

WPF DA ANIMATSIYALARDAN FOYDALANISH

Yusupov Mirsaid Abdulaziz o'gli*Farg'ona davlat universiteti kafedra o'qituvchisi**mirsaidbeka@gmail.com***Isaqova Gulasal Ahmadjon qizi***Farg'ona davlat universiteti 2-kurs talabasi**fbffbf475@gmail.com*

Annotatsiya: Mazkur ilmiy maqolada zamonaviy grafik interfeyslarni yaratishda keng qo'llanilayotgan **WPF (Windows Presentation Foundation)** texnologiyasida **animatsiyalardan foydalanish imkoniyatlari** tahlil qilinadi. Unda WPF tizimida harakatlanuvchi obyektlar, vizual o'zgarishlar va foydalanuvchi tajribasini oshirishga xizmat qiluvchi animatsiya elementlari dasturlashdagi o'rni yoritilgan. Shuningdek, maqolada **Storyboard**, **DoubleAnimation**, **ColorAnimation** kabi asosiy animatsion komponentlarning ishlash prinsiplari va amaliy misollar asosida ularni amalga oshirish usullari keltirilgan. Dasturchilarga interaktiv interfeyslar yaratishda animatsiyaning foydalari, samaradorlikka ta'siri va dizayn yondashuvlari batafsil yoritiladi.

Maqolada shuningdek, WPF arxitekturasida **XAML** va **C# tilida animatsiyalarni kodlash**, real vaqt rejimida animatsiyalarni boshqarish, shartli animatsiyalar va foydalanuvchi harakatlariga javob beruvchi animatsion funksiyalar ko'rib chiqiladi. Tadqiqot natijalarida, animatsiyalarning foydalanuvchining dastur bilan ishlash sifatini oshirish, ma'lumotlarni vizual tarzda yetkazish va dastur dizaynini yaxshilashdagi roli asoslab berilgan.

Ushbu maqola WPF platformasida grafik interfeys ishlab chiquvchilar, axborot texnologiyalari sohasida tahsil olayotgan talaba va ilmiy izlanuvchilar uchun foydali manba bo'lib xizmat qiladi.

Kalit so'zlar: WPF, animatsiya, Storyboard, DoubleAnimation, XAML, C#, foydalanuvchi interfeysi, grafik dasturlash, dizayn, interaktivlik, vizualizatsiya, harakat effekti, dasturiy ta'minot, foydalanuvchi tajribasi.

Аннотация: В данной научной статье анализируются возможности использования анимации в технологии WPF (Windows Presentation Foundation), которая широко применяется при создании современных графических интерфейсов. В нем рассматривается роль анимированных объектов, визуальных изменений и элементов анимации в программировании в системе WPF для улучшения пользовательского опыта. В статье также представлены принципы работы основных компонентов анимации, таких как Storyboard, DoubleAnimation и ColorAnimation, и методы их реализации на практических примерах. Разработчикам будет предоставлен подробный обзор преимуществ

анимации, ее влияния на производительность и подходов к проектированию при создании интерактивных интерфейсов.

В статье также рассматривается кодирование анимации на XAML и C# в архитектуре WPF, управление анимацией в реальном времени, условная анимация и функции анимации, реагирующие на действия пользователя. Результаты исследования подтверждают роль анимации в улучшении пользовательского опыта, визуальной передаче информации и улучшении дизайна приложений.

Статья будет полезна разработчикам графических интерфейсов на платформе WPF, студентам, обучающимся в сфере информационных технологий, и исследователям.

Ключевые слова: WPF, анимация, Storyboard, DoubleAnimation, XAML, C#, пользовательский интерфейс, графическое программирование, дизайн, интерактивность, визуализация, эффекты движения, программное обеспечение, пользовательский опыт.

Annotatsiya: This scientific article analyzes the possibilities of using animations in WPF (Windows Presentation Foundation) technology, which is widely used in creating modern graphical interfaces. It highlights the role of animation elements in programming, which serve to improve the user experience, moving objects, visual changes in the WPF system. The article also presents the principles of operation of such basic animation components as Storyboard, DoubleAnimation, ColorAnimation and methods of their implementation based on practical examples. The benefits of animation, its impact on efficiency and design approaches for programmers when creating interactive interfaces are explained in detail.

The article also considers animation coding in XAML and C# within the WPF architecture, real-time animation control, conditional animations and animation functions that respond to user actions. The research results substantiate the role of animations in improving the quality of user interaction with the application, visually conveying information and improving application design.

This article will serve as a useful resource for developers of graphical interfaces on the WPF platform, students studying in the field of information technology, and researchers.

Keywords: WPF, animation, Storyboard, DoubleAnimation, XAML, C#, user interface, graphic programming, design, interactivity, visualization, motion effects, software, user experience.

KIRISH

Axborot texnologiyalarining jadal sur'atlar bilan rivojlanib borayotgani dasturiy ta'minot ishlab chiqishda ham yangi yondashuvlarni talab qilmoqda. Ayniqsa, foydalanuvchiga qulay, zamonaviy va interaktiv grafik interfeyslarni yaratish

bugungi kundagi eng muhim vazifalardan biriga aylangan. Bu borada Microsoft tomonidan taklif etilgan **Windows Presentation Foundation (WPF)** texnologiyasi dasturchilar uchun keng imkoniyatlar yaratadi. Aynan WPF texnologiyasining kuchli tomonlaridan biri bu — **animatsiyalardan keng va samarali foydalanish imkoniyati** hisoblanadi. Animatsiyalar foydalanuvchi interfeysining jonli, chiroyli va interaktiv bo'lishiga xizmat qiladi. Oddiy ilovalardan tortib murakkab tizimli dasturlargacha, har qanday vizual elementni harakatga keltirish, rangini o'zgartirish, ko'rinishini asta-sekin chiqarish yoki yo'qotish, shuningdek foydalanuvchi harakatiga javoban animatsion javoblarni shakllantirish bugungi interfeys dizaynining ajralmas qismiga aylangan. WPF bu imkoniyatlarni yaratishda **XAML (eXtensible Application Markup Language)** va **C#** dasturlash tillarining sinergiyasiga asoslanadi. Mazkur ilmiy maqolada WPF texnologiyasida animatsiyalardan foydalanishning nazariy asoslari, amaliy qo'llanish holatlari, texnik yechimlari va foydalari yoritiladi. Shu bilan birga, WPF platformasida eng ko'p ishlatiladigan animatsion komponentlar — **Storyboard**, **DoubleAnimation**, **ColorAnimation**, **ThicknessAnimation** kabi sinflarning ishlash mexanizmlari, ularni kodlash usullari va XAML muhiti orqali integratsiyalashgan ishlanmalari tahlil qilinadi.

Bugungi dasturlash jarayonlarida animatsiyalar foydalanuvchiga intuitiv interfeysni taqdim etish bilan birga, dasturning dizayn sifati, foydalanish qulayligi va hatto marketing salohiyatini ham oshiradi. Shu boisdan, animatsion imkoniyatlarning chuqur o'rganilishi dasturiy mahsulotlarni samarali loyihalash va amaliyotga tatbiq etishda muhim rol o'ynaydi. Maqolada shuningdek, **real vaqt rejimida animatsiyalarni boshqarish, tugma bosilishi, sichqoncha harakati yoki boshqa foydalanuvchi amallariga javoban avtomatik animatsiyalarni qo'llash**, shartli animatsiyalarni yaratish kabi ilg'or yondashuvlar ham batafsil ko'rib chiqiladi. Ushbu yondashuvlar nafaqat vizual go'zallikni ta'minlaydi, balki foydalanuvchi bilan dastur o'rtasidagi muloqotni yanada samarali va qulay qiladi.

NATIJALAR

O'tkazilgan tadqiqotlar va WPF asosida yaratilgan test loyihalar asosida quyidagi muhim natijalarga erishildi:

WPF (Windows Presentation Foundation) texnologiyasi animatsiyalarni amalga oshirishda yuqori darajadagi qulaylik va moslashuvchanlikni ta'minlaydi. Xususan, XAML orqali deklarativ usulda animatsiya yozish imkoniyati dasturchilarga interfeys komponentlarini sodda va tushunarli shaklda harakatlantirish, o'zgartirish va boshqarish imkonini beradi. Bu yondashuv nafaqat samaradorlikni oshiradi, balki dasturiy ta'minotning vizual ko'rinishini ham sezilarli darajada yaxshilaydi.

Tadqiqot davomida **Storyboard** sinfining animatsiyalarni guruhlash va ularni sinxronlashtirib boshqarishdagi roli aniqlandi. Ayniqsa, foydalanuvchi harakatlariga bog'liq bo'lgan hodisaviy animatsiyalar (masalan, sichqoncha bosilishi yoki kursor

ustiga olib borilishi) foydalanuvchi interfeysini sezilarli darajada jonlantiradi. Bu esa, interaktivlikni oshirish bilan birga, dasturdan foydalanish qulayligini ta'minlaydi.

DoubleAnimation, ColorAnimation, ThicknessAnimation kabi sinflarning qo'llanilishi orqali foydalanuvchi interfeysining har qanday xususiyatlarini – rang, hajm, pozitsiya, zichlik kabi parametrlarini osonlik bilan o'zgartirish va animatsiyalash imkoniyati mavjudligi isbotlandi. Ayniqsa, ushbu animatsiyalarni vaqt oralig'ida (Duration) va boshlanish kechikishi (BeginTime) bilan sozlash orqali murakkab vizual effektlarni yaratish mumkinligi amaliyotda tasdiqlandi.

Eksperimental tajribalar davomida yaratilgan kichik WPF ilovalarda animatsiyalarni qo'llash orqali foydalanuvchi tajribasi (UX) oshganligi, ilova dizayni esa professional darajaga ko'tarilganligi kuzatildi. Bu animatsiyalarning axborotni tushunarli yetkazish, foydalanuvchining e'tiborini kerakli joyga jalb qilish va harakatlar oqibatlarini ko'rsatishda muhim rol o'ynashini isbotlaydi.

Tahlillar shuni ko'rsatdiki, **kod orqali (C#) animatsiyani yaratish** WPFdagi deklarativ (XAML) usulga nisbatan yanada dinamikroq va foydalanuvchi harakatlariga moslashuvchanroq animatsiyalar yaratishga imkon beradi. Bu ayniqsa murakkab holatlarda, ya'ni foydalanuvchi kiritmalariga asosan avtomatik animatsiyalarni ishlab chiqishda o'z aksini topdi.

WPF tizimida animatsiyalar resurslar asosida (ResourceDictionary) markazlashtirilgan holda yaratilishi mumkinligi aniqlanib, bu usul yordamida kod soddalashadi, animatsiyalar qayta ishlatiladi va interfeys elementlarining bir xilligi ta'minlanadi.

Shuningdek, dastur samaradorligini oshirish maqsadida **Hardware Rendering** va **GPU tezlashtirish** imkoniyatlaridan foydalanilgan WPF animatsiyalarida tizim yuklamasining kamaygani, animatsiyalarning silliqqligi oshgani va ishlash tezligi yaxshilanganligi amalda tasdiqlandi.

MUHOKAMA

WPF texnologiyasi o'zining interfeysni qurishdagi keng imkoniyatlari bilan zamonaviy dasturlash muhitida o'ziga xos o'ringa ega. Ayniqsa, animatsiyalar orqali vizual dinamizm yaratish imkoniyati ushbu platformani boshqa texnologiyalardan ajratib turadi. Animatsiya elementlarining dasturdagi funksional roli faqat ko'rgazmali ta'sirchanlik bilan cheklanmay, balki ma'lumotni yetkazish samaradorligini oshirishga ham xizmat qiladi. Animatsiyalarning asosiy maqsadi — foydalanuvchi bilan dastur o'rtasida jonli aloqa o'rnatish, ya'ni dasturning holatini yoki foydalanuvchi harakatlariga javoban sezilarli o'zgarishlarni ko'rsatib berishdir. Masalan, tugmani bosganda u rangini o'zgartirishi yoki siljishi, bu foydalanuvchiga "javob" bo'ladi. Bunday oddiy ko'rinishdagi animatsiyalar ham psixologik qulaylik yaratadi.

WPF da eng ko'p qo'llaniladigan animatsiya usullari orasida **DoubleAnimation, ColorAnimation, ThicknessAnimation** va **PointAnimation**

bor. Har biri o'ziga xos xususiyatga ega bo'lib, tegishli parametrlar ustida vaqt mobaynida o'zgarish kiritadi. Misol uchun, DoubleAnimation yordamida obyektning eni, bo'yi, koordinatasi yoki shaffofligi animatsiyalanadi. XAML kodida animatsiya yozish deklarativ yondashuv orqali amalga oshiriladi. Bu dasturchiga vizual elementlar va ularning xatti-harakatlarini o'zaro bog'langan holda, aniq va strukturalangan tarzda yaratish imkonini beradi. Storyboard komponenti bir nechta animatsiyalarni bir vaqtning o'zida ishga tushirishga xizmat qiladi.

Muhokama jarayonida aniqlanishicha, **WPF animatsiyalarining samarali ishlashi** uchun ular ilovaning umumiy dizayn konsepsiyasiga uyg'un bo'lishi lozim. Ya'ni, ortiqcha animatsiyalar foydalanuvchini chalg'itadi, keragidan ortiq vizual effektlar esa ilova ish faoliyatini sekinlashtiradi. Shu sababli, har bir animatsiya aniq funktsional vazifani bajarmog'i kerak. C# tilida animatsiyalarni dinamik tarzda yaratish orqali foydalanuvchining real vaqtdagi harakatlariga moslashuvchan tizimlar ishlab chiqish mumkin. Masalan, foydalanuvchi biror obyekt ustiga olib borganida, u o'z o'lchamini avtomatik o'zgartirib turishi yoki boshqa animatsion reaksiyalarni bajarsa, bu interfeysni "jonli" qiladi. Ushbu tadqiqotda shuningdek **harakatlanuvchi vizual obyektlar** bilan ishlashda aniqlik va aniqlik bilan bog'liq texnik muammolar ham ko'rib chiqildi. Ayniqsa, murakkab animatsiyalarda kadrlarni sinxronlashtirish va resurslar yuklamasini hisobga olish zarurati mavjud. Bu borada RenderTransform va BeginStoryboard kabi vositalar muhim ahamiyatga ega. Tajribaviy ishlanmalar davomida ma'lum bo'ldiki, foydalanuvchi interfeysidagi kichik animatsiyalar ham ularning dasturga bo'lgan ishonchini oshiradi. Bu animatsiyalar foydalanuvchini dasturni boshqarishga qiziqtiradi, shuningdek, foydalanuvchiga qanday amal bajarilganini tushunarli tarzda ko'rsatadi. Muhokama jarayonida shuningdek EventTrigger, DataTrigger va VisualStateManager orqali animatsiyalarni holatga qarab boshqarish mexanizmlari ham tahlil qilindi. Bu usullar yordamida murakkab interaktiv holatlarni soddalashtirish mumkin bo'ladi. WPF animatsiyalari ayniqsa **ta'limiy dasturlar, multimedia ilovalari va korporativ ma'lumot platformalari** uchun katta qulayliklar yaratadi. Misol uchun, o'qituvchilar uchun mo'ljallangan ilovalarda tushunchalarni animatsion tarzda ko'rsatish o'quvchilarning e'tiborini jalb qiladi va mavzuni chuqurroq anglashlariga yordam beradi.

Muhokamada, EasingFunction turlari orqali animatsiya harakatini tabiiyroq ko'rsatish imkoniyatlari ham ko'rib chiqildi. Ular yordamida harakat tezligini boshlanishi va tugashida silliq qilish, shok effekti berish yoki orqaga sakratish effektlarini yaratish mumkin. Interfeysda animatsiyalardan foydalanganda ular ortidan **ma'lumotlar xavfsizligi, javob berish tezligi va moslashtiriluvchanlik** kabi jihatlar ham hisobga olinishi kerak. Bu parametrlar foydalanuvchi tajribasini optimallashtirishda muhim o'rin tutadi. Texnologik jihatdan, WPF animatsiyalari **GPU tezlashtirish** asosida ishlaydi, bu esa vizual harakatlarning silliqligini ta'minlaydi. Biroq, noto'g'ri optimallashtirilgan animatsiyalar grafik tizim yukini

oshirib, umumiy ishlash tezligiga salbiy ta'sir ko'rsatishi mumkin. Muhokama davomida resurslar orqali (ResourceDictionary) animatsiyalarni bir marta yozib, ularni bir nechta sahifalarda qayta ishlatish tajribalari ham ko'rib chiqildi. Bu yondashuv kodni ixchamlashtiradi va uni boshqarishni osonlashtiradi. Shuningdek, foydalanuvchining individual tajribasini oshirish maqsadida personalizatsiya qilingan animatsiyalarni qo'llash mumkinligi aniqlandi. Masalan, foydalanuvchi kirgan sahifaga qarab, o'ziga xos animatsiyalar ko'rsatilsa, u ilovada o'zini ko'proq qulay his qiladi.

```
<Window x:Class="WpfAnimationDemo.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Animatsiya Misoli" Height="350" Width="525">
    <Grid>
        <Button Name="MyButton" Width="100" Height="40" Content="Bos!"
HorizontalAlignment="Left"    Margin="10,150,0,0"    VerticalAlignment="Top"
Click="MyButton_Click">
            <Button.RenderTransform>
                <TranslateTransform x:Name="buttonTransform"/>
            </Button.RenderTransform>
        </Button>
    </Grid>
</Window>
```

```
using System.Windows;
using System.Windows.Media.Animation;
using System.Windows.Media;
```

```
namespace WpfAnimationDemo
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void MyButton_Click(object sender, RoutedEventArgs e)
        {
            // Harakat animatsiyasi (gorizontal o'zgarish)
            DoubleAnimation moveAnimation = new DoubleAnimation();
```

```

moveAnimation.From = 0;
moveAnimation.To = 300;
moveAnimation.Duration = new Duration(TimeSpan.FromSeconds(1));
moveAnimation.AutoReverse = true;

```

```

buttonTransform.BeginAnimation(TranslateTransform.XProperty,
moveAnimation);

```

```

// Rangi o'zgaruvchi animatsiya
ColorAnimation colorAnimation = new ColorAnimation();
colorAnimation.From = Colors.LightBlue;
colorAnimation.To = Colors.OrangeRed;
colorAnimation.Duration = new Duration(TimeSpan.FromSeconds(1));
colorAnimation.AutoReverse = true;

```

```

MyButton.Background = new SolidColorBrush(Colors.LightBlue);

```

```

MyButton.Background.BeginAnimation(SolidColorBrush.ColorProperty,
colorAnimation);

```

```

    }

```

```

    }

```

```

}

```

Natijasi

1. Button bosilganda u **chapdan o'ngga harakat qiladi** (300 pikselgacha).
2. Tugma fon rangi **LightBlue'dan OrangeRed'gacha** o'zgaradi.
3. Harakat va rang **avtomatik orqaga qaytadi** (AutoReverse)

XULOSA

WPF (Windows Presentation Foundation) texnologiyasi zamonaviy grafik interfeyslarni yaratishda keng imkoniyatlarga ega bo'lib, ayniqsa animatsiyalardan foydalanish orqali foydalanuvchi interfeysini jonlantirish, vizual estetikani oshirish va foydalanuvchi tajribasini (UX) yaxshilashda beqiyos vositadir. Mazkur ilmiy maqolada olib borilgan nazariy tahlillar va amaliy tadqiqotlar asosida aniqlanishicha, WPF animatsiyalari dasturiy mahsulotlarga sezilarli funksional va vizual afzalliklar olib keladi.

WPF da Storyboard, DoubleAnimation, ColorAnimation, ThicknessAnimation, EasingFunction kabi komponentlar yordamida turli animatsion yechimlar sodda, ammo samarali tarzda amalga oshirilishi mumkin. Bu vositalar foydalanuvchining harakatlariga moslashuvchan tarzda javob beruvchi, silliq va jozibali interfeyslar

yaratishga xizmat qiladi. Ayniqsa XAML va C# tili orqali animatsiyalarni kombinatsiyalash, real vaqtli boshqaruv, resurslar orqali animatsiyalarni umumlashtirish — ularning barchasi zamonaviy WPF dasturlarining ajralmas tarkibiy qismlaridir.

ADABIYOTLAR RO‘YXATI:

1. Petzold, C. Applications = Code + Markup: A Guide to the Microsoft Windows Presentation Foundation. Microsoft Press, 2006. – 528 b.
2. Nathan, A. WPF Unleashed. Sams Publishing, 2008. – 744 b.
3. MacDonald, M. Pro WPF in C# 2008: Windows Presentation Foundation with .NET 3.5. Apress, 2008. – 1040 b.
4. Агапитов, И.В. WPF и XAML: Разработка современных интерфейсов. СПб: БХВ-Петербург, 2014. – 384 б.
5. MSDN Library – <https://docs.microsoft.com/en-us/dotnet/desktop/wpf>
6. Albahari, J., & Albahari, B. C# 9.0 in a Nutshell. O'Reilly Media, 2021.
7. Бондаренко, Е.В. WPF на практике: пошаговое руководство. – М.: ДМК Пресс, 2015. – 320 б.
8. Официальная документация Microsoft – <https://learn.microsoft.com>
9. Josh Smith, Advanced MVVM Techniques with WPF, 2012.
10. UX Planet: The Power of UI Animation in Modern Applications, <https://uxplanet.org>
11. Karimberdiyevich, O. M., & Abdulaziz o'g'li, Y. M. (2024). SUN'IY INTELEKTNING AFZALLIKLARI VA KAMCHILIKLARI. IZLANUVCHI, 1(1), 75-85.
12. Karimberdiyevich, O. M., & Abdulaziz o'g'li, Y. M. (2024). NEYRO KOMPYUTERLAR. YANGI O'ZBEKISTON, YANGI TADQIQOTLAR JURNALI, 1(5), 19-27.
13. Karimberdiyevich, O. M., & Abdulaziz o'g'li, Y. M. (2024). K-YAQIN QO'SHNI ALGORITMI. IZLANUVCHI, 1(1), 122-124.
14. Abdulaziz o'g'li, Y. M. (2025). WPFDA ANIMATSIYA YARATISHNI QO'LLANISHI. MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS, 1(4), 172-175.
15. Abdulaziz o'g'li, Y. M. (2025). MOLIYA VA HISOB-KITOB ILOVALARIDA WPF BILAN ISHLASH. MODERN PROBLEMS IN EDUCATION AND THEIR SCIENTIFIC SOLUTIONS, 1(4), 189-193.