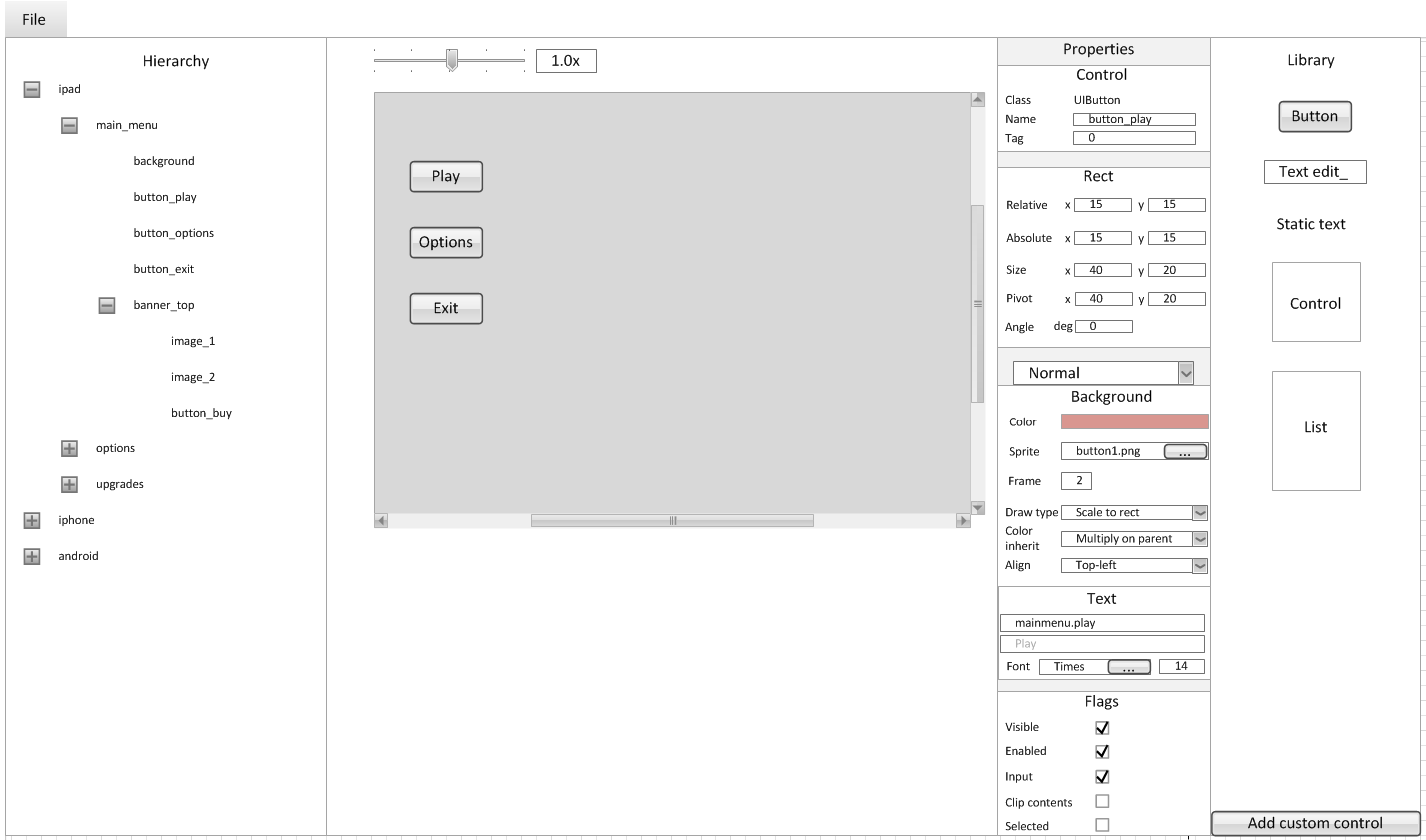
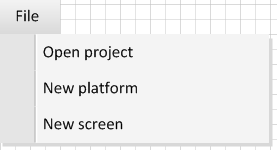
UI Editor



## File



### Open Project

Открывает диалог File Open, в котором выбирается папка проекта. Папка проекта имеет следующую структуру:

Project

DataSource

UI

ipad

ui.project

screens

main\_menu.yaml

options.yaml

upgrades.yaml

controls

complex\_button.yaml

custom\_control.yaml

iphone

ui.project

screens

main\_menu.yaml

options.yaml

upgrades.yaml

controls

complex\_button.yaml

custom\_control.yaml

android

ui.project

screens

main\_menu.yaml

options.yaml

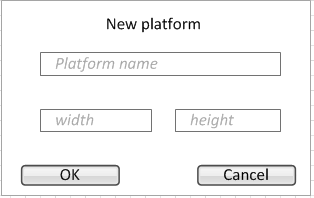
upgrades.yaml

controls

complex\_button.yaml

custom\_control.yaml

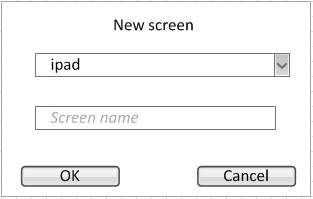
### New platform



Добавляет папку Project/DataSource/UI/%*PLATFORM\_NAME*%, соответственно меняется панель Hierarchy.

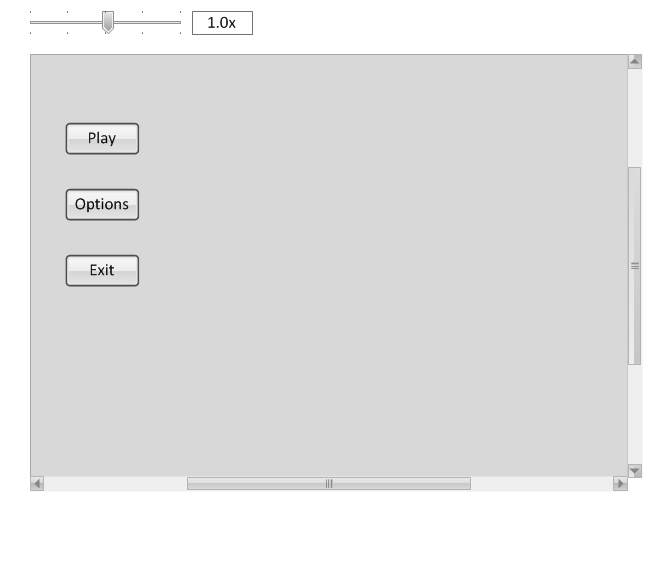
В этой папке создается ui.project – yaml файл c полями *width* и *height*, которые определяют размер всех создаваемых экранов на данной платформе.

### New screen



Создает файл Project/DataSource/UI/%*SELECTED\_PLATFORM*%/screens/%*SCREEN\_NAME*%, соответственно меняется панель Hierarchy.

## Editor



В этом окне происходит непосредственно выбор и перемещение контролов. Рендер проходит через dava.framework, т.е. картинка выглядит точно так же, как она будет выглядеть в клиентском приложении.

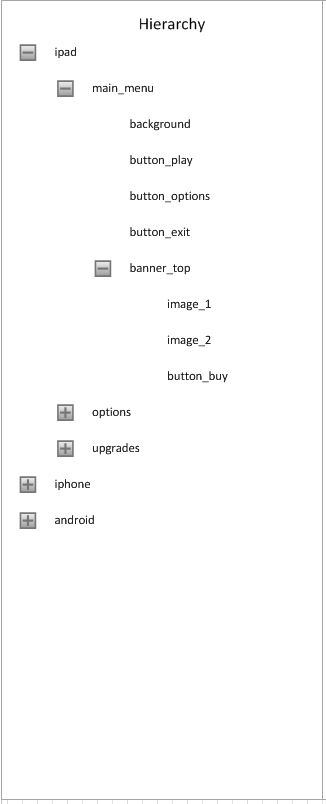
Если редактируемый экран не влазит в qt-контрол, добаляются скроллбары. Также можно скролить экран комбинацией *Space+mousemove* (как в Photoshop).

Зум контролируется слайдером либо комбинацией *Ctrl+mousewheel.*

Контрол клонируется комбинацией *Ctrl+mousemove*.

Активный (выбранный) контрол помечается красной рамкой через *control->SetDebugDraw(true)*. Контролы можно перемещать мышкой. Активный контрол также можно двигать стрелками с клавиатуры.

## Hierarchy

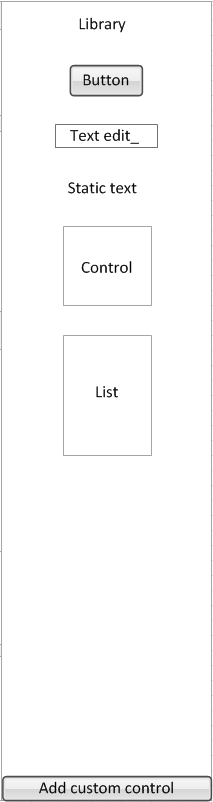


Отображает иерархию экранов/контролов. На верхнем уровне иерархии – папки из Project/DataSource/UI/, потом – экраны и контролы. Названия папок и экранов берутся из названий соответствующих папок и файлов. Название контрола – *UIControl::name*. Иконка соответствует классу контрола.

Поддерживается drag’n’drop контролов как внутри экрана, так и между экранами.

## Library

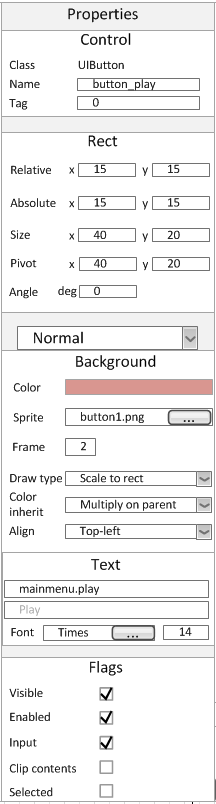
Библиотека контролов.



При перетаскивании на Editor контролы добавляются в редактируемый экран.

Кнопка *Add custom control*  вызывает меню открытия файлов и позволяет выбрать .yaml файл с кастомным контролом из папки Project/DataSource/UI/%*SELECTED\_PLATFORM*%/controls/

## Properties



Служит для отображения и редактирования свойств активного контрола.

### Control

* Class – с++ класс контрола
* Name – *UIControl::name*. Редактируется также через панель Hierarchy. При добавлении контрола из Library назначается имя типа *shortClassName\_autoIncrementIndex*, например *button\_2*, *text\_1*. *autoIncrementIndex* начинается с единицы и растет для каждого класса контролов отдельно в рамках одного экрана. Например, если создаются 4 контрола на одном экране в указанном порядке: button, text, button, text, то они будут имена соответственно: *button\_1, text\_1, button\_2, text\_2.* При клонировании контрола к его имени добавляется суффикс ­*\_copyN*. Например, *button\_play => (clone) => button\_play\_copy1*. Если имя контрола уже заканчивалось на ­*\_copy1*, то имя меняется на *\_copy2* (если такое имя уже занято, то на *\_copy3* и т.д.).
* Tag – *UIControl::tag*.

### Rect

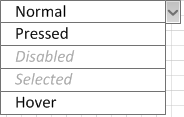
* Relative – *UIControl::relativePosition*
* Absolute – *UIControl::absolutePosition*
* Size – *UIControl::size*
* Pivot – *UIControl::pivotPoint*
* Angle – *UIControl::angle*

### State selector



Combobox, который переключает состяние контрола, которое будет редактироваться в панели Background (а для класса *UIButton* еще и в панели Text).

Если для какого-то состояния пользователь не менял никакие настройки, то название этого состояния выделено другим шрифтом. В примере

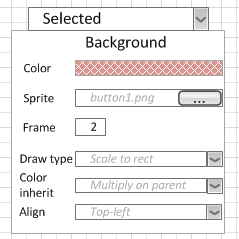


для *Disabled* и *Selected* настройки не менялись.

### Background

* Color – UIControl::background::color
* Sprite – UIControl::background::spr::relativePathname
* Frame – UIControl::background::color
* Draw type – UIControl::background::type
* Color inherit – UIControl::background::colorInheritType
* Align – UIControl::background::align

Если в State selector’е выбрано любое состояние кроме Normal и какая-то настройка не отличается от соответствующей в состоянии Normal (либо пользователь не менял ее явно), то она выделяется другим шрифтом. Если совпадает цвет, то он отображается с паттерном. В примере



в состояние Selected все настройки, кроме Frame совпадают с состоянием Normal.

### Text

Существует для классов UIButton и UIStaticText.



В первой строке задается ключ для файла локализации. Вторая строка при этом отображает реальный текст, который соответсвует этому ключу. Этот же текст отображается на контролах в панели Editor.

Font. Запускает диалог Open File, где можно выбрать нужный шрифт из папки проекта.

Font size. Размер шрифта.

### Flags

* Visible – *UIControl::visible*
* Enabled – *UIControl::SetDisabled()*
* Input – *UIControl::inputEnabled*
* Clip contents – *UIControl::clipContents*
* Selected – *UIControl::SetSelected()*