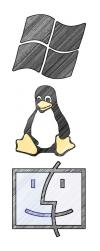
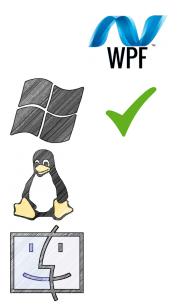
Avaloniau!

Cross-platform
Open source
UI Framework core!





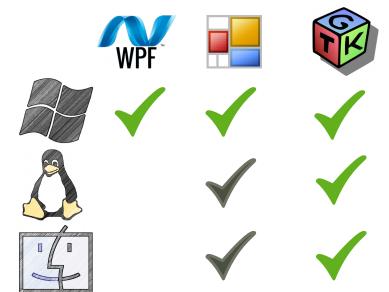








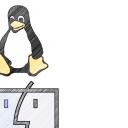






















































































































































КАК МНОЖАТСЯ СТАНДАРТЫ: (См.: ЗАРЯДНЫЕ УСТРОЙСТВА, КОДИРОВКИ, МГНОВЕННЫЕ СООБЩЕНИЯ И Т.Д.)

CUTYALLUA: ECT6 14 КОНКУРИРУЮЩИХ CTAHLAPTOB.



CKOPO

CNTYAUNA: ECT6 15 КОНКУРИРУЮЩИХ CTAHLAPTOB.

































































































































































































































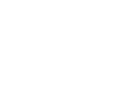














































В чём проблемы обёрток над нативными контролами?

В чём проблемы обёрток над нативными контролами?

В чём проблемы обёрток над нативными контролами?

• Приведение к платформ общему знаменателю

В чём проблемы обёрток над нативными контролами?

• Приведение к платформ общему знаменателю

• Сложность разработки UI с богатой стилизацией по макету от дизайнера

• **Система привязок** (обходится XForms, Eto.Forms, MvvmCross)

• **Система привязок** (обходится XForms, Eto.Forms, MvvmCross)

• Человекочитаемая разметка, в которой можно верстать (обходится XForms/Eto.Forms, так же нечто похожее есть в Android)

• **Система привязок** (обходится XForms, Eto.Forms, MvvmCross)

• Человекочитаемая разметка, в которой можно верстать (обходится XForms/Eto.Forms, так же нечто похожее есть в Android)

• Структура контролов на шаблонах

I'm a button!



- △ □ [Button]

 ☐
 - - ∠ IF contentPresenter [ContentPresenter]

```
<Border
  BorderBrush="{TemplateBinding BorderBrush}"
  BorderThickness="{TemplateBinding BorderThickness}"
  Background="{TemplateBinding Background}" ...>
  <ContentPresenter</pre>
       ContentTemplate="{TemplateBinding ContentTemplate}
       Content="{TemplateBinding Content}" .../>
</Border>
```

```
<ControlTemplate.Triggers>
  <Trigger Property="IsMouseOver" Value="True">
    <Setter Property="Background" TargetName="border" Value="#FFBEE6|</pre>
    <Setter Property="BorderBrush" TargetName="border" Value="#FF3C7I</pre>
  </Trigger>
  <Trigger Property="IsPressed" Value="True">
    <Setter Property="Background" TargetName="border" Value="#FFC4E5I</pre>
    <Setter Property="BorderBrush" TargetName="border" Value="#FF2C62</pre>
  </Trigger>
  <Trigger Property="IsEnabled" Value="False">
    <Setter Property="Background" TargetName="border" Value="#FFF4F4F</pre>
    <Setter Property="BorderBrush" TargetName="border" Value="#FFADB2</pre>
    <Setter Property="TextElement.Foreground" TargetName="contentPres</pre>
  </Trigger>
</ControlTemplate.Triggers>
```

Преимущества контролов без внешнего вида (lookless)

• Отделение логики работы и свойств контрола от его представления

• Возможность менять вид контрола практически *любым* образом по месту использования

ДЕМО

Чего мы хотим от UI-фреймворка?

- Все плюшки WPF
- Переносимость
- Open-source
- Орен 55.
 100% managed-код
 - Рендеринг «пиксель-в-пиксель» *везде*
 - Использование новых фич языка и экосистемы

29

AvaloniaUI

(netstandard1.1)

- биндинги
- logical/visual tree
- рендерер(ы)
- контролы
- прочее

30

AvaloniaUI

(netstandard1.1)

- биндинги
- logical/visual tree
- рендерер(ы)
- контролы
- прочее

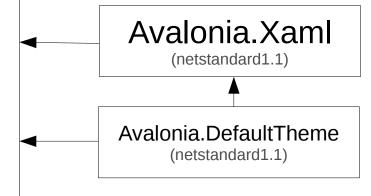
Avalonia.Xaml

(netstandard1.1)

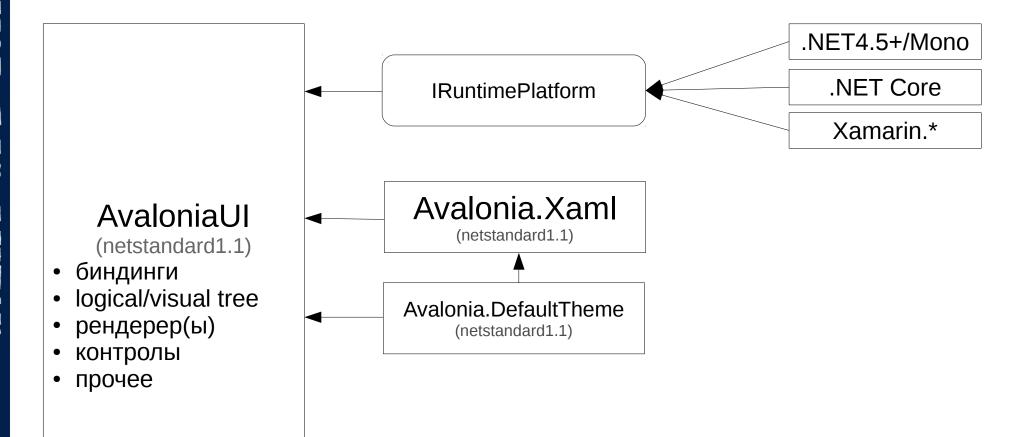
AvaloniaUI

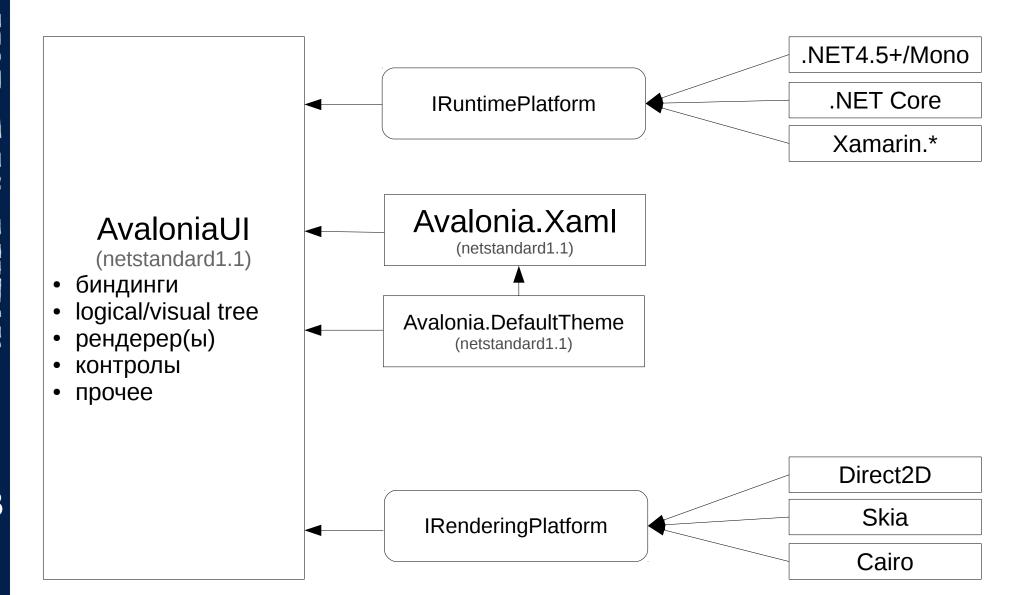
(netstandard1.1)

- биндинги
- logical/visual tree
- рендерер(ы)
- контролы
- прочее









IDrawingContextImpl

```
Matrix Transform { get; set; }
Matrix Transform { get; set; }
void Clear(Color color);
void DrawImage(IBitmapImpl src, double opacity, Rect sect, Rect dest)
    void DrawLine(Pen pen, Point p1, Point p2);
void DrawLine(Pen pen, Point p1, Point p2);
void DrawGeometry(IBrush brush, Pen pen, IGeometryImpl geometry);
void DrawRectangle(Pen pen, Rect rect, float cornerRadius);
void DrawText(IBrush color, Point origin, IFormattedTextImpl text);
void FillRectangle(IBrush brush, Rect rect, float cornerRadius);
    void PushClip(Rect clip);
    void PushOpacity(double opacity);
```





Arc Absolute.out Arc Relative.expe Arc Relative.out.

cted.pna

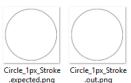
Border_Nested_R

otate.expected.p



otate.out.png







CubicBezier Abs olute.out.png



CubicBezier Relat CubicBezier Relat

ive.out.png





lips_Path.expecte lips_Path.out.png bsolute.expected.







HorizontalLine A bsolute.out.pna

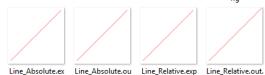


png





Image_Stretch_N one.out.pna



pected.png

t.png

.expected.png



ected.pna



png

LinearGradientBr ush RedBlue_Hor

ed.pna

LinearGradientBr ush RedBlue Hor ush RedBlue Vert ush RedBlue Vert asks Element.exp

ive.expected.png

LinearGradientBr izontal Fill.expect izontal Fill.out.pn ical Fill.expected.

ical Fill.out.png





ected.png

















out.png





_Stroke_8px.out.p neCap.expected.



neCap.out.png

ke.expected.png





















LeftDest.expected





Fill.expected.pn



Fill.out.pna



Fill ClipToBoun ds.expected.png



lute.expected.pn

VerticalLine Abso VerticalLine Abso VerticalLine Relat VerticalLine Relat lute.out.pna ive.expected.pna



VisualBrush Fill NoTile.expected.



NoTile.out.png

VisualBrush NoSt retch_FlipX_TopL retch_FlipX_TopL

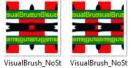


eftDest.out.png

VisualBrush NoSt



LeftDest.out.png



retch_FlipXY_Top retch_FlipXY_Top retch_FlipY_TopL retch_FlipY_TopL eftDest.expected. eftDest.out.png



VisualBrush NoSt retch_NoTile_Alig retch_NoTile nment_BottomRi ght.expected.png

VisualBrush NoSt nment_BottomRi ght.out.png

VisualBrush NoSt pected.png



VisualBrush NoSt VisualBrush NoSt VisualBrush NoSt nment_Center.ex nment_Center.ou nment_TopLeft.e nment_TopLeft.o xpected.png



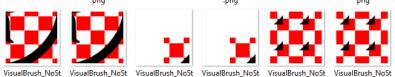
VisualBrush NoSt

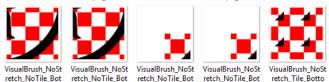






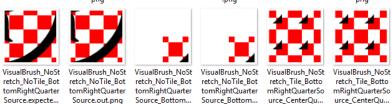
Dest.out.png





eftDest.expected.









VisualBrush Unif .expected.png

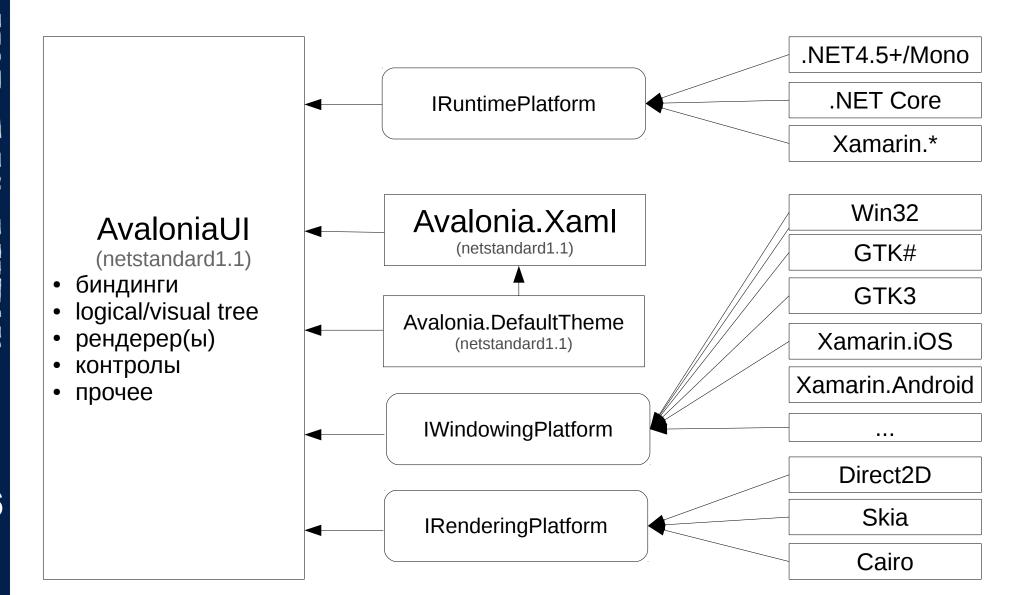
VisualBrush Unif

ormToFill_NoTile ormToFill_NoTile houldBeUsable_B houldBeUsable_R houldBeUsable_R gra8888.expected

WritableBitmapS WritableBitmapS .png

gra88888.out.png

WritableBitmapS gba8888.expecte gba8888.out.png d.png



Action<Rect> Paint { get; set; } Action<Size> Resized { get; set; } Action<double> ScalingChanged { get; set; } void Invalidate(Rect rect); void SetInputRoot(IInputRoot inputRoot); Point PointToClient(Point point); Point PointToScreen(Point point); void SetCursor(IPlatformHandle cursor); Action Closed { get; set; }

IEnumerable<object> Surfaces { get; }

ITopLevelImpl

Action<RawInputEventArgs> Input { get; set; }

Size ClientSize { get; }

double Scaling { get; }

```
ITopLevelImpl
```

```
Size ClientSize { get; }
double Scaling { get; }
IEnumerable<object> Surfaces { get; }
Action<RawInputEventArgs> Input { get; set; }
Action<Rect> Paint { get; set; }
Action<Size> Resized { get; set; }
Action<double> ScalingChanged { get; set; }
void Invalidate(Rect rect);
void SetInputRoot(IInputRoot inputRoot);
Point PointToClient(Point point);
Point PointToScreen(Point point);
void SetCursor(IPlatformHandle cursor);
Action Closed { get; set; }
```

30

Иерархия виджетов верхнего уровня

TopLevel

WindowBase

Окно на экране Перемещение, смена размера, активность, скрытие

TopLevel

Window

Окно с декорациями (заголовок, кнопки)

WindowBase

Окно на экране Перемещение, смена размера, активность, скрытие

TopLevel

Window

Окно с декорациями (заголовок, кнопки)

Popup

Topmost окно без рамки

WindowBase

Окно на экране Перемещение, смена размера, активность, скрытие

TopLevel

Window

Окно с декорациями (заголовок, кнопки)

Popup
Тортоst окно без рамки

EmbeddableControlRoot

TopLevel с поддержкой передачи фокуса

WindowBase

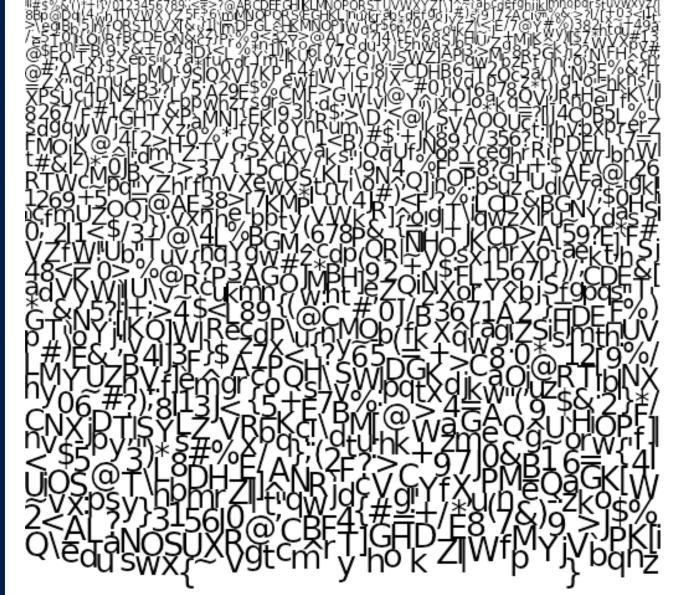
Окно на экране Перемещение, смена размера, активность, скрытие

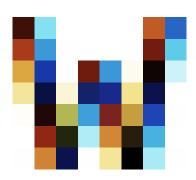
TopLevel

ДЕМО

Проблемы наивного метода отрисовки (рекурсивно бежим по сцене)

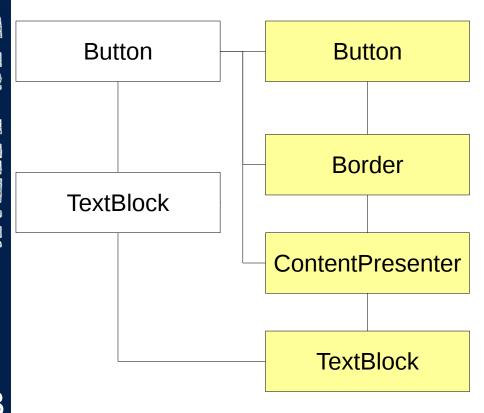
- Сложно понять, что надо отрисовывать, а что нет
- Занимаем много времени на UI-потоке
- Полупрозрачная отрисовка тормозит
- Отрисовка с поворотами и масштабированием тормозит (особенно шрифты)





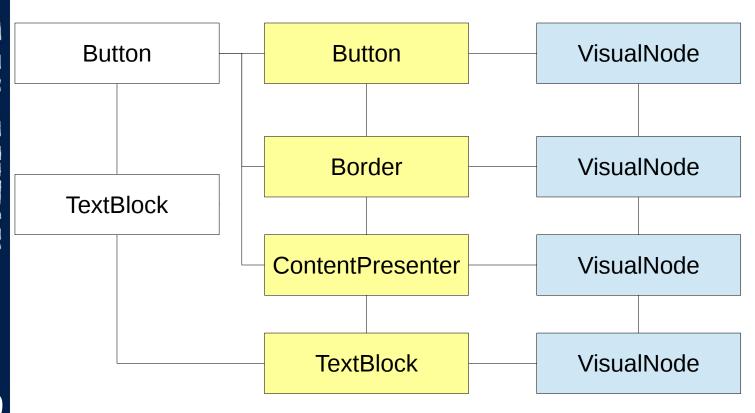
Logical Tree Visual Tree

Render Tree



Logical Tree Visual Tree

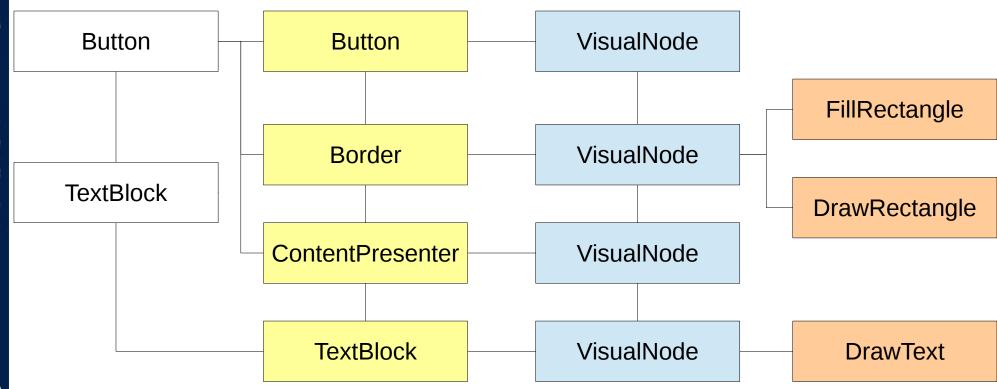
Render Tree



Logical Tree

Visual Tree

Render Tree



ДЕМО

Отличия от WPF

- AvaloniaObject (то же самое, что и DependencyObject)
- Использует ReactiveExtensions
- Другая система стилей
- Отличия в иерархии контролов
- AvaloniaProperty (как DependencyProperty, но строго типизирована)
 - StyledProperty
 - DirectProperty

```
public static readonly DirectProperty<TextBox, int>
   SelectionEndProperty =
   AvaloniaProperty.RegisterDirect<TextBox, int>(
                nameof(SelectionEnd),
                o => o.SelectionEnd,
                (o, v) => o.SelectionEnd = v);
```

```
public static readonly DirectProperty<TextBox, int>
   SelectionEndProperty =
   AvaloniaProperty.RegisterDirect<TextBox, int>(
                nameof(SelectionEnd),
                o => o.SelectionEnd,
                (o, v) => o.SelectionEnd = v);
public int SelectionEnd
  get { return selectionEnd; }
   set { SetAndRaise(SelectionEndProperty, ref
        selectionEnd, CoerceCaretValue(value)); }
```

ДЕМО

56

Чего пока нет?

(версия 0.5)

- Стабильности внутреннего АРІ
- Не хватает многих подсистем, таких как API работы со шрифтами и обработки тач-событий
- Поддержка мобильных устройств на экспериментальном уровне
- «Умный» рендерер пока не в главной ветке
- Для деплоя требует знания целевых платформ
- Документация :(
- Интеграция с IDE
- Выловлены не все проблемы с особенностями платформ

```
SetDefaultSize(200, 200);
btn.Clicked += delegate
    int x, y;
    this.GetSize(out x, out y);
    Console.WriteLine("Clicked ClientSize: " +
        x + " " + y);
    Resize(x + 20, y + 30);
    //???
    this.GetSize(out x, out y);
    Console.WriteLine("AfterClicked ClientSize: " +
        x + " " + y);
```

```
SetDefaultSize(200, 200);
btn.Clicked += delegate
    int x, y;
    this.GetSize(out x, out y);
    Console.WriteLine("Clicked ClientSize: " +
        x + "" + y);
    Resize(x + 20, y + 30);
    //???
    this.GetSize(out x, out y);
    Console.WriteLine("AfterClicked ClientSize: "
        x + " " + y);
```

```
SetDefaultSize(200, 200);
btn.Clicked += delegate
    int x, y;
    this.GetSize(out x, out y);
    Console.WriteLine("Clicked ClientSize: " +
        x + " " + y);
    Resize(x + 20, y + 30);
   //???
    this.GetSize(out x, out y);
    Console.WriteLine("AfterClicked ClientSize: "
        x + " " + y);
```

```
SetDefaultSize(200, 200);
btn.Clicked += delegate
    int x, y;
    this.GetSize(out x, out y);
    Console.WriteLine("Clicked ClientSize: " +
        x + " " + y);
    Resize(x + 20, y + 30);
   //???
    this.GetSize(out x, out y);
    Console.WriteLine("AfterClicked ClientSize: "
        x + "" + y);
```

```
SetDefaultSize(200, 200);
btn.Clicked += delegate
    int x, y;
    this.GetSize(out x, out y);
    Console.WriteLine("Clicked ClientSize: " +
        x + "" + y);
    Resize(x + 20, y + 30);
   //???
    this.GetSize(out x, out y);
    Console.WriteLine("AfterClicked ClientSize:
```

б) 230, 220

B) 220, 268

г) 180, 170

a) 220, 230 б) 230, 220

в) 220, 268 г) 180, 1 д) правильного ответа г) 180, 170 среди вариантов нет

Запланировано на 0.6

- Генерация бандлов для OSX и пакетов для линукса в один КЛИК
- Переезд на Portable.Xaml
- Нативный бакэнд для OSX
- Сделать превьювер для *nix-платформ и интегрировать его хоть куда-нибудь
- Доделать рендерер в отдельном потоке
- Доработать документацию

65

AvaloniaUI.github.io



https://github.com/AvaloniaUI/Avalonia



https://gitter.im/AvaloniaUI/Avalonia

Спикер Никита Цуканов

nikita.d.tsukanov@gmail.com Skype: kekekeks