Контекстно-ориентированная система рендеринга двухмерной графики для OS X

Автор: Хабаров Александр 8”Б”

Научный руководитель: Дединский Илья Рудольфович

Актуальность

Низкоуровневая графика OS X: Core Graphics - C framework

Pros	Cons
• Полный низкоуровневый доступ к системе	• Большой объем кода • Отсутствие ООП и событийности • Сложность программ

?

Актуальность

Низкоуровневая графика OS X: Core Graphics - C framework

Pros	Cons
• Полный низкоуровневый доступ к системе	• Большой объем кода • Отсутствие ООП и событийности • Сложность программ

Lightweight wrapper: TX-C & Obj-C framework

Pros	Cons
• ОПП и событийность • Небольшой объем кода • Простота программ	• Ограниченный доступ к системе

Цель

- Реализовать фреймворк, облегчающий:
  - Рендеринг двухмерной графики
  - Попиксельный доступ к контексту фреймворка Core Graphics операционной системы OS X 10.8
  - Защиту внутренних данных в многопоточных приложениях

CGBitmapContext: Создание

```
_context = CGBitmapContextCreate(
    self.data,                          // Массив точек
    self.bounds.size.width,             // Ширина объекта
    self.bounds.size.height,            // Высота объекта
    _TX_BITS_PER_COMPONENT,             // Кол-во бит
    self.bytesPerRow,                   // Кол-во байт в ряду
    colorSpace,                         // Способ передачи цвета
    kCGImageAlphaPremultipliedLast |    // Premultiplied RGBA
    kCGBitmapByteOrder32Big             // Формат - big endian
    );
```

CGBitmapContext: Использование

```
CGImageRef imageRef =
    CGBitmapContextCreateImage(context); // Создание изображения

[[[NSImage alloc] initWithCGImage:imageRef
                             size:NSZeroSize]

drawAtPoint:CGPointMake(self.bounds.origin.x,  // Рендеринг
                        self.bounds.origin.y)
fromRect:   self.bounds
operation:  NSCompositeSourceOver
fraction:   1.0];

CGImageRelease(imageRef);                      // Освобождение ресурсов
```

Событие перерисовки → вывод содержимого на экран

Core Graphics

Высокоуровневая разработка

Низкоуровневая 2D графика

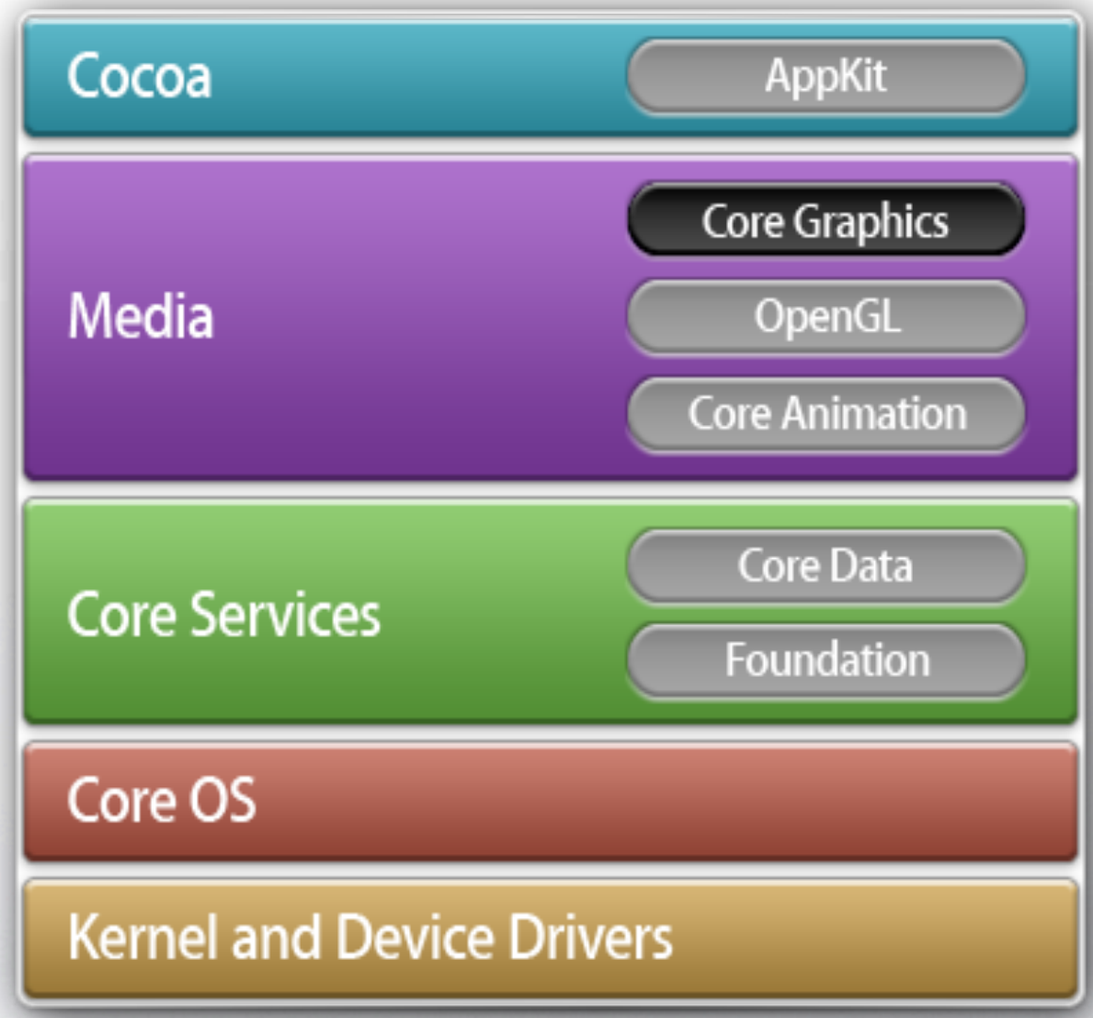
3D графика

Низкоуровневая 2D анимация

Низкоуровневая работа с данными

Низкоуровневая разработка

Ядро и драйвера устройств



ООП-Wrapper (оболочка)

Прямоугольник

C (Core Graphics)	Objective-C

`CGRect rect = CGRectMake(`
`  x, y,`
`  width, height`
`  );`
 	
`CGContextFillRect  (context, rect);`
`CGContextStrokeRect(context, rect);`
	`[canvas`
	`  drawRectangleAtX:x`
	`  y:y`
	`  width:width`
	`  height:height];`

ООП-Wrapper

Ломаная

C	Objective-C

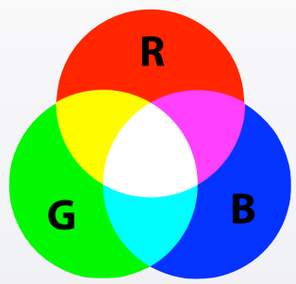
`CGContextAddLines(`
`  context,`
`  points,`
`  size`
`  );`
	`[canvas`
	`drawLinesThroughPoints:points`
	`  size:size];`

ООП-Wrapper

Класс цвета в формате RGBA
для совметсимости
между массивом пикселей и контекстом

C

```
data[i + 0] = color.red;
data[i + 1] = color.green;
data[i + 2] = color.blue;
data[i + 3] = color.alpha;
```



Objective-C

```
@interface TXColor
...
@end

txSetPixel(x, y, c);
```

ООП-Wrapper

Objective-C-интерфейс позволяет использовать парадигмы объектно-ориентированного программирования при работе с низкоуровневыми фреймфорками

Установка шрифта и цвета линий

C	Objective-C
`CGContextSetLineWidth(context, width);`	`[canvas setLineWidth:width];`
`CGContextSelectFont(context, font, size, kCGEncodingMacRoman);`	`[canvas selectFont:font withSize:size];`

Возможности API TX Mac

- Методы для отображения двухмерных объектов: от текста до геометрических фигур
- Гибкая система изменения параметров
- Способы создания графических контекстов с возможностью автоматического или ручного освобождения
- Средства для рендеринга изображений
- Возможность блокировать автообновление холста

C-Wrapper

Создание контекстов

Core Graphics	TX Mac

`CGContextRef context =`
`  CGBitmapContextCreate(`
`  NULL,`
`  width,`
`  height,`
`  bits,`
`  width * bits / 2,`
`  colorSpace,`
`  kCGImageAlphaPremultipliedLast|`
`  kCGBitmapByteOrder32Big);`
	`txCreateCompatibleDC(sizeX,`
	`  sizeY);`

C-Wrapper

Рендеринг изображений

Core Graphics	TX Mac

`CGAffineTransform transform =`
`  CGAffineTransformMake(1,0,0,-1,0,height);`
 	
`CGContextConcatCTM(context, transform);`
 	
`CGContextDrawImage(context, dest, image);`
 	
`CGContextConcatCTM(context, transform);`
	`txDrawImage(context,`
	`image,`
	`dest`
	`);`

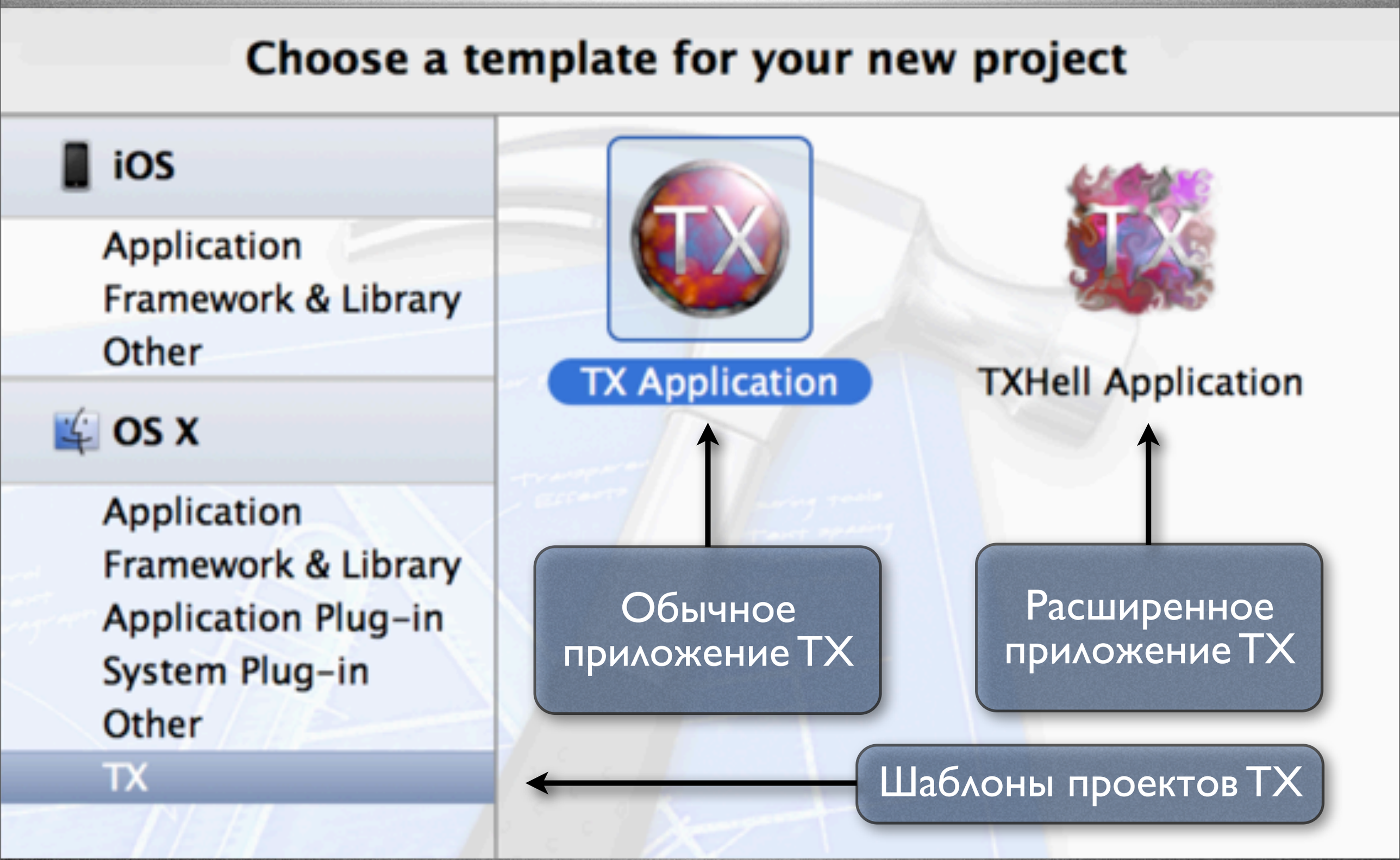
C-Wrapper

C-интерфейс позволяет сильно упростить сложный код, написанный как на C, так и на Objective-C.

Установка цвета

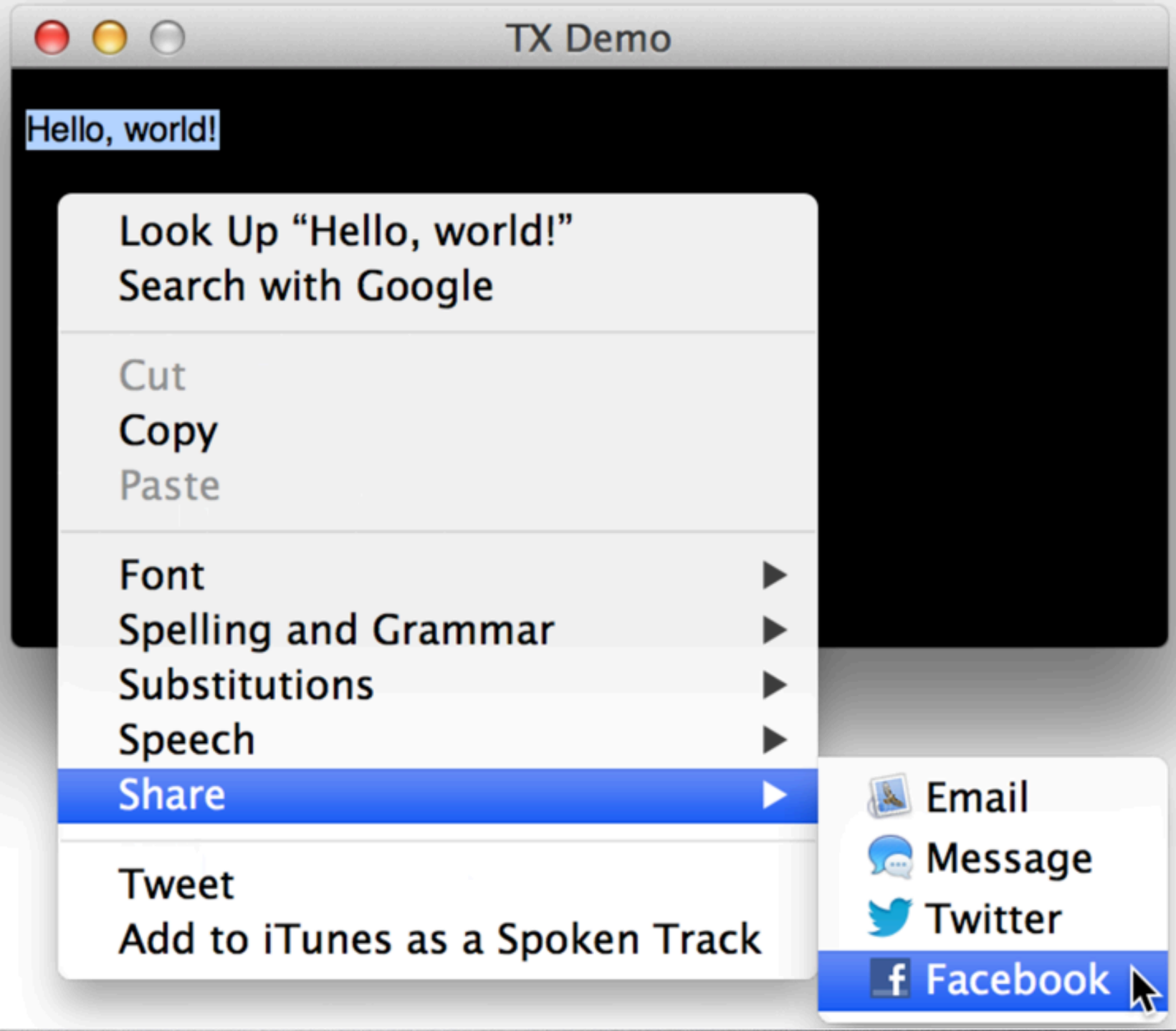
C & Objective-C	TX Mac
`CGContextSetRGBStrokeColor(self.context, red, green, blue, alpha);` `self.currentColor = [[TXColor alloc] initWithRed:red * 255 green:green * 255 blue:blue * 255 alpha:alpha * 255];`	`txSetColor(color);`

Создание проекта



Пример прогрммы “Hello, world”

```
// Main header of TX.framework

#include "TX/TXLib.h"

// main like in C++

int main() {

//  The simplest program

    cout << "Hello, world!";

    return 0;
}

// 0_o

// Facebook,
// Twitter,
// Google,
// Apple VoiceOver,
// Apple Dictionary,
// iMessage
//     integrated...

// Share your errors
//     Any time you like
```



Пример прогрммы TXHell

```
int main() {
    void (^distortions) // Image processing through TXHell
          (TXImage *, int, int) =
         ^(TXImage *image, int x, int y) {
         *image =
             txCIBumpDistortionFilter(
                 *image,
                 txVectorMake(
                     460 + rand () % 20,
                     50  + rand () % 20),
                 @120,
                 @0.7);
         *image =
             txCIBumpDistortionFilter(
                 *image,
                 txVectorMake(
                     860 + rand () % 20,
                     50   + rand () % 20),
                 @100,
                 @0.7);
    };
    ...
    drawLogo(txlogo, logoSize, value, distortions); // Rendering through TXMac
    ...
}
```


Результаты

- Реализован фреймворк-wrapper для облегчения попиксельного рендеринга 2D графики, а также API для упрощения:
  - Консольного ввода и вывода
  - Получения состояний мыши и клавиатуры
  - Взаимодействия с операционной системой OS X 10.8

Спасибо за внимание!

Источники

- https://developer.apple.com/devcenter/mac/index.action
- https://developer.apple.com/library/mac/navigation/
- http://www.wikipedia.org/
- https://itunesu.itunes.apple.com/WebObjects/LZDirectory.woa/ra/1/directory/courses/593208016/feed
- http://www.stackoverflow.com/