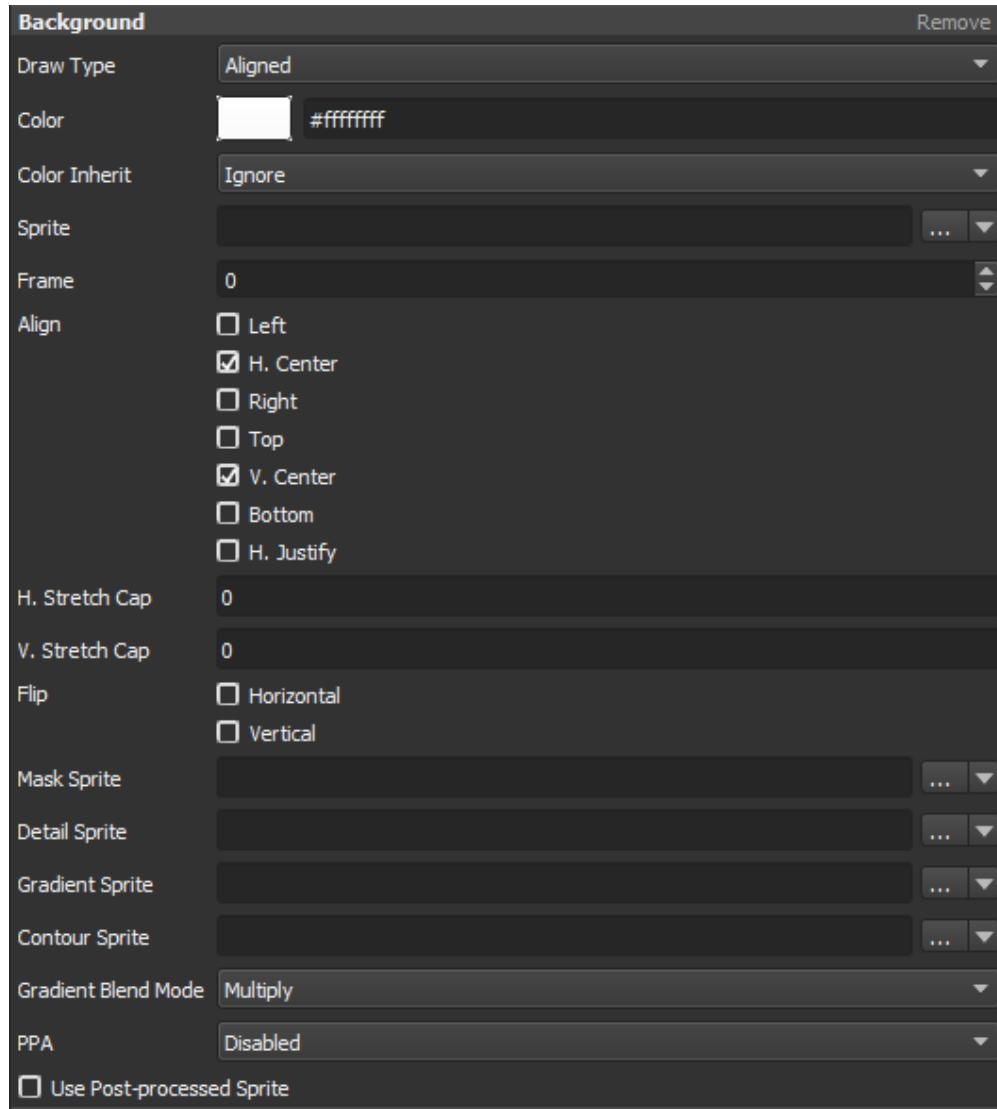


## 2. Content Components

- [Background](#)
- [Text](#)
- [Rounded Corners](#)
- [Border](#)

### Background

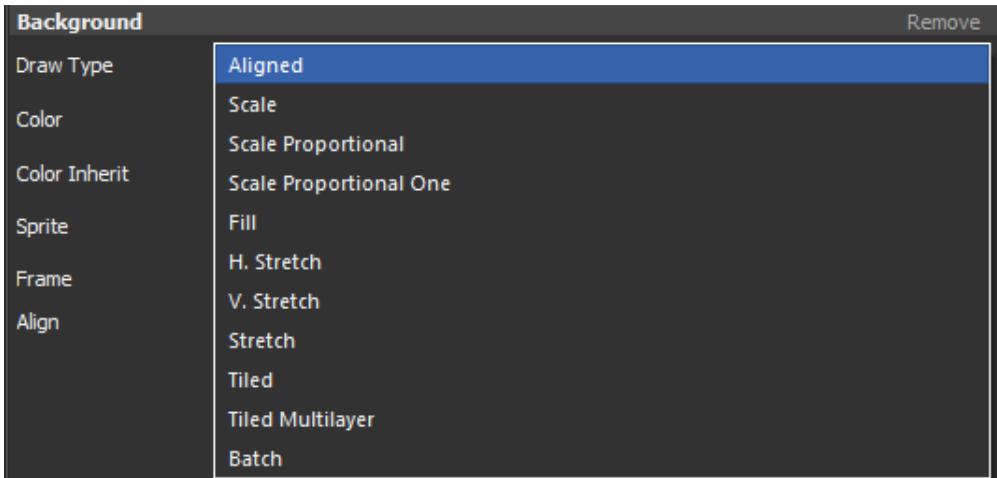
Компонента **Background** панели Modern Properties находится во вкладке Content и имеет следующий вид:



Рассмотрим основанные свойства компоненты Background:

#### Draw Type

Свойство Draw Type служит для указания типа заливки контрола:



**Aligned** - Выравнивает спрайт по центру контроля с сохранением дефолтного размера;

**Scale** - Растигивает назначенный в поле sprite спрайт по контролю (без сохранения пропорций)

**Scale Proportional** - Растигивает назначенный в поле sprite спрайт по контролю (пропорционально)

**Scale Proportional One** - Растигивает назначенный в поле sprite спрайт по контролю, ориентируясь на размер большей стороны (пропорционально)

**Fill** - Заполняет контрол цветом. Цвет указывается в поле Color

**H. Stretch** - Растигивает назначенный в поле sprite спрайт по контролю в горизонтальном направлении (вертикальное значение остается неизменным)

**V. Stretch** - Растигивает назначенный в поле sprite спрайт по контролю в вертикальном направлении (горизонтальное значение остается неизменным)

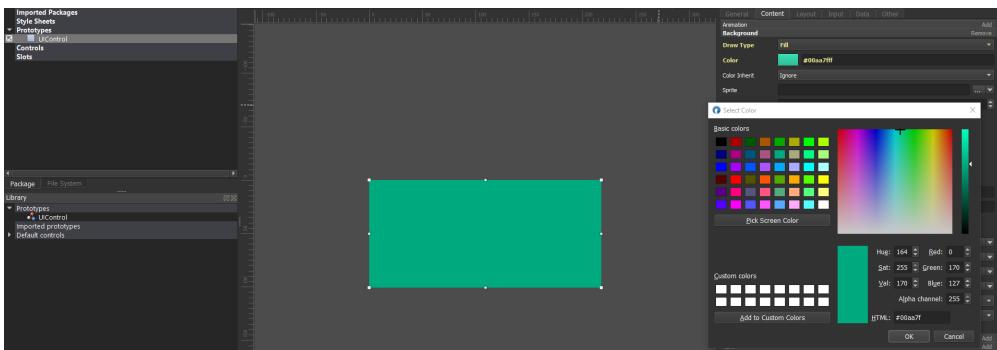
**Stretch** - Растигивает назначенный в поле sprite спрайт по контролю (без сохранения пропорций)

**Tiled** - Заполняет все пространство контроля спрайтами (спрайты дефолтного размера)

**Tiled Multilayer** - В сочетании с настройками mask, detail, gradient, contour дает комбинированный бэкграунд

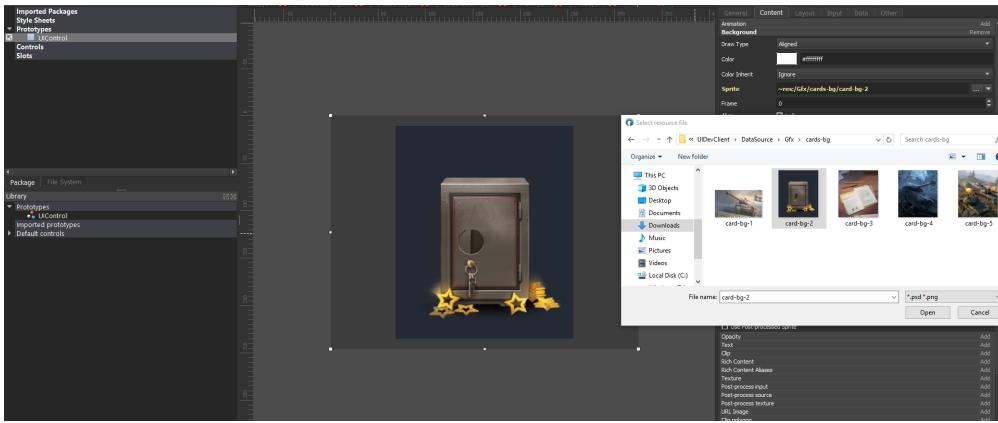
### Color

Свойство Color позволяет задавать цвет фона, работает совместно с Draw Type - Fill . По клику на белый прямоугольник открывается окно Select color, в котором можно выбрать интересующий нас цвет.



### Sprite

Поле Sprite служит для указания пути к psd либо png - файлу, который выполняет роль спрайта. Чтобы указать путь необходимо кликнуть по троеточию справа в поле sprite. Откроется окно выбора ресурсов, в окне необходимо выбрать интересующий нас файл.



Свойство **Sprite** доступно для биндинга. Пример того, как можно биндить свойства приведен в документе о Data Binding System.

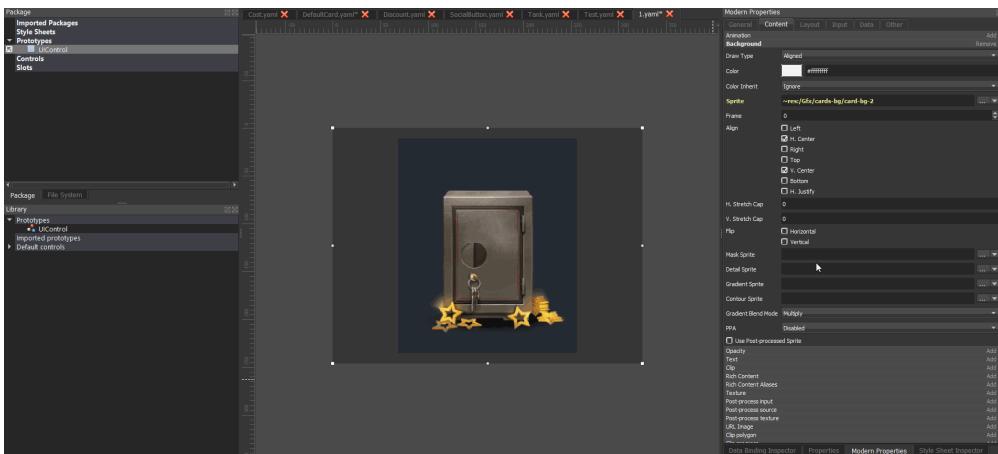
### Align

Свойство Align позволяет располагать заданный спрайт в пределах контроля слева/справа/сверху/снизу либо по центру.



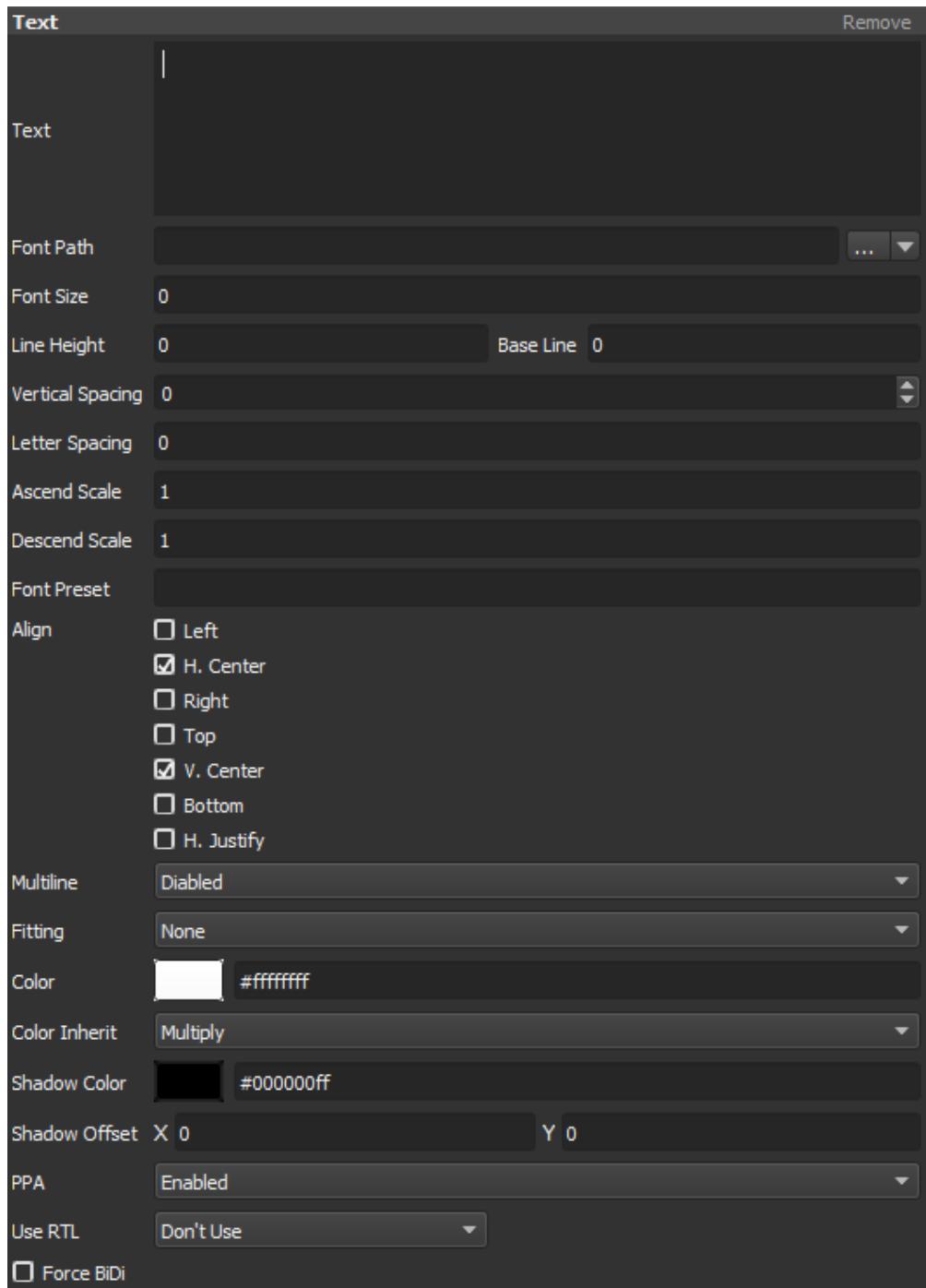
### Flip

Свойство Flip позволяет отражать картинку по горизонтали либо по вертикали:



## Text

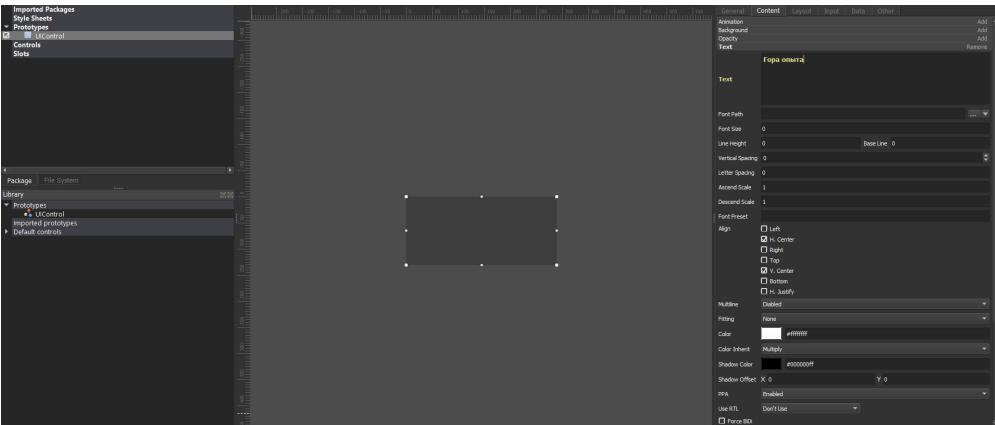
Компонента **Text** предназначена для работы с текстом, панели Modern Properties компонента находится во вкладке Content и имеет следующий вид:



Рассмотрим основанные свойства компоненты Text:

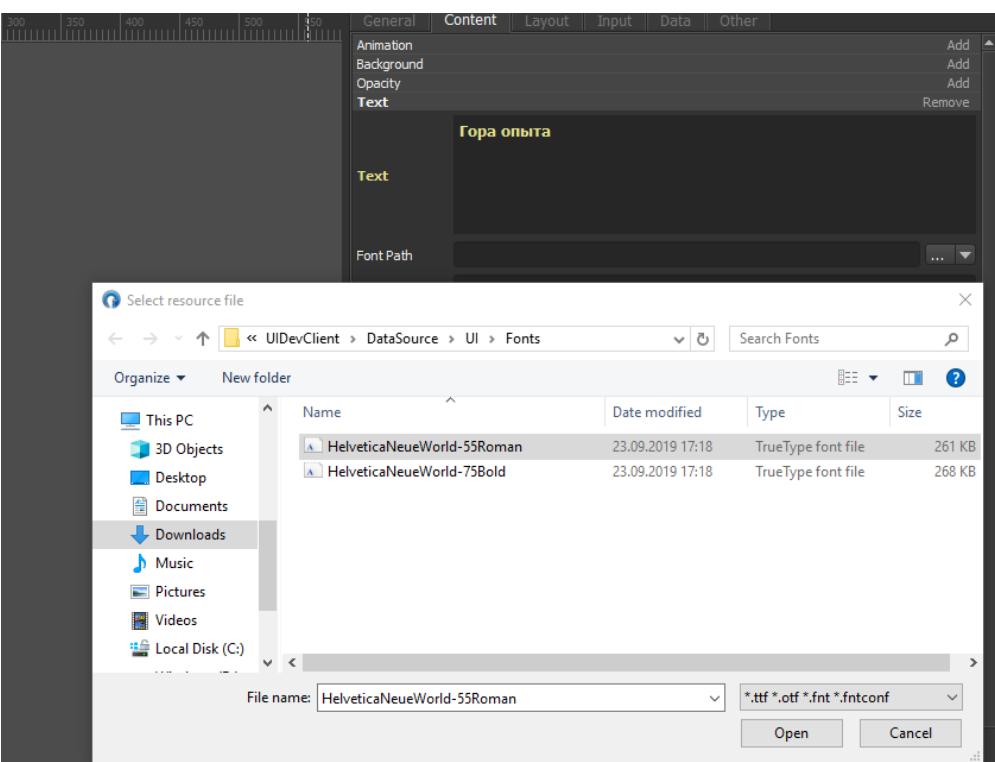
### Text

Поле Text предназначено для ввода текста. Для того чтобы текст отобразился в контролле в центральном виджете необходимо кроме текста задать также шрифт в поле Font Path.

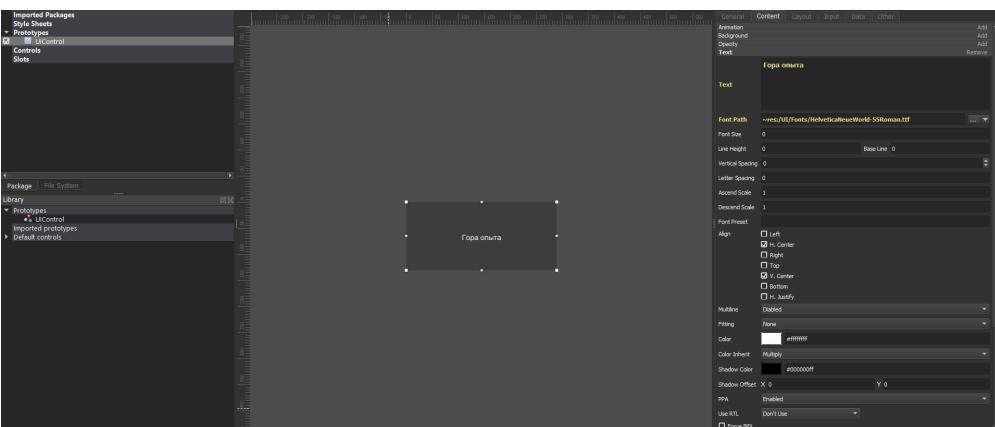


### Font Path

Поле Font Path предназначено для указания файла со шрифтом. Для этого справа в поле необходимо нажать на кнопку "...", после этого откроется окно выбора ресурса. В окне необходимо выбрать интересующий нас шрифт и нажать Open.



После того как был выбран интересующий нас шрифт, в центральном виджете начнет отображаться введенный ранее текст.



## Font Size

Свойство Font Size служит для указания размера шрифта.



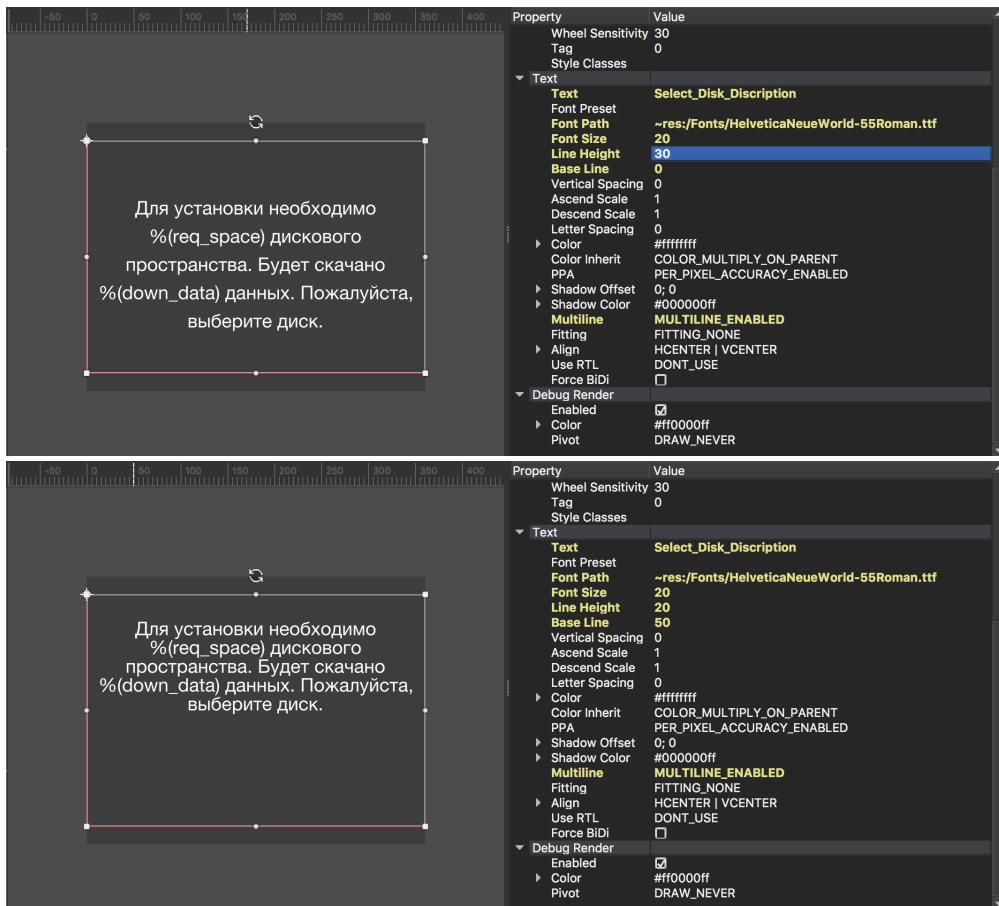
## Line Height & Base Line

Свойства Line Height и Base Line используются для многострочного текста в компоненте Text и выполняют следующие функции:

- Line Height - в данном свойстве задается высота одной строки текста (в Size Policy значение content size будет равно этому значению, расстояние от начала строки до начала следующей в многострочном тексте будет равно этому значению плюс Vertical Spacing)
- Base Line - свойство отвечает за указание расстояния от линии, на которой располагается текст, до низа текстового блока, размер которого задается Line Height.

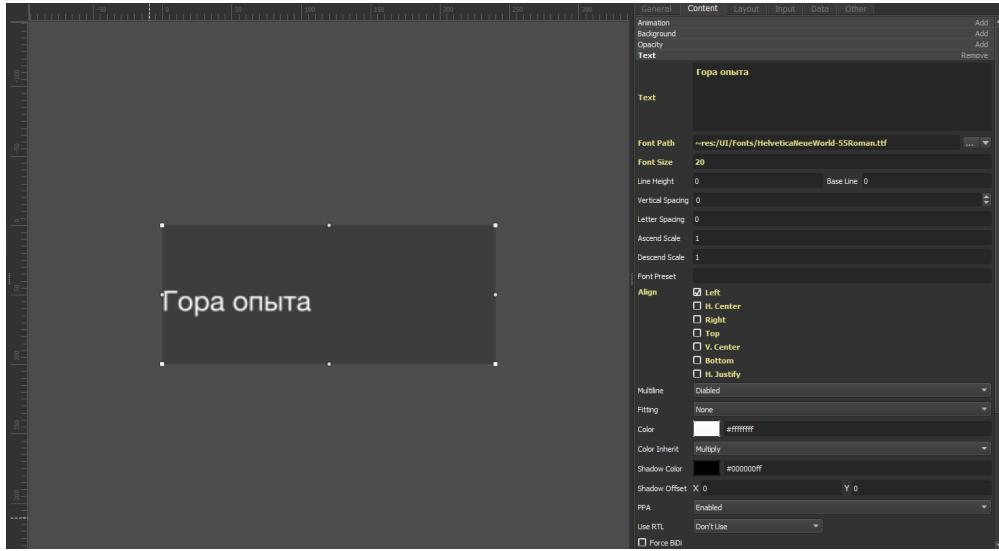
Если значение свойства Line Height не равно нулю, то высота строки задается через свойство Line Height и работает свойство Base Line, но перестают работать Ascend Scale и Descend Scale. В противном случае высота строки вычисляется из размера шрифта и свойств Ascend Scale и Descend Scale.

Также значение свойств Line Height и Base Line можно задавать через стили.



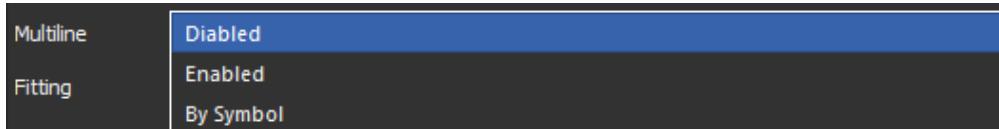
## Align

Свойство Align предназначено для выравнивания текста. По умолчанию текст располагается в центре. Но доступна возможность выравнивать по левому, правому краю, а также располагать текст сверху и снизу.



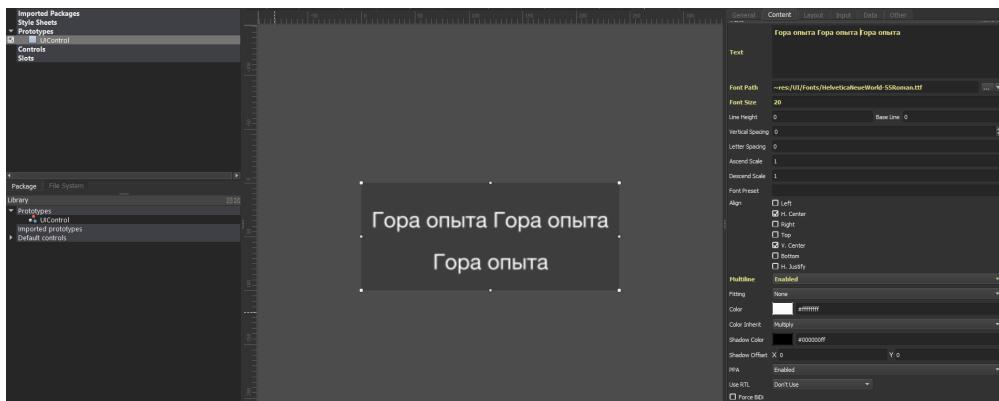
## Multiline

Свойство используется в случае, если длина текста не помещается в одной строке. Опция Multiline позволяет разделить текст на несколько строк.



Enabled - выполняется перенос текста по словам

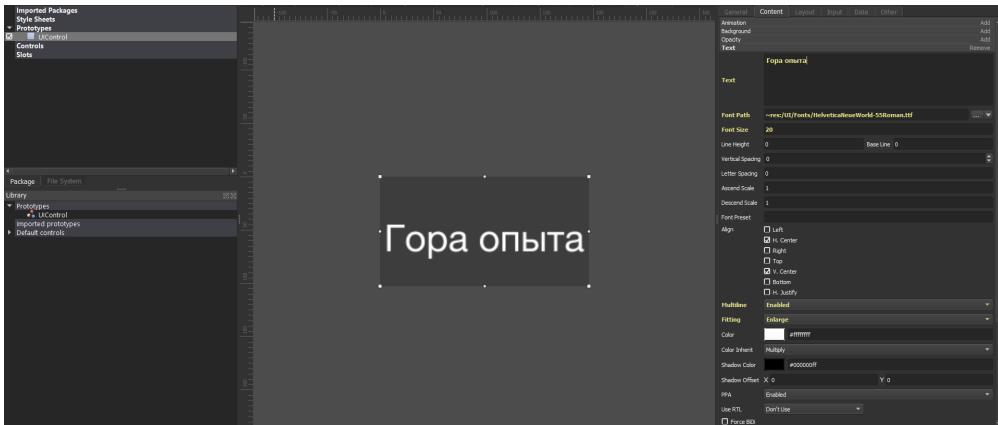
By Symbol - выполняется перенос текста посимвольно



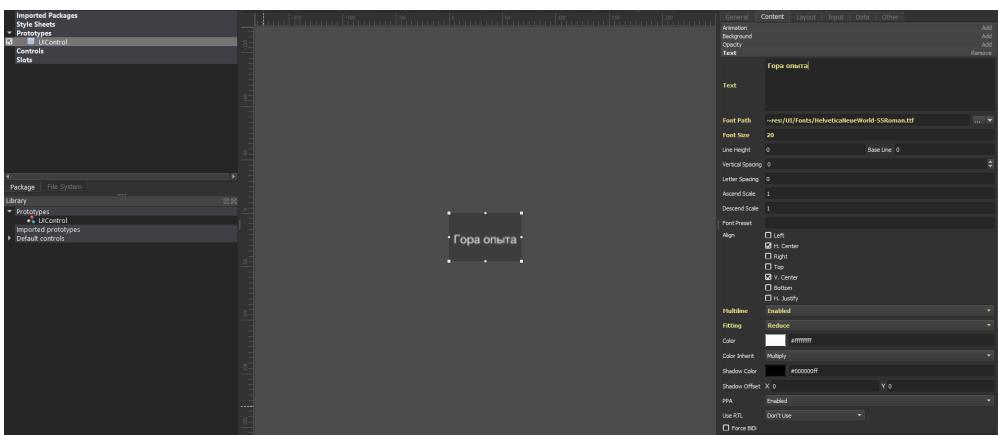
## Fitting

Свойство Fitting позволяет выбирать вариант подгонки текста в соответствии с размером контрола.

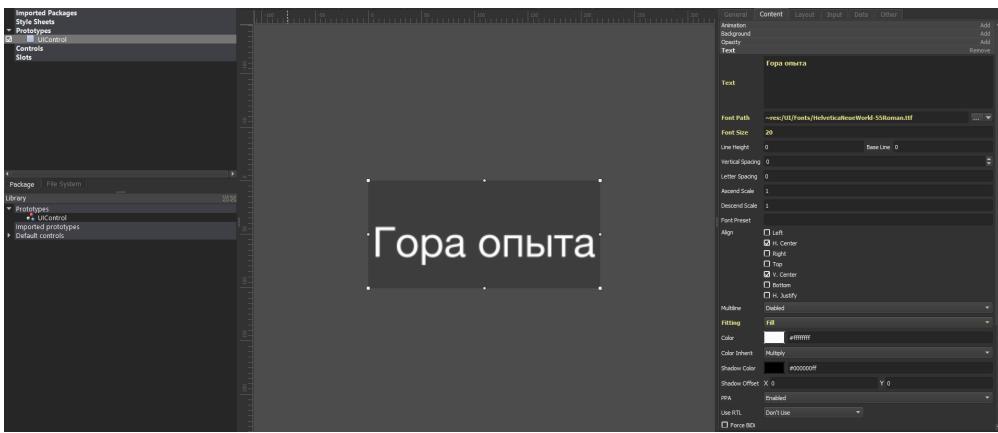
Enlarge- Размер шрифта увеличивается пропорционально границам контрола. Уменьшаться может только до размера текущего шрифта:



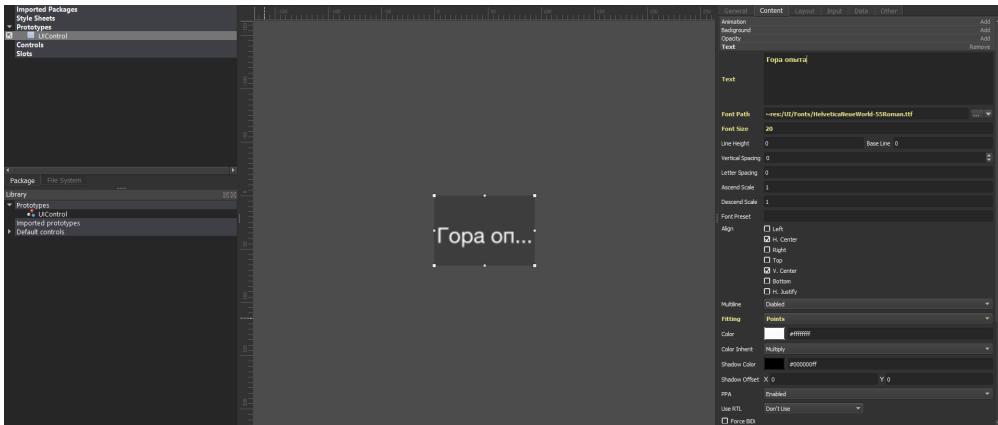
Reduce - Размер шрифта уменьшается пропорционально границам контрола. Увеличиваться может только до размера текущего шрифта:



Fill - Размер шрифта изменяется пропорционально границам контрола, может как увеличиваться так и уменьшаться на большую величину:

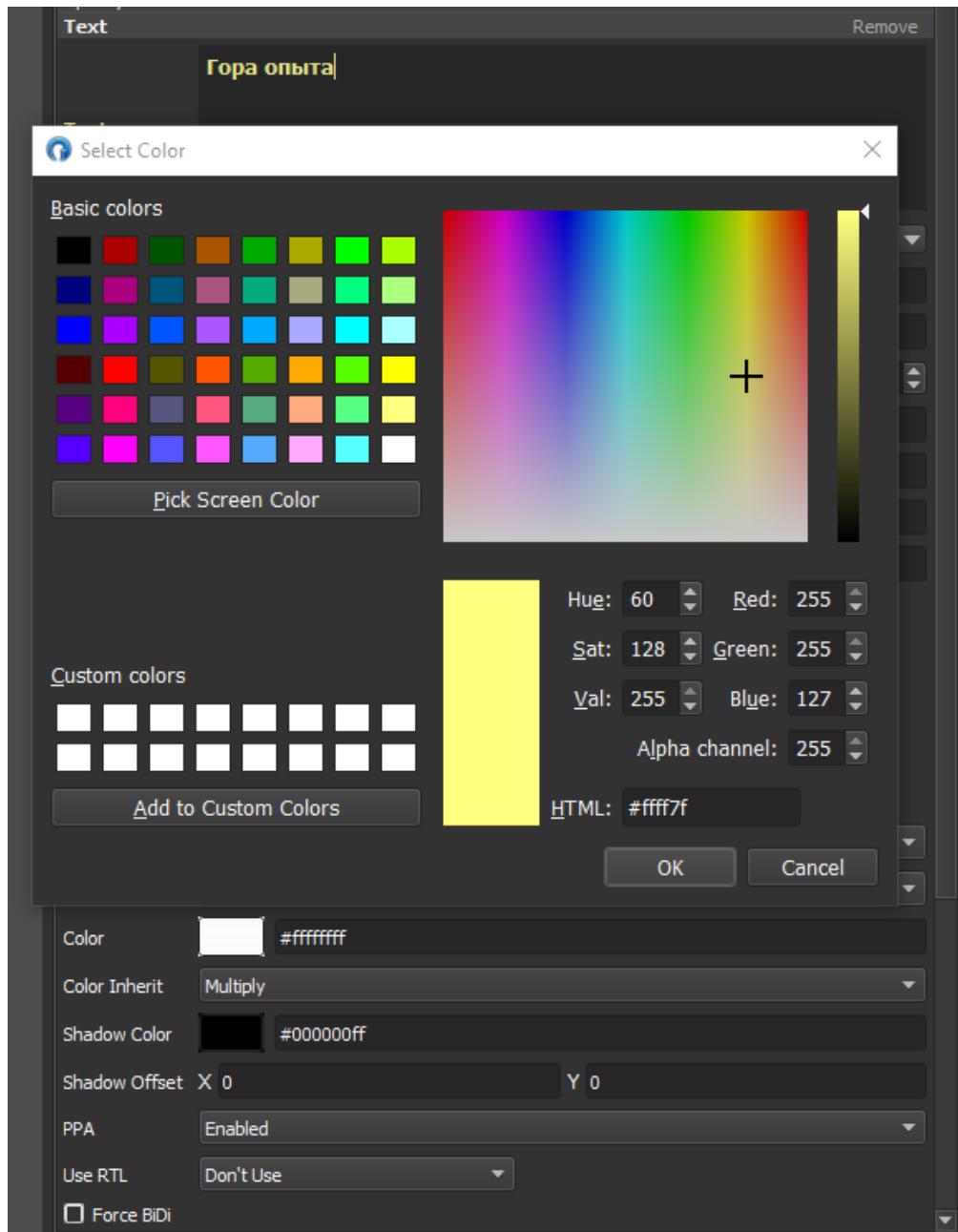


Points - Когда текст не помещается в границах контрола - последние символы замещаются троеточием:



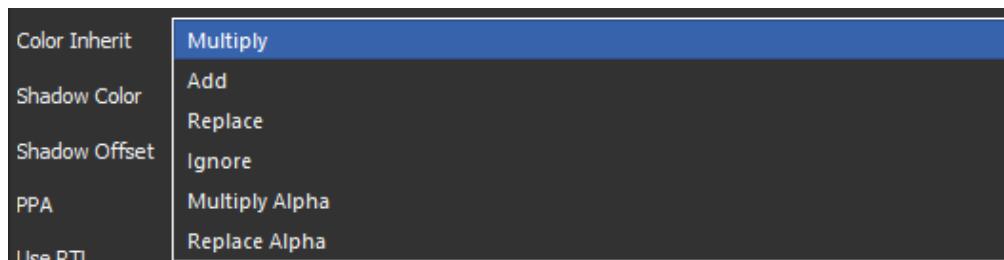
## Color

Свойство Color позволяет задавать цвет текста. По клику на прямоугольник открывается окно Select color, в котором можно выбрать интересующий нас цвет.



### Color Inherit

В случае, если контрол имеет заливку каким либо цветом, свойство Color Inherit позволяет задавать смешение цвета фона с цветом текста.



Multiply - Происходит умножение: конечный результат = RGB-значение цвета текста дочернего контрола умноженное на RGB-значение цвета слоя родительского контрола

Add - Происходит добавление: конечный результат = RGB-значение цвета текста дочернего контрола добавленное к RGB-значению цвета слоя родительского контрола

Replace- Происходит замещение цвета текста текущего дочернего контрола на цвет слоя родительского контрола

Ignore - Остается исходный цвет текста текущего контрола

Multiply Alpha- Происходит умножение текущего цвета текста дочернего контрола и родительского цвета фона с учетом альфа-канала

Replace Alpha -Происходит замещение цвета текста на родительский альфа-канал

## Rounded Corners

Компонента **Rounded Corners** работает совместно с компонентой **Background** на панели Modern Properties находится во вкладке Content. Компонента имеет следующий вид:



**Radius** - поле, предназначенное для задания радиуса скругления (можно задать радиус как для каждой вершины контрола так и по осям x, y в отдельности).

**IsRound** - чек, отвечающий за скругление. Если чек включен, угол будет скругляться, если чек выключен, угол будет скошенным на расстояниях соответствующих радиусу.

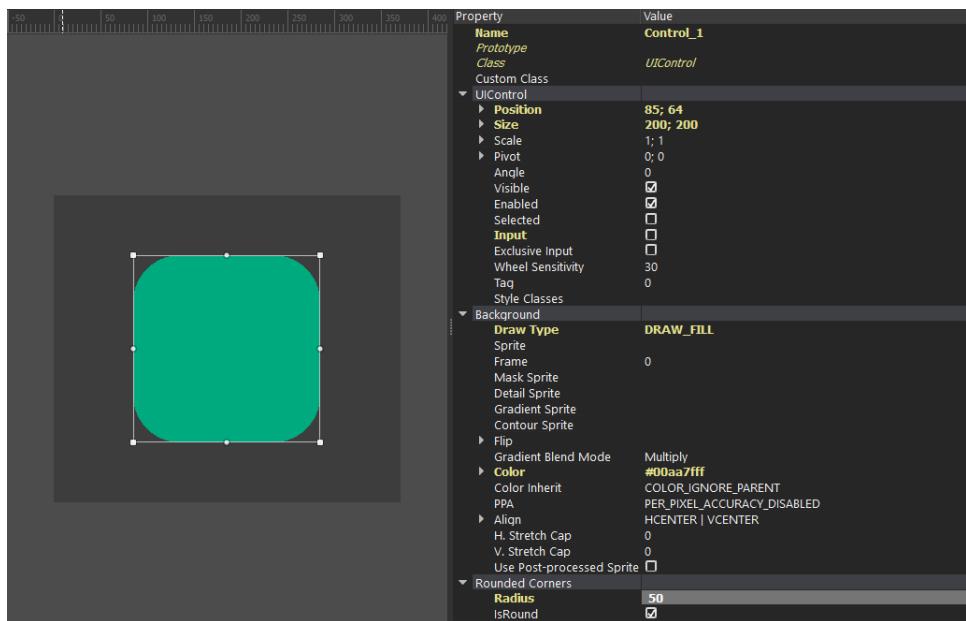
Чек IsRound включен	Чек IsRound выключен
<p>The screenshot shows a teal-colored rounded rectangle with rounded corners. The background is dark gray. The component settings show 'Radius' set to 50 and 'IsRound' checked.</p>	<p>The screenshot shows a teal-colored square with sharp corners. The background is dark gray. The component settings show 'Radius' set to 50 and 'IsRound' unchecked.</p>

В поле Radius значение радиуса задается в следующем формате: x,y; x,y; x,y; x,y; - где x,y - радиусы по x и по y соответственно для каждой вершины контрола.

Варианты задания радиуса:

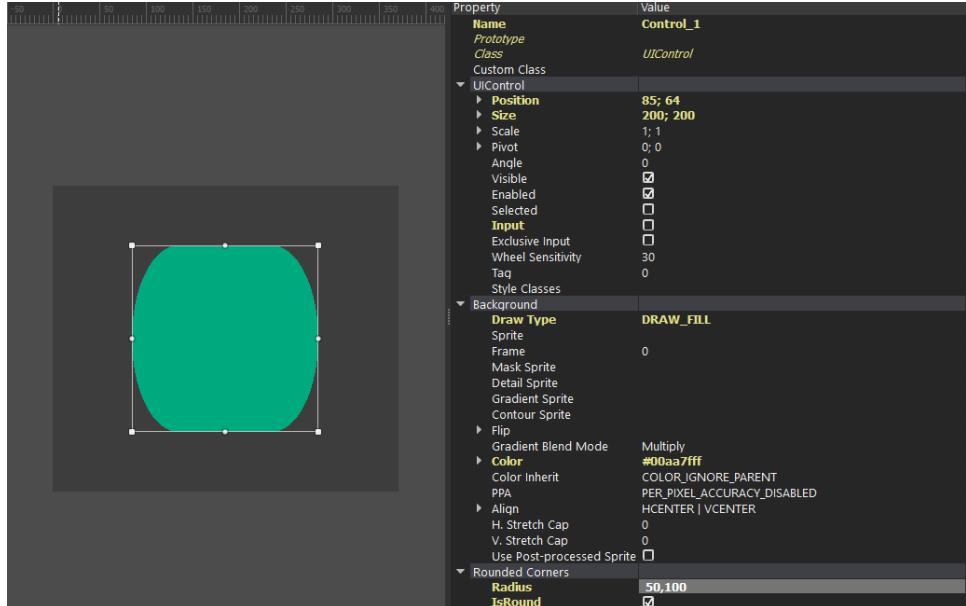
1) При вводе в поле Radius одного значения, это будет означать, что всем вершинам будет задан одинаковый радиус по всем осям.

Пример: **ввод (50)** приведет к применению следующих значений: (50,50; 50,50; 50,50; 50,50;). В поле Radius при этом запись будет иметь следующий вид: (50).



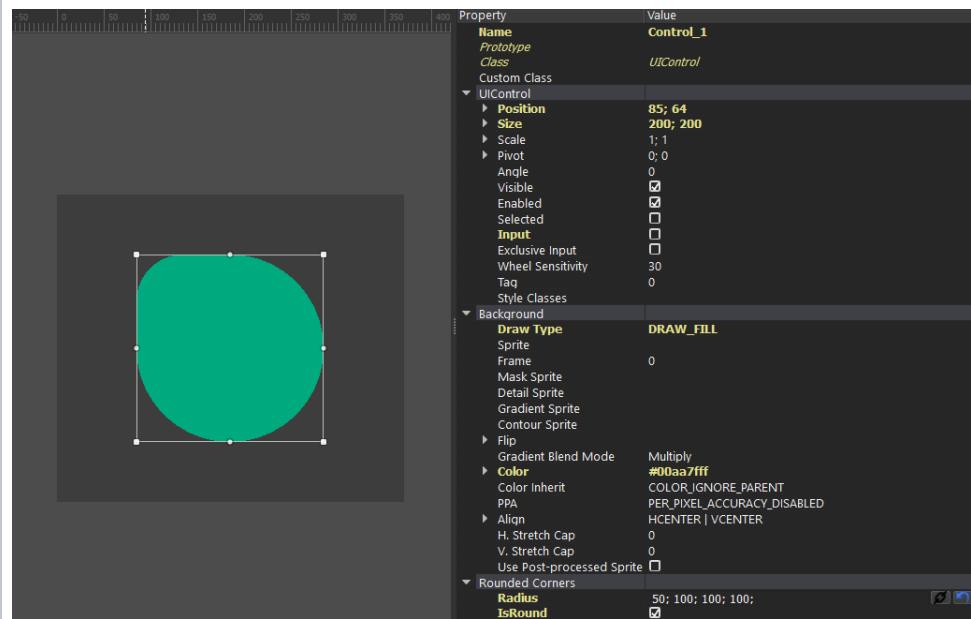
2) При вводе в поле Radius двух значений через запятую, это будет означать, что у всех вершин одинаковый радиус, по оси x - первое число, а по оси y - второе.

Пример: **ввод (50,100)** приведет к применению следующих значений: (50,100; 50,100; 50,100; 50,100;). В поле Radius при этом запись будет иметь следующий вид: (50,100)

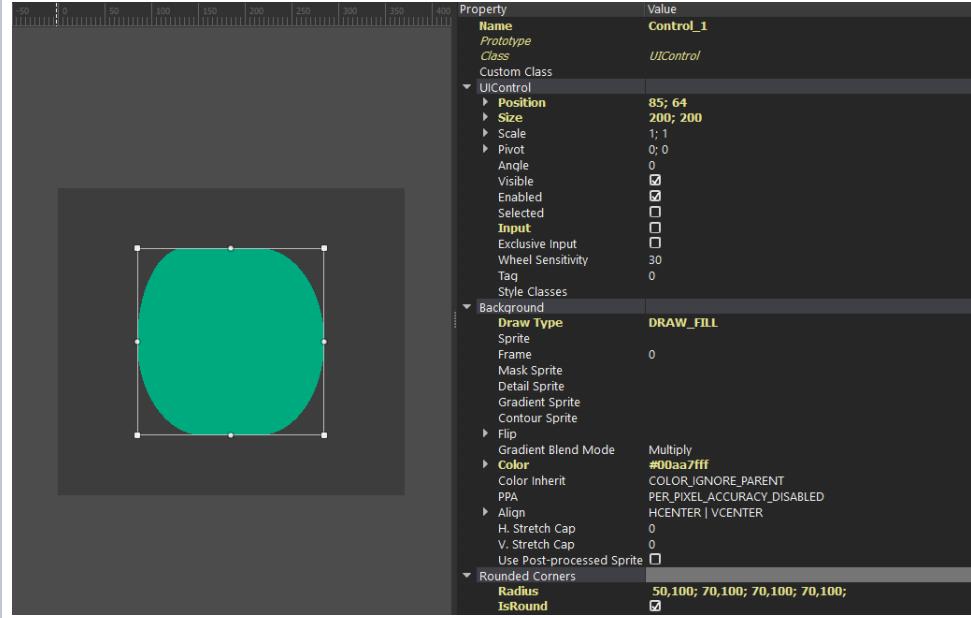


3) При вводе в поле Radius двух и более значений через ";" , это будет означать, что последнее значение отделенное ";"  
продублируется для остальных значений.

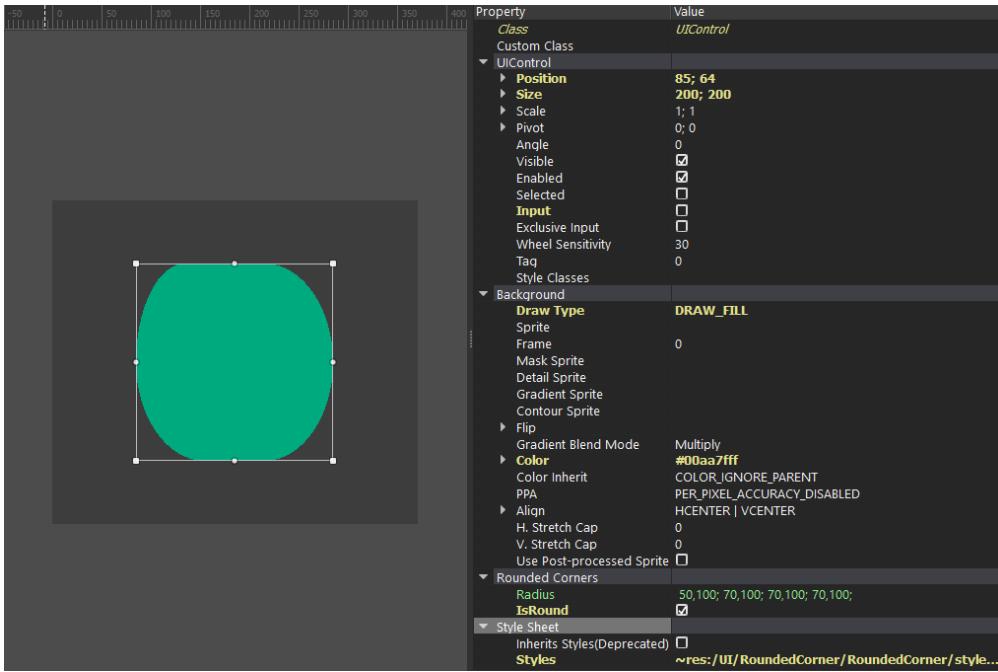
Пример: **ввод (50; 100)**  
приведет к применению следующих значений: (50,50;  
100,100; 100,100; 100,100;) т.е. у верхней левой вершины радиус по x = 50 и по y = 50, для верхней правой вершины и всех остальных вершин радиус по x = 100 и по y = 100. В поле Radius при этом запись будет иметь следующий вид: (50; 100;  
100; 100;).



4) Пример: **ввод (50,100;  
70,100)** приведет к применению следующих значений: (50,100; 70,100;  
70,100; 70,100;) т.е у верхней левой вершины радиус по оси x = 50, по оси y =100, для всех остальных вершин радиус по x = 70, по y = 100. В поле Radius при этом запись будет иметь следующий вид: (50,100;  
70,100; 70,100; 70,100;).

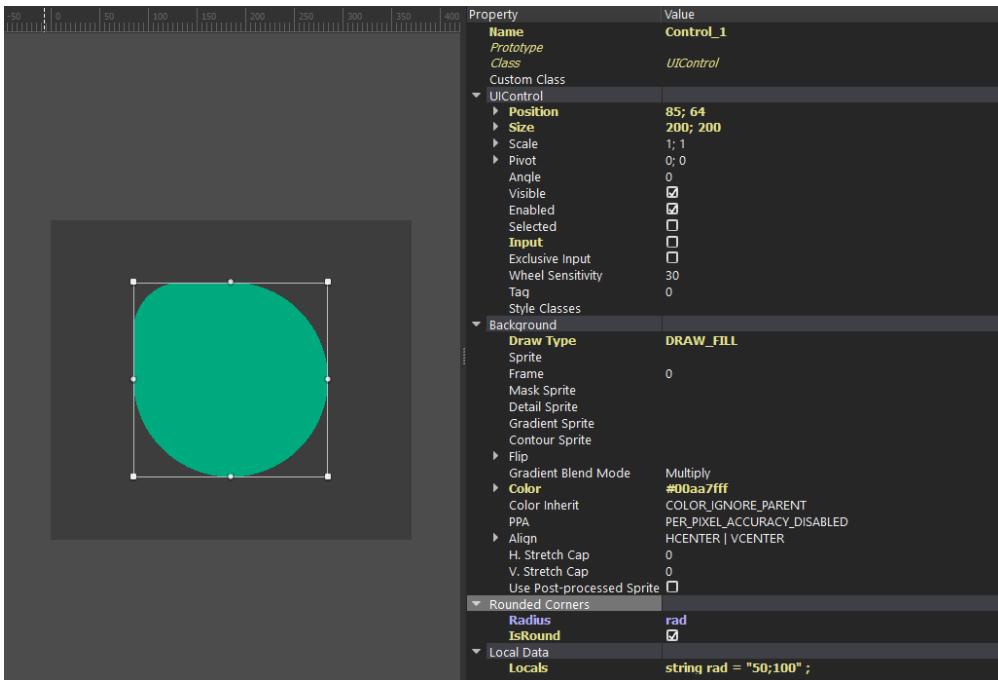


Радиус можно задать стилем, задание аналогично, как в примерах в таблице.



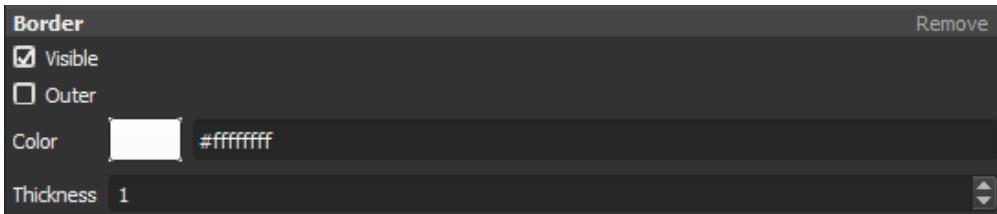
Радиус можно задать через Data Binding.

Для этого необходим задать либо параметр, либо локальный аргумент типа string.



## Border

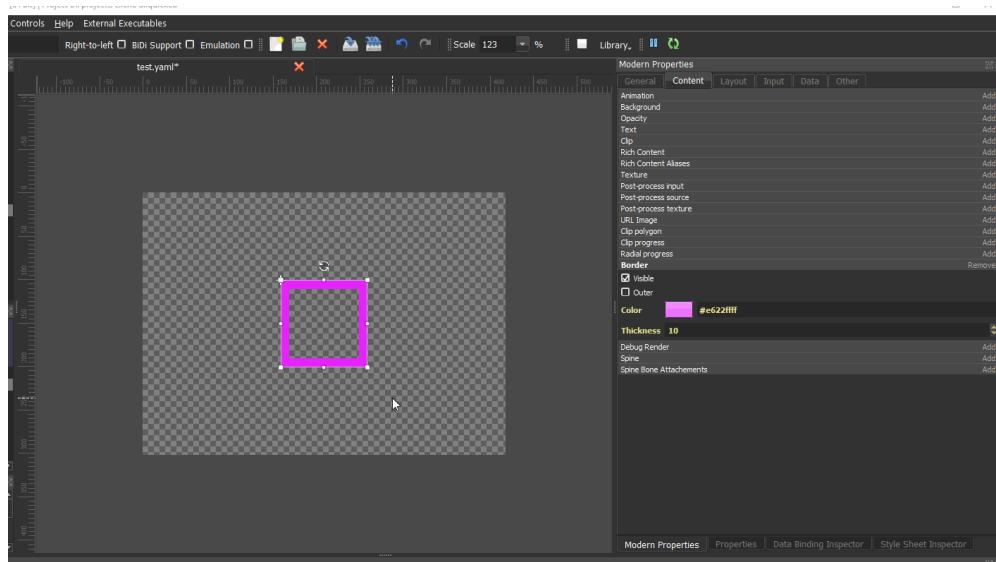
Компонента Border предназначена для отрисовки границ контроля. На панели Modern Properties она находится во вкладке Content и имеет следующий вид:



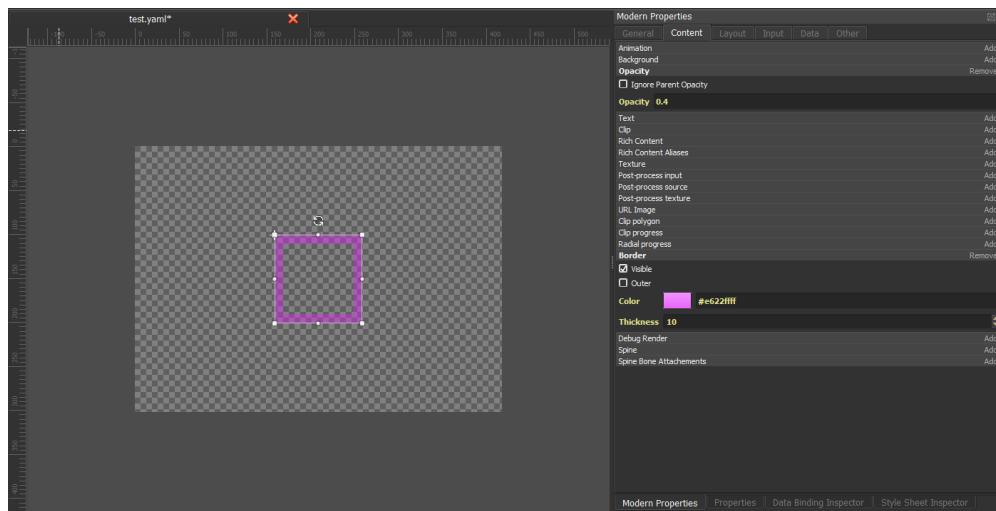
Чекбокс **Visible** отвечает за отображение компоненты.

Поле **Thickness** предназначено для указания ширины границы в пикселях.

Чекбокс **Outer** отвечает за расположение границы контрола. По умолчанию чекбокс выключен и отрисовка границы будет располагаться внутри контрола, при включении чекбокса - снаружи.



Компонента **Border** может взаимодействовать с компонентой **Opacity**:



Реализована возможность задавать свойства компоненты при помощи стилей:

