UDK. 004.514.62

WINDOWS FORMS .NET ILOVALARNING DIZAYNIDA UI/UX FRAMEWORKLARINING AFZALLIKLARI

Buriyev Javoxir Nosir o‘g‘li. *o‘qituvchi, Termiz davlat universiteti, Termiz*

ADVANTAGES OF UI/UX FRAMEWORKS IN DESIGNING WINDOWS FORMS .NET APPLICATIONS

Buriev Javokhir Nosir o‘g‘li. *lecturer, Termez state university, Termez*

Annotatsiya: .Net texnologiyasida Windows Forms ilovalarining interfeyslarini yaratishda, loyihalashda, dizaynida UI/UX freymvorklarning o’rni katta bo‘lib, ular orqali yaratilayotgan ilovaning dizaynini sodda loyihalash mumkin.

Kalit so‘zlar: .Net texnologiyasi, freymvorklar, UI/UX, dizayn, button, boshqaruv elementlari, GUNA, Bunifu.

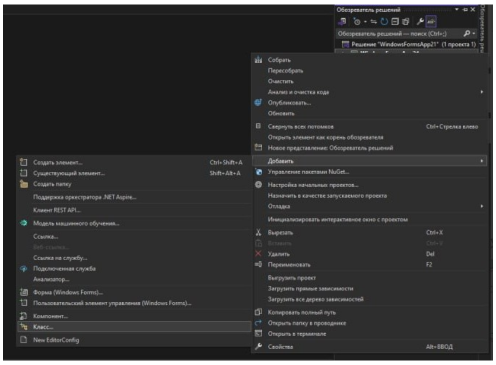
Аннотатсия: Фреймворки UI/UX играют важную роль в создании, проектировании и проектировании интерфейсов приложений Windows Forms в технологии .Net, и с их помощью можно легко спроектировать дизайн создаваемого приложения.

Ключевые слова: технология .Net, фреймворки, UI/UX, дизайн, кнопка, элементы управления, GUNA, Bunifu.

Annotation: UI/UX frameworks play a major role in creating, designing and designing the interfaces of Windows Forms applications in .Net technology and with the help of them one can easily design the design of the application being created.

Key words: .Net technology, frameworks, UI/UX, design, button, control elements, GUNA, Bunifu.

C# dasturlash tili asosiga qurilgan .Net texnologiyalarida Windows Form ilovalarini yaratish orqali ko‘plab sohalarda mavjud masalalarni yechuvchi dasturlar va web ilovalar ommalashib bormoqda. Bu texnologiyalar orqali ko’plab boshqarish tizimlari, hisob-kitob tizimlari, ilovalar yaratish imkoni mavjud. Bu esa o‘z navbatida ilovalarning dizaynini yaratishga bir qancha shartlar qo‘ymoqda. Dasturchilar Winform ilovalarning dizaynini yaratishda standart boshqaruv elementlaridan foydalanishi va ularga ishlov berishida ko’plab qiyinchiliklar yuzaga kelishi mumkin. Shuningdek bu vazifalar dasturchidan ko’plab vaqt talab qilishi mumkin. Masalan birgina rangi, shakli, border radiusi kabi ko’plab xususiyatlari o‘zgaruvchan bo’lgan Button elementini yaratish uchun quyidagi birqancha amallar ketma- ketligini bajarish zarur hisoblanadi. Ishni avvalo loyihaga yangi class qo’shishdan boshlashimiz zarur:


Обозреватель решений
Обозреватель решений — поиск (Ctrl+;)
Решение "WindowsFormsApp21" (1 проекта 1)
Собрать
Пересобрать
Очистить
Анализ и очистка кода
Опубликовать...
Обновить
Свернуть всех потомков Ctrl+Стрелка влево
Открыть элемент как корень обозревателя
Новое представление: Обозреватель решений
Добавить
Управление пакетами NuGet...
Настройка начальных проектов...
Назначить в качестве запускаемого проекта
Отладка
Инициализировать интерактивное окно с проектом
Вырезать Ctrl+X
Вставить Ctrl+V
Удалить Del
Переименовать F2
Выгрузить проект
Загрузить прямые зависимости
Загрузить все дерево зависимостей
Копировать полный путь
Открыть папку в проводнике
Открыть в терминале
Свойства Alt+ВВОД
Создать элемент... Ctrl+Shift+A
Существующий элемент... Shift+Alt+A
Создать папку
Поддержка оркестратора .NET Aspire...
Клиент REST API...
Модель машинного обучения...
Ссылка...
Веб-ссылка...
Ссылка на службу...
Подключенная служба
Анализатор...
Форма (Windows Forms)...
Пользовательский элемент управления (Windows Forms)...
Компонент...
Класс...
New EditorConfig

1-rasm.

Bu yerda yaratmoqchi bo‘lgan elementimizga nom beramiz, ya’ni yaratilgan sinf nomi va element nomi bir xil bo‘ladi. Yaratgan sinfimiz ichiga quyidagi kodlarni yozib chiqamiz:

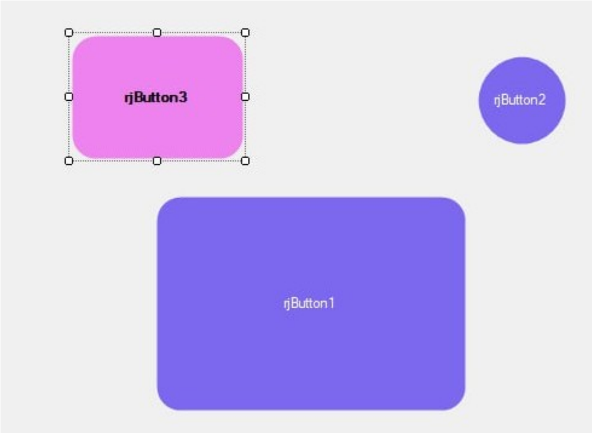
```
using System.Windows.Forms;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.ComponentModel;
namespace WindowsFormsApp21
{
   internal class RJButton:Button
   {
      private int borderSize = 0;
      private int borderRadius = 40;
      private Color borderColor = Color.PaleVioletRed;
      [Category("RJ Code Advance")]
      public int BorderSize { get => borderSize; set { borderSize = value; this.Invalidate(); } }
      [Category("RJ Code Advance")]
      public int BorderRadius { get => borderRadius; set { borderRadius = value; this.Invalidate(); } }
      [Category("RJ Code Advance")]
      public Color BorderColor { get => borderColor; set { borderColor = value; this.Invalidate(); } }
      [Category("RJ Code Advance")]
      public Color BackgroundColor{get=>this.BackColor;set{this.BackColor=value;}}
      [Category("RJ Code Advance")]
      public Color TextColor {get=>this.ForeColor;set{this.ForeColor = value; } }
      public RJButton()
      {
         this.FlatStyle = FlatStyle.Flat;
         this.FlatAppearance.BorderSize = 0;
         this.Size = new Size(150, 40);
         this.BackColor = Color.MediumSlateBlue;
         this.ForeColor = Color.White;
      }
      private GraphicsPath GetFigurePath( RectangleF rect, float radius)
      {
         GraphicsPath path = new GraphicsPath();
         path.StartFigure();
         path.AddArc(rect.X, rect.Y, radius, radius, 180, 90);
         path.AddArc(rect.Width-radius, rect.Y, radius, radius, 270, 90);
         path.AddArc(rect.Width-radius, rect.Height-radius,radius,radius, 0, 90);
         path.AddArc(rect.X, rect.Height-radius, radius, radius, 90, 90);
         path.CloseFigure();
         return path;
      }
```

```
        protected override void OnPaint(PaintEventArgs pevent)
        {
           base.OnPaint(pevent);
           pevent.Graphics.SmoothingMode = SmoothingMode.AntiAlias;

           RectangleF rectSurface = new RectangleF(0, 0, this.Width, this.Height);
           RectangleF rectBorder =new RectangleF(1,1,this.Width-0.8F,this.Height-1);
           if(borderRadius > 2)
           {
            using(GraphicsPath pathSurface=GetFigurePath(rectSurface, borderRadius))
            using(GraphicsPath pathBorder=GetFigurePath(rectBorder,borderRadius-1F))
            using (Pen penSurfece = new Pen(this.Parent.BackColor,2))
            using (Pen penBorder = new Pen(borderColor, borderSize))
            {
                penBorder.Alignment = PenAlignment.Inset;
                this.Region = new Region(pathSurface);
                pevent.Graphics.DrawPath(penSurfece, pathSurface);
                if( borderSize >= 1)
                    pevent.Graphics.DrawPath(penBorder, pathBorder);
             }} else {
              this.Region = new Region(rectSurface);
              if (borderSize >= 1)
              {
                using ( Pen penBorder = new Pen(borderColor, borderSize))
                {
                  penBorder.Alignment = PenAlignment.Inset;
                  pevent.Graphics.DrawRectangle(penBorder, 0, 0, this.Width - 1, this.Height - 1);
                    }
                 }
             }
        }
        protected override void OnHandleCreated(EventArgs e)
        {
            base.OnHandleCreated(e);
            this.Parent.BackColorChanged+=newEventHandler(Container_BackColorChanged);
        }
        private void Container_BackColorChanged(object sender, EventArgs e)
        {
            if(this.DesignMode)
                this.Invalidate();
        }
    }
}
```

Bu yerda yangi Button obyekti yaratib olinib, uning BorderSize, BorderRadius, BorderColor, BackgroundColor, TextColor xossalarini o‘zgaruvchan qilib yozib chiqilgan.

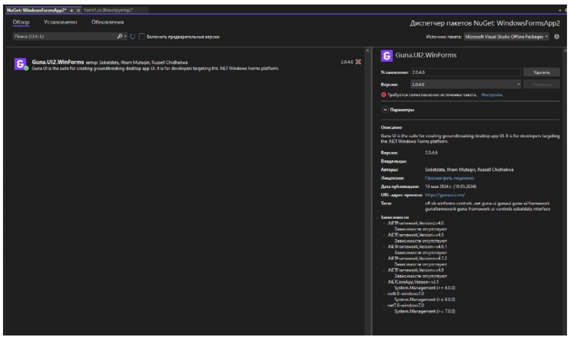
Shundan so‘ng loyihamizni build qilganimizda yangi elementimiz elementlar panelida paydo bo’ladi va biz undan foydalanishimiz mumkin.



2-rasm.

Ammo zamonaviy .Net texnologiyalari asosida ishlab chiqilgan va UI/UX dizayndagi muammolarni yechish imkoni sifatida ko'plab UI/UX dizayn freymvorklar yaratilgan. Bu dasturchiga qisqa vaqt sarflab, ajoyib dizayndagi ilovalar yaratish imkonini beradi. Bularga DevExpress, MetroFramework, Materialskin, Bunifu Framework, GUNA Framework kabilar misol bo'la oladi.

Guna Frameworkni ko'rib chiqadigan bo'lsak, bu frameworkni biz https://www.nuget.org/packages/Guna.UI2.WinForms/#versions-body-tab saytidan ustanovochniy paketini olamiz va C:\Program Files (x86)\Microsoft SDKs\NuGetPackages\nuget.frameworks\4.6.4 papkaga yuklangan paketimizni joylashtiramiz. Yangi winform loyihasini yaratamiz va u yerdan proyekt papkasi ustiga o'ng tugma orqali NuGet paketlarini boshqarish bo'limiga o'tamiz, ko'rsatish (browse) bo'limiga o'tamiz va qidiruv qismiga guna deb yozamiz. Paydo bo'lgan Guna.UI2.WinForms paketini o'rnatamiz (1-rasm).


NuGet: WindowsFormsApp2*
Form1.cs [Конструктор]*
Обзор
Установлено
Обновления
Диспетчер пакетов NuGet: WindowsFormsApp2
Поиск (Ctrl+L)
Включить предварительные версии
Источник пакета: Microsoft Visual Studio Offline Packages
Удалить
Обновить
Параметры

Guna.UI2.WinForms автор: Sobatdata, Ilham Mutaqin, Russell Chidhakwa 2.0.4.6
Guna UI is the suite for creating groundbreaking desktop app UI. It is for developers targeting the .NET Windows Forms platform.

Guna.UI2.WinForms

Установлено: 2.0.4.6

Версия: 2.0.4.6

Требуется сопоставление источника пакета. Настроить

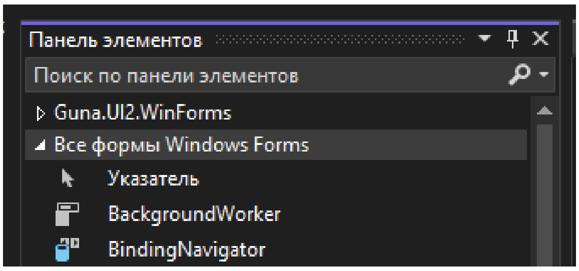
Описание
Guna UI is the suite for creating groundbreaking desktop app UI. It is for developers targeting the .NET Windows Forms platform.

Версия: 2.0.4.6
Владельцы:
Авторы: Sobatdata, Ilham Mutaqin, Russell Chidhakwa
Лицензия: Просмотреть лицензию
Дата публикации: 10 мая 2024 г. (10.05.2024)
URL-адрес проекта: https://gunaui.com/
Теги: c# vb winforms controls .net guna ui gunaui guna-ui framework gunaframework guna-framework ui-controls sobatdata interface

Зависимости
- .NETFramework,Version=v4.0
  - Зависимости отсутствуют
- .NETFramework,Version=v4.5
  - Зависимости отсутствуют
- .NETFramework,Version=v4.6.1
  - Зависимости отсутствуют
- .NETFramework,Version=v4.7.2
  - Зависимости отсутствуют
- .NETFramework,Version=v4.8
  - Зависимости отсутствуют
- .NETCoreApp,Version=v3.1
  - System.Management (>= 6.0.0)
- net6.0-windows7.0
  - System.Management (>= 6.0.0)
- net7.0-windows7.0
  - System.Management (>= 7.0.0)

3-rasm.

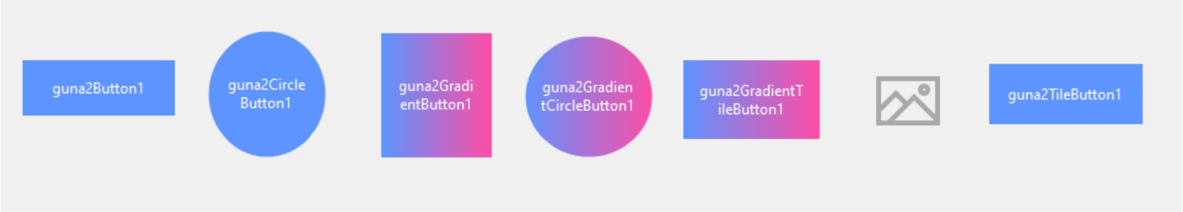
O'rnatilgandan so'ng elementlar panelini tekshirsak yangi guna elementlari hosil bo'lgfanini ko'rishimiz mumkin.



4-rasm.

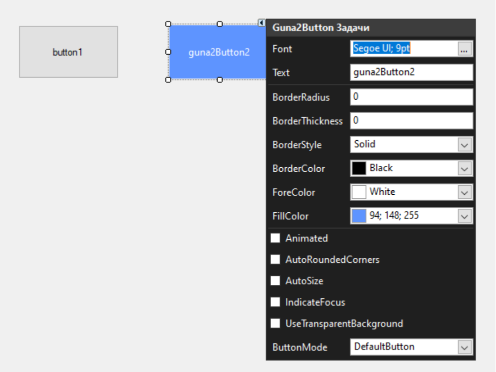
Hosil bo’lgan elementlardan foydalanishni boshlashimiz bilan bizga aktivatsiya oynasi ochiladi va bizdan litsenziya so’raydi. Biz bu freymvorkdan 14 kunlik sinov bersiyasi sifatida foydalanishimiz yoki to’lov qilib pulli versiyalarini sotib olishimiz mumkin. Bepul versiyani ustanovka qilish uchun trial tugmasini bosamiz va hosil bo’lgan oyna orqali ro’yxatdan o’tamiz.

Guna paketi orqali o’zimizning dizaynimizni yaratish imkoniga egamiz. Masalan birgina tugma(button)ning 7 xil turi mavjud. Bular: Guna2Button, Guna2CircleButton, Guna2GradientButton, Guna2GradientCircleButton, Guna2GradientTileButton, Guna2ImageButton, Guna 2TileButton (3-rasm).



5-rasm.

Oddiy button va Guna2Buttonning farqini quyida ko’rishimiz mumkin:



Guna2Button Задачи

Font: Segoe UI; 9pt
Text: guna2Button2
BorderRadius: 0
BorderThickness: 0
BorderStyle: Solid
BorderColor: Black
ForeColor: White
FillColor: 94; 148; 255
□ Animated
□ AutoRoundedCorners
□ AutoSize
□ IndicateFocus
□ UseTransparentBackground
ButtonMode: DefaultButton

6-rasm.

Guna orqali joylashtirilgan tugmaning ko’plab parametrlarini elementning yuqorisida joylashgan tugmani bosish orqali o’zgartirish mumkin. Bu esa ishning samaradorligini oshirish imkonini beradi.

Form oynasini ham o’z xohishingizga ko’ra qayta dizayn qilib chiqishingiz mumkin. Bunda boshqaruv oyna, menyular oynasi, asosiy oyna kabi o’zingizning dizayningizni loyihalashingiz mumkin.

Shunday qilib biz ikki xil usul yordamida o‘zimizning ilovalarimiz dizaynini ishlab chiqishni ko‘rdik. Ikkinchi usulimiz freymvork texnologiyalari orqali ishimiz sifati, tezligi va samaradorligini oshirish imkonini beradi.

Foydalanilgan adabiyotlar.

1. Microsoft_Visual_C_#_Step_by_Step, Implementing the user interface for a Universal Windows Platform app, Chapter25, 26, page 850.

2. Pro C# 10 with NET 6, Andrew Troelsen, Phil Japikse, Chapter 27, page 1157.
3. Fundamentals of Computer Programming with C# Svetlin Nakov, page 856.
4. https://www.nuget.org/
5. https://www.youtube.com/@gunaui4933
6. https://www.youtube.com/@amstudio5222
7. https://github.com/mulyawansentosa/Guna.UI-Framework-Lib