоуровневый» слой. Этот слой вызывает стандартные функции для ОС Windows и MacOS; для GNU/Linux происходит вызов библиотек X11 или Wayland.

При создании Avalonia вдохновлялись встроенной в .Net библиотекой WPF, поэтому Avalonia, как и WPF, поддерживает разметку XAML для описания интерфейса, MVVM-паттерн для построения приложения. Кроме того, структура Avalonia очень схожа со структурой WPF, что позволяет программистам на WPF быстро переучиваться на другую библиотеку [1].

Несмотря на весомые достоинства указанной библиотеки, она не лишена недостатков. Главный из них заключается в том, что данная библиотека разрабатывается энтузиастами, а это сильно сказывается на качестве. Больших проектов на библиотеке немного, но все же они есть. На данный момент Avalonia является самой перспективной библиотекой для построения кроссплатформенного графического интерфейса на C#.

Uno Platform

Uno Platform – это библиотека с открытым исходным кодом, разрабатываемая энтузиастами [2, 8, 9].

Интерфейс библиотеки такой же, как и у библиотеки WinUI, которая используется для построения графических интерфейсов в ОС Windows 10. Если разработка ведется на ОС Windows, то Uno просто передает свои полномочия библиотеке WinUI, но если разработка ведется на других ОС, например мобильных (Android/iOS), то Uno создает нативный («родной») элемент управления для данной целевой ОС. Это позволяет легко портировать свои приложения на другие ОС с минимальным изменением кода.

Несмотря на то, что Uno Platform создавалась с оглядкой на WinUI, она не лишена недостатков. Недостатки примерно такие же, как и у Avalonia: Uno разрабатывается энтузиастами (что сказывается на качестве),

имеет невысокую популярность и малое количество проектов. Тем не менее Uno Platform, как и Avalonia, является довольно перспективным выбором.

Eto.Forms

Eto.Forms – это библиотека с открытым исходным кодом, разрабатываемая энтузиастами [3]. При ее разработке сильно вдохновлялись библиотекой WinForms, поэтому переход с этой библиотеки на Eto будет достаточно легким.

Eto.Forms базируется на использовании нативных компонентов графического интерфейса на целевой ОС. Так, при разработке под ОС Windows Eto.Forms будет создавать интерфейс на базе библиотек WinForms или WPF (на выбор программиста); при разработке для ОС GNU/Linux – GTK#3; MacOS – MonoMac и т.д. Таким образом, Eto.Forms является надстройкой над другими библиотеками для разработки графических интерфейсов на C#. За счет этого можно разрабатывать один проект на Eto.Forms и запускать его на других ОС, не изменяя интерфейс проекта.

Недостатки у Eto.Forms те же самые, что и у Avalonia с Uno Platform: невысокое качество, малое сообщество программистов, небольшое число проектов.

GTK# и его вариации

GTK# – это библиотека с открытым исходным кодом, которая является надстройкой для другой открытой библиотеки – GTK, написанной на языке программирования C. На базе данной библиотеки создано много программ, особенно под ОС GNU/Linux. Самые известные программы на GTK – это растровый редактор GIMP и графические оболочки под GNU/Linux: GNOME, XFCE, MATE, Cinnamon, Lxde и другие.

Главной особенностью GTK# является возможность разрабатывать приложения под GNU/Linux, используя как возможности платформы .Net, так и системные компоненты ОС GNU/Linux.

Основная проблема библиотеки GTK# заключается в использовании старой библиотеки GTK 3, в то время как уже существует GTK 4. GTK# довольно медленно развивается. Кроме того, на базе GTK# существуют библиотеки для связывания .Net и GTK (например, Gir.Core) [4].

Несмотря на указанные недостатки, GTK# и другие библиотеки на ее базе являются хорошим выбором при разработке графических приложений на ОС GNU/Linux.

Xamarin

Xamarin – это библиотека, разработанная одноименной компанией. Она рассчитана в первую очередь на мобильные устройства (Android и iOS), однако приложения на Xamarin также могут работать и на настольных ОС Windows и Mac.

Xamarin является надстройкой над нативными элементами управления графическим интерфейсом на своих ОС. Таким образом, при создании элемента «кнопка» на Xamarin происходит создание этого элемента на «родных» для данной ОС языках программирования (Java у Android, Objective-C у iOS) [5, 9].

В 2016 году корпорация Microsoft купила Xamarin [9]. В 2024 году поддержка библиотеки Xamarin была прекращена. В качестве замены еще в 2022 году программисты создали библиотеку MAUI.

Xamarin в первую очередь рассчитан на разработку мобильных приложений, однако разработка приложений для Windows и MacOS тоже поддерживается. В наши дни эта библиотека не пользуется популярностью: при создании мобильных приложений предпочтения отдаются нативным для платформы языкам программирования. Если необходимо разработать приложения под мобильные устройства на языке C#, то эффективнее использовать MAUI или Avalonia.

.Net MAUI

.Net MAUI – это библиотека, разработанная корпорацией Microsoft для их программной платформы .Net. Как и Xamarin, MAUI рассчитана на мобильные платформы, но эта библиотека также поддерживает ОС Windows и MacOS.

Главная цель MAUI заключается в том, чтобы улучшить устаревший Xamarin. Библиотека отличается новой архитектурой, что повышает производительность приложений и облегчает разработку. В отличие от Xamarin, для MAUI достаточно разработать только один проект. Внутри одного проекта будут специальные папки «Platforms/<Название целевой платформы>», где разместится платформозависимый код, который станет использоваться только при разработке на данной ОС [6, 8].

Библиотека MAUI была создана для того, чтобы закрыть недостатки Xamarin, но у нее есть некоторые недостатки. В первую очередь это известность: MAUI является довольно новой технологией, поэтому проектов на ней немного, что негативно сказывается на популярности. Кроме того, MAUI не поддерживает ОС GNU/Linux. Некоторые энтузиасты пытались портировать MAUI на данные ОС, но проект был закрыт [8].

Blazor

Blazor – это фреймворк для создания интерактивных приложений с графическим интерфейсом. В отличие от других перечисленных технологий, Blazor рассчитан скорее на веб-приложения. Blazor вдохновлялся такими фреймворками, как Angular, React, VueJS, но вместо языка JavaScript в Blazor используется C#. Интерфейс на Blazor описывается через HTML и CSS.

Приложения на Blazor можно запускать как обычные настольные с применением технологии Blazor Hybrid. В этом случае приложение на

Blazor открывается поверх библиотеки MAUI. Это похоже на разработку приложений на базе технологии Electron. При использовании Blazor Hybrid позволяет запускать как клиентскую, так и серверную часть на одной машине [7, 8].

Blazor является хорошим выбором для тех, кто собирается разрабатывать настольные приложения с графическим интерфейсом и имеет опыт разработки веб-приложений. Однако в этом кроется и главный недостаток Blazor: приложение на Blazor Hybrid – это локальная версия веб-приложения, а не чистый графический интерфейс, который получается путем вызова нативных функций целевой платформы.

Игровые движки (Unity, Godot, Redot, MonoGame и подобные)

Язык программирования C# часто применяется в игровых движках для описания сценариев. Это сделало язык программирования C# довольно популярным в сфере разработки видеоигр.

Одним из самых известных игровых движков, использующих язык программирования C#, является Unity. Он получил широкую популярность, особенно в любительской среде, поскольку имеет бесплатную лицензию для определенных разработчиков, а также большое количество учебного материала [10].

Кроме Unity, существует игровой движок с открытым исходным кодом Godot. Он тоже поддерживает программирование на языке C# и является полностью бесплатным. При этом он хуже проработан, что делает его менее популярным [11].

Существуют и другие игровые движки, которые позволяют вести разработку на языке программирования C#, но они не столь известны как в профессиональной среде, так и в любительской.

Программист может разработать приложение с графическим интерфейсом, поскольку в игровых движках есть возможность создавать интерфейс. Эта возможность используется для создания показателя здоровья у персонажа или меню паузы/настроек. Тем не менее игровые движки используются в первую очередь для разработки видеоигр, а не для создания приложений с графическим интерфейсом. Использовать движки для таких приложений будет неудобно, так как выбор элементов управления графическим интерфейсом довольно мал и многое придется разрабатывать самому. Несмотря на названные недостатки, применение игровых движков может быть целесообразно, если существует необходимость использовать 3D-графику в приложении.

Небольшое сравнение кроссплатформенных библиотек для C#.Net представлено в таблице.

Сравнение основных кроссплатформенных библиотек
для построения графического интерфейса на C#/.Net

Библиотека	Платформы	Поддержка
<i>1</i>	<i>2</i>	<i>3</i>
Avalonia	Windows	Да
	MacOS	Да
	GNU/Linux	Да
	Android	Да
	iOS	Да
	Web	Да
UNO Platform	Windows	Да
	MacOS	Да
	GNU/Linux	Да
	Android	Да
	iOS	Да
	Web	Да
Eto.Forms	Windows	Да
	MacOS	Да
	GNU/Linux	Да
	Android	Да
	iOS	Да
	Web	Нет
GTK#	Windows	Да
	MacOS	Да
	GNU/Linux	Да
	Android	Нет
	iOS	Нет
	Web	Нет
Xamarin	Windows	Да
	MacOS	Да
	GNU/Linux	Нет
	Android	Да
	iOS	Да
	Web	Нет
MAUI	Windows	Да
	MacOS	Да
	GNU/Linux	Нет
	Android	Да
	iOS	Да
	Web	Нет
Blazor	Windows	Да
	MacOS	Да
	GNU/Linux	Да
	Android	Да
	iOS	Да
	Web	Да

1	2	3
Unity (игровой движок)	Windows	Да
	MacOS	Да
	GNU/Linux	Да
	Android	Да
	iOS	Да
	Web	Да
Godot (игровой движок)	Windows	Да
	MacOS	Да
	GNU/Linux	Да
	Android	Да
	iOS	Да
Игровые движки (остальные)	Windows	Зависит от движка
	MacOS	Зависит от движка
	GNU/Linux	Зависит от движка
	Android	Зависит от движка
	iOS	Зависит от движка
	Web	Зависит от движка

Помимо перечисленных библиотек, существуют, например, QtSharp, SpaceVII и некоторые другие, но они не получили широкой популярности среди разработчиков, из чего следует, что на этих библиотеках написано мало программ.

QtShar – это одна из попыток связать между собой библиотеки фреймворка Qt, написанные на языке программирования C++, и язык программирования C# с платформой .Net [12]. По задумке, это очень похоже на библиотеку GTK#, но, в отличие от GTK#, QtSharp и ее аналоги не получили широкой популярности, так как связывание нативных библиотек Qt и QML с платформой .Net является очень сложной задачей.

Библиотека SpaceVII основана на графической технологии OpenGL и библиотеке GLFW, являющейся надстройкой над OpenGL. В отличие от перечисленных библиотек, SpaceVII, помимо языков платформы .Net (C#, F#, VB.Net), поддерживает язык программирования Java [13]. Несмотря на это явное преимущество в сравнении с остальными библиотеками, SpaceVII не получила какого-либо дальнейшего развития. Последняя альфа-версия данной библиотеки была выпущена в 2020 году [14]. Таким образом, не имеет смысла ее подробно рассматривать.

В качестве вывода можно заявить, что у языка программирования C# и платформы .Net существует много кроссплатформенных библиотек для разработки графического интерфейса. У всех есть одна проблема – небольшая популярность, и это может затруднить разработку приложений. Если необходимо создать высоконагруженное приложение, чувствительное к ресурсам компьютера, лучше выбрать другой язык программирования (C/C++) и, соответственно, другие библиотеки (Qt, GTK). Тем не менее при

разработке простых приложений и решении какой-либо прикладной бизнес-задачи возможностей таких библиотек хватит. Самыми перспективными кроссплатформенными библиотеками GUI для C#.Net являются Avalonia, MAUI и UNO.

Библиографический список

1. Avalonia UI – Open-Source .NET XAML Framework. URL: <https://avaloniaui.net/> (дата обращения: 14.10.2025).
2. Uno Platform: Build Cross-Platform .NET Apps Faster. URL: <https://platform.uno/> (дата обращения: 14.10.2025).
3. Cross-platform GUIs with Open Source .NET Using Eto.Forms // Scott Hanselman’s Blog. URL: <https://www.hanselman.com/blog/crossplatform-guis-with-open-source-net-using-etofrms> (дата обращения: 14.10.2025).
4. Попов Е. GTK и C | Введение в GTK. Первое приложение. URL: <https://metanit.com/sharp/gtk/1.1.php> (дата обращения: 14.10.2025).
5. Подробно о Xamarin. URL: <https://habr.com/ru/articles/188130/?ysclid=mgp5heowc517035949> (дата обращения: 14.10.2025).
6. Что такое .NET MAUI. URL: <https://learn.microsoft.com/ru-ru/dotnet/maui/what-is-maui?view=net-maui-9.0> (дата обращения: 14.10.2025).
7. Попов Е. Blazor | Что такое Blazor. URL: <https://metanit.com/sharp/blazor/1.1.php> (дата обращения: 14.10.2025).
8. Пишем GUI на .NET для Linux. URL: <https://www.calabonga.net/blog/post/dotnet-gui-for-linux> (дата обращения: 14.10.2025).
9. Cross-Platform C# UI Technologies. URL: <https://itbinc.io/blog/posts/2019/august/cross-platform-c-ui-technologies/> (дата обращения: 14.10.2025).
10. Unity Documentation. URL: <https://docs.unity.com/en-us?ref=jinxxy> (дата обращения: 14.10.2025).
11. Документация Godot Engine (4.x) на русском языке. URL: <https://godot-ru.readthedocs.io/ru/4.x/> (дата обращения: 14.10.2025).
12. Репозиторий проекта QtSharp. URL: <https://gitlab.com/ddobrev/QtSharp?ysclid=mgzdjqgyuw844469142> (дата обращения: 14.10.2025).
13. SpaceVIL – кроссплатформенный GUI фреймворк для разработки на .Net Core, .Net Standard и JVM. URL: <https://habr.com/ru/articles/448790/> (дата обращения: 14.10.2025).
14. SpaceVessel. URL: <https://spvessel.com/> (дата обращения: 14.10.2025).

OVERVIEW OF CROSS-PLATFORM LIBRARIES FOR BUILDING A GRAPHICAL INTERFACE IN THE C# PROGRAMMING LANGUAGE

I.T. Novikov, V.V. Lebedev

***Abstract.** The article discusses libraries for building a cross-platform graphical interface (GUI) in the C# programming language within the .Net software platform.*

***Keywords:** GUI, C#, GUI libraries, cross-platform.*

Об авторах:

НОВИКОВ Илья Тимурович – магистрант, ФГБОУ ВО «Тверской государственный технический университет», Тверь. E-mail: ilyanovickov@mail.ru

ЛЕБЕДЕВ Владимир Владимирович – кандидат технических наук, доцент, профессор кафедры электронных вычислительных машин, ФГБОУ ВО «Тверской государственный технический университет», Тверь. E-mail: Lebedev_vl69@mail.ru

About the authors:

NOVIKOV Ilya Timurovich – Master's Student, Tver State Technical University, Tver. E-mail: ilyanovickov@mail.ru

LEBEDEV Vladimir Vladimirovich – Candidate of Technical Sciences, Associate Professor, Professor of the Department of Electronic Computing Machines, Tver State Technical University, Tver. E-mail: Lebedev_vl69@mail.ru

УДК 338.1

ОБЗОР МЕТОДОЛОГИЙ ОЦЕНКИ УРОВНЯ ЦИФРОВОЙ ЗРЕЛОСТИ ТОРФОДОБЫВАЮЩЕГО ПРЕДПРИЯТИЯ

Т.Б. Яконовская

© Яконовская Т.Б., 2026

***Аннотация.** В статье рассмотрены методологические подходы к оценке цифровой зрелости торфодобывающего предприятия. Отмечено, что оценка цифровой зрелости подразумевает выявление степени внедрения современных информационных технологий и цифрового*