

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
Кафедра компьютерного моделирования и многопроцессорных систем

Рычков Андрей Евгеньевич

**Выпускная квалификационная работа бакалавра**

iOS библиотека с открытым исходным кодом для  
редактирования PDF файлов

Направление 010300

Фундаментальная информатика и информационные технологии

Научный руководитель,  
доктор физико-математических наук,  
профессор  
Андрианов С. Н.

Санкт-Петербург

2016

# Содержание

Содержание .....	2
Введение .....	3
Формализация цели и задач .....	5
Обзор существующих решений .....	6
Глава 1. Отображение PDF .....	9
1.1. Основные принципы отображения PDF .....	9
1.2. Интерфейс и основные элементы управления .....	10
Глава 2. Редактирование PDF .....	14
2.1. Добавление аннотаций на страницы документа .....	14
2.2. Изменение порядка страниц в документе .....	19
2.3. Смена ориентации страниц в документе .....	20
Глава 3. Предлагаемые методы и инструменты .....	22
Заключение .....	24
Список литературы .....	25
Приложение .....	26

# Введение

Часто в наше время программисты используют готовые решения каких-либо задач для ускорения процесса разработки. Такими решениями являются библиотеки, предоставляемые другими людьми. Одними из главных различий между этими библиотеками являются способы распространения и открытость исходного кода. Больше всего ценятся так называемые “Open source<sup>1</sup>” проекты. Вы можете абсолютно бесплатно воспользоваться чужим решением, оценить его, поблагодарить автора за время, которое он потратил на создание этого продукта для других людей. Или, если вам не хватает его возможностей, можно самим дополнить проект. Одним из ярких примеров является ядро операционной системы Linux. Без преувеличения, каждый день различные обновления добавляются к исходному проекту, который был создан больше 20 лет назад.

Благодаря поддержке общества приложения с открытым исходным кодом получают постоянные обновления, которые приносят новый функционал или исправляют существующие ошибки. Таким образом, при условии актуальности проблемы, решаемой приложением, оно получает широкую огласку в IT обществе и равнодушие среди программистов-пользователей, которые хотят интегрировать это решение в свой проект. Такие приложения имеют меньше недоработок, так как активно поддерживаются не только ограниченной службой технической поддержки в случае закрытого приложения, а частью программистского общества.

Файлы в формате PDF<sup>2</sup> используются для представления документов по всему миру, и часто возникает необходимость в их редактировании как для

---

<sup>1</sup> Open source (Свободное программное обеспечение) – программное обеспечение, пользователи которого имеют права («свободы») на его неограниченную установку, запуск, свободное использование, изучение, распространение и изменение (совершенствование), а также распространение копий и результатов изменения.

<sup>2</sup> PDF (Portable Document Format) - межплатформенный формат электронных документов, разработанный фирмой Adobe Systems.



Рис. 1. Диаграмма распределения файлов по форматам в мировом интернете

личных целей (например, добавление различных заметок на страницы), так и для бизнеса (электронная подпись, которую можно визуально демонстрировать на странице).

Согласно источникам [1][2] по данным за июль 2013 года формат PDF является основным для значительной части файлов, доступных в мировом интернете. Как показано на Рисунке 1, ни один из форматов не близок по популярности к PDF.

Многие люди предпочитают мобильные устройства компьютерам при просмотре документов, так как они легче и удобнее в использовании. Для мобильных устройств существуют различные программы для просмотра и редактирования файлов в формате PDF, но все они имеют закрытый исходный код.

## Формализация цели и задач

**Целью** данной работы является создание библиотеки для редактирования PDF файлов на операционной системе iOS.

Основные задачи, которые необходимо решить для достижения поставленной цели:

- 1) реализовать возможность постраничного просмотра PDF-документов;
- 2) создать элементы управления для быстрого перемещения по документу;
- 3) адаптировать размещение элементов пользовательского интерфейса ко всем существующим разрешениям экранов мобильных устройств под управлением операционной системы iOS;
- 4) оптимизировать скорость работы приложения на больших файлах;
- 5) обеспечить плавность анимаций и отзывчивость пользовательского интерфейса;
- 6) реализовать следующие возможности по редактированию PDF документов:
  - 1) добавление аннотаций разных видов на страницы документа;
  - 2) отмена и восстановление последних действий по добавлению аннотаций;
  - 3) изменение существующего порядка страниц в документе;
  - 4) поворот страниц документа на 4 различных угла (0, 90, 180 и 270 градусов);
- 7) выложить на открытые ресурсы исходный код приложения;
- 8) предоставить инструменты для быстрой и легкой интеграции в проекты;
- 9) обеспечить доступность приложения для изменений со стороны сообщества.

## Обзор существующих решений

Существует множество библиотек для просмотра и редактирования PDF-файлов на операционной системе iOS, однако у большинства из них присутствуют один или даже два значительных недостатка: платная лицензия (для использования некоторых вариантов требуется подписка) или закрытость исходного кода. Платность обуславливается тем, что компания, предоставляющая данный продукт, самостоятельно занимается его поддержкой и дальнейшими доработками. Присутствие исходного кода является одной из главных причин выбора той или иной библиотеки. Чаще всего они поставляются в бинарном формате вместе с заголовочными файлами<sup>1</sup>. Это позволяет вам добавлять некоторую ограниченную функциональность, так называемые “расширения”, к возможностям библиотеки в вашем приложении, но если вы имеете в своем распоряжении исходный код, то вы можете не только добавлять новый функционал, но и улучшать и переделывать отдельные места.

Что касается производительности, то все изученные мною решения демонстрируют плавность работы даже при просмотре и редактировании документов с более чем 1000 страницами [3].

Все рассмотренные мною платные варианты имеют один очень важный недостаток – цена. Она начинается примерно с 1000\$ за использование библиотеки в одном проекте. Ими нельзя пользоваться в учебных или научных целях без покупки лицензии. Ни одна из рассмотренных мною платных библиотек не позволяет производить перестановку и поворот страниц документа.

---

<sup>1</sup> Заголовочный файл – файл, содержимое которого автоматически добавляется препроцессором в исходный текст в том месте, где указана нужная директива.

Отдельно выделяются две библиотеки с открытым исходным кодом.

PDF Reader Core [4] – отличный выбор для чтения PDF файлов. Предоставляет элементы управления для быстрого перемещения по документу и предварительного просмотра коллекции страниц. Инструменты для редактирования файлов не предусмотрены.

LazyPDFKit [5] построен на основе кодовой базы PDF Reader Core. Предлагает инструменты только по добавлению различных аннотаций на страницы документа. Изначально задумывался автором как проект с закрытым исходным кодом, но, обнаружив огромное количество ошибок, он открыл доступ к нему и прекратил поддерживать проект.

Преимущества реализации MNKPDFEditor перед бесплатными аналогами:

- 1) Актуальный на текущий момент код, использующий современные методы разработки приложений.
- 2) Возможность поворота страниц документа.
- 3) Возможность перестановки страниц документа

Преимущества MNKPDFEditor перед платными аналогами:

- 1) Открытость исходного кода.
- 2) Поддержка со стороны сообщества.
- 3) Возможности поворота и перестановки страниц.
- 4) Бесплатность.
- 5) Доступность обсуждения нововведений и исправлений.
- 6) Возможность взять библиотеку за основу своей собственной и разработать совершенно новый продукт.

В Таблице 1 приведены основные сходства и различия между MNKPDFEditor и существующими аналогами в категории просмотра и редактирования PDF файлов.

	Fast PDFKit [6]	PS PDFKit [7]	FOXIT emb. PDF [8]	PDF Touch SDK [9]	PDF Reader Core	Lazy PDFKit	MNK PDF Editor
Включает исходный код	Нет	Нет	Нет	Да	Да	Да	Да
Включает тех. поддержку	Да	Да	Да	Да	Нет	Нет	Нет
Лицензия	Разовая	Разовая	Разовая	Разовая	MIT	MIT	MIT
Аннотации	Да	Да	Да	Да	Нет	Да	Да
Изменение порядка страниц	Нет	Нет	Нет	Нет	Нет	Нет	Да
Поворот страниц	Нет	Нет	Нет	Нет	Нет	Нет	Да
Текстовый поиск	Да	Да	Да	Да	Нет	Нет	Нет
Обновления	Регулярно	Регулярно	Регулярно	Регулярно	Обновляет общество	Нет поддержки	Обновляет общество
Цена	от 990€	На основе договора	На основе договора	от 999\$	Бесплатно	Бесплатно	Бесплатно

Таблица 1. Сравнение MNKPDFEditor с аналогами



# Глава 1. Отображение PDF

## 1.1. Основные принципы отображения PDF

Библиотека позволяет постранично просматривать PDF файлы, осуществляя навигацию между страницами с помощью жеста перетаскивания, увеличивать текст и изображения в несколько раз с помощью жеста щипка двумя пальцами с последующей навигацией по увеличенной странице, при этом векторные элементы на странице изменяют свой размер без потери качества.

Страница, которую требуется отобразить на экране, разбивается на плитки одинакового размера [10][11]. Каждая плитка отвечает за отображение определенной части страницы, асинхронно<sup>1</sup> выполняя операции рисования в фоновых потоках, что позволяет сохранять плавность работы программы даже при загрузке страниц с большим количеством контента.

При необходимости (например, при изменении масштаба просмотра страницы) те плитки, которые являются видимыми в рамках экрана устройства, получают команду, требующую обновить их содержимое в следующем цикле рисования.

Даже на современных мобильных устройствах отрисовка страницы с большим количеством содержимого в хорошем качестве занимает в среднем несколько секунд, а пользователь не привык ждать отображения контента, поэтому необходимо заполнить это время либо вставив индикатор ожидания, либо отобразить контент в худшем качестве, чем оригинал.

Для этого в библиотеке используется механизм кэширования<sup>2</sup>. Когда страница должна появиться на экране первый раз, в фоновом потоке создается

---

<sup>1</sup> Асинхронное событие (в информатике) – событие, которое возникает вне зависимости от основного потока выполнения программы.

<sup>2</sup> Кэширование – временное сохранение данных, которые с большой вероятностью могут быть запрошены в ближайшее время.

её копия в плохом качестве, которая отрисовывается гораздо быстрее оригинала, сохраняется во временное хранилище на диске и показывается пользователю. Затем плитки постепенно рисуют своё содержимое поверх копии. В следующий раз при открытии этой страницы будет произведена проверка существования копии в плохом качестве, и если она будет присутствовать, то мгновенно отобразится на экране. Если же временное хранилище будет очищено системой (по причине нехватки места) или пользователем (при очистке файлов), то процесс повторится заново.

Кроме того, сами плитки используют кэширование своего содержимого на системном уровне, что позволяет также гораздо быстрее демонстрировать пользователю конечное изображение требуемого качества.

Также показ изображения в плохом качестве подходит для быстрого изменения масштаба страницы. Как было указано выше, плитке необходимо некоторое время, чтобы обновить своё содержимое в векторном формате и отрисовать его. Поэтому нужные плитки становятся полностью прозрачными, и вместо них отображаются части худшего изображения.

Таким образом, пользователь при повседневном использовании приложения не будет замечать относительно долгой отрисовки страниц. При быстром перемещении между страницами плохое качество не заметно, тогда как все страницы наполнены контентом.

## **1.2. Интерфейс и основные элементы управления**

Пользовательский интерфейс программы состоит из различных кнопок, надписей и индикаторов. Расчет местоположения и размера каждого элемента ведется в независимых от устройства единицах измерения – точках. Это позволяет создавать универсальную разметку для всех видов экранов. Например, количество пикселей на экране iPhone 3GS совпадает с количеством точек: 320x480. У iPhone 4/4S разрешение экрана равно 640x960 пикселей, но количество точек остается тем же – 320x480.

Библиотека поддерживает различные ориентации устройства – портретную и ландшафтную. При изменении ориентации (например, вызванном поворотом устройства) все элементы поменяют свои размеры пропорционально изменению высоты и ширины экрана. После окончания поворота каждому элементу будет отправлено сообщение о необходимости произвести повторные расчеты размеров и местоположения и отрисовать своё содержимое.

Основные элементы интерфейса представлены на рисунке 2.

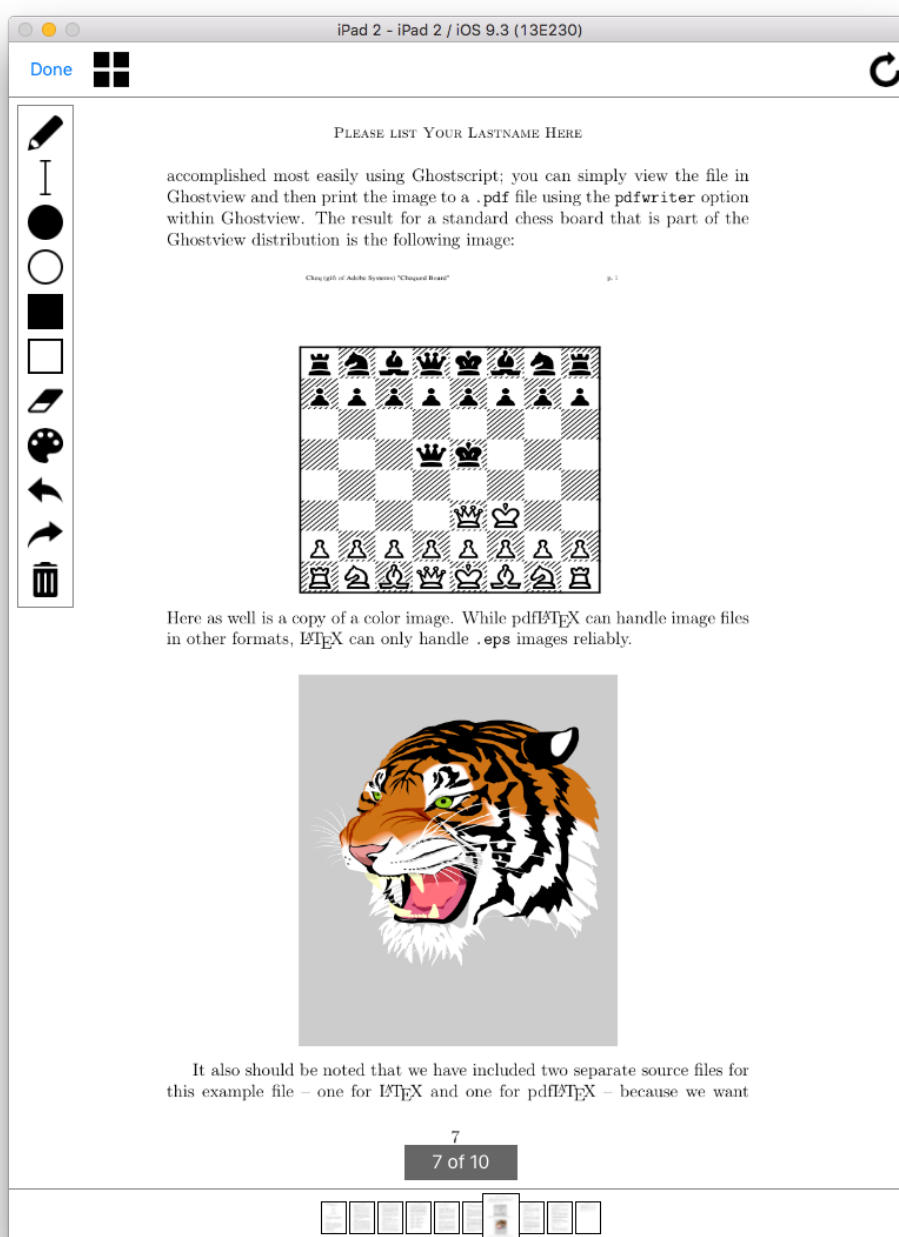


Рис. 2. Интерфейс программы

По центру, занимая большую площадь экрана, находится контент PDF страницы.

В нижней части расположен номер просматриваемой страницы и общее количество страниц в документе. Под ним находится элемент управления перемещением по документу, используемый для ускорения навигации. На нем отображаются уменьшенные копии страниц документа. Текущая страница выделяется увеличенным размером относительно остальных. Она выступает в роли “ползунка”, перемещая который пользователь может просматривать различные страницы документа. Количество страниц на нём либо соответствует количеству страниц в документе, либо если все они не умещаются на экране по ширине, то равно максимально возможному количеству умещающихся на нем страниц.

В правом верхнем углу расположена кнопка поворота текущей страницы на 90 градусов.

В левой части экрана находится панель с инструментами добавления аннотаций к странице. Нажатие на любой инструмент переведет программу в режим редактирования, в котором пользователь может добавлять, удалять и изменять аннотации к документу.

Кнопка “Done” (локализованная на системный язык устройства) отвечает за завершение сеанса работы с документом и закрывает его, при необходимости сохраняя на диск совершенные изменения, включающие в себя измененный порядок страниц, их поворот и нанесённые на них аннотации.

Кнопка с изображением 4 квадратов открывает экран просмотра коллекции страниц документа (Рисунок 3). Они отображаются в виде таблицы миниатюрных копий, каждая из которых по ширине занимает треть или четверть (в зависимости от ориентации устройства) экрана.

Здесь нажатие на любую страницу вернет пользователя на предыдущий экран полноэкранного просмотра выбранной страницы, а долгое нажатие на неё переведет таблицу в режим редактирования, и, путём перетаскивания

выбранного элемента, появляется возможность менять его местоположение в документе.



Рис. 3. Коллекция миниатюрных страниц документа

## Глава 2. Редактирование PDF

### 2.1. Добавление аннотаций на страницы документа

В библиотеке реализована возможность добавления различных аннотаций на страницы документа, таких как прямые линии, эллипсы и прямоугольники (заполненные и нет), а также произвольные фигуры. Для каждого типа можно выбрать цвет, толщину линии и прозрачность. В случае необходимости изменения уже добавленных аннотаций существуют инструменты отмены и возврата предыдущих действий, стирательная резинка и инструмент очистки.

Добавление аннотаций происходит на отдельный холст, который расположен поверх страниц документа. Изначально он прозрачный, и пользователь постепенно заполняет его аннотациями, используя представленные в программе инструменты.

Для поддержки реализации произвольных инструментов добавления аннотаций был создан протокол `MNKPFDrawingTool`, которому должен удовлетворять каждый класс, который хочет выступать в роли такого инструмента.

Основные положения этого протокола:

- 1) Класс должен иметь следующие переменные экземпляра:
  - 1) `lineColor`, используемая для установки цвета линии;
  - 2) `lineAlpha`, используемая для установки прозрачности линии;
  - 3) `lineWidth`, используемая для установки ширины линии;
- 2) Класс должен реализовать следующие методы экземпляра:
  - 1) `setInitialPoint(initial point)`, принимающий в качестве параметра начальную точку рисования и не возвращающий никаких переменных. Он предназначен для проведения необходимой подготовки инструмента к рисованию, например, установки пера в начальную точку или инициализации различных переменных.

- 2) `moveFromPoint(start point) toPoint(end point)`, принимающий в качестве параметров начальную и конечную точки рисования и не возвращающий никаких переменных. В реализации этого метода следует производить обновление переменных, отвечающих за координаты на холсте, и перемещение пера в конечную точку.
- 3) `draw()`, не принимающий никаких параметров и не возвращающий значений. Он вызывается при необходимости отображения текущей аннотации.

В библиотеке присутствуют стандартные инструменты:

- 1) **прямая линия.** Экземпляр класса запоминает начальную точку, а затем изменяет конечную точку в соответствии с движениями пальца пользователя и в методе `draw()` рисует сглаженную прямую линию между ними:

- 1) `CGContextRef context = UIGraphicsGetCurrentContext();`
- 2) `CGContextSetAlpha(context, self.lineAlpha);`
- 3) `CGContextSetLineWidth(context, self.lineWidth);`
- 4) `CGContextSetLineCap(context, kCGLineCapRound);`
- 5) `CGContextSetStrokeColorWithColor(context, self.lineColor.CGColor);`
- 6) `CGContextMoveToPoint(context, _startPoint.x, _startPoint.y);`
- 7) `CGContextAddLineToPoint(context, _endPoint.x, _endPoint.y);`
- 8) `CGContextStrokePath(context);`

- 2) **прямоугольник.** Экземпляр класса запоминает стартовую точку, затем в зависимости от изменения конечной точки рисует прямоугольник нужной формы и размера. Конечная точка служит для вычисления длины и ширины прямоугольника. В зависимости от выбранного типа заполняется либо контур, либо весь прямоугольник:

- 1) `CGContextRef context = UIGraphicsGetCurrentContext();`
- 2) `CGContextSetAlpha(context, self.lineAlpha);`

```

3) CGRect rectToDraw = CGRectMake(_startPoint.x,
    _startPoint.y, _endPoint.x - _startPoint.x, _endPoint.y -
    _startPoint.y);
4) if (self.isFilled) {
5)     CGContextSetFillColorWithColor(context,
        self.lineColor.CGColor);
6)     CGContextFillRect(context, rectToDraw);
7) } else {
8)     CGContextSetStrokeColorWithColor(context,
        self.lineColor.CGColor);
9)     CGContextSetLineWidth(context, self.lineWidth);
10) CGContextStrokeRect(context, rectToDraw);
11) }

```

3) эллипс. Использует тот же алгоритм, что и для построения прямоугольника, только эллипс вписывается в него:

```

1) CGContextRef context = UIGraphicsGetCurrentContext();
2) CGContextSetAlpha(context, self.lineAlpha);
3) CGRect rectToDraw = CGRectMake(self.startPoint.x,
    self.startPoint.y, self.endPoint.x - self.startPoint.x,
    self.endPoint.y - self.startPoint.y);
4) if (self.filled) {
5)     CGContextSetFillColorWithColor(context,
        self.lineColor.CGColor);
6)     CGContextFillEllipseInRect(context, rectToDraw);
7) } else {
8)     CGContextSetStrokeColorWithColor(context,
        self.lineColor.CGColor);
9)     CGContextSetLineWidth(context, self.lineWidth);
10) CGContextStrokeEllipseInRect(context, rectToDraw);
11) }

```

4) произвольное рисование. Для рисования произвольных фигур используются квадратичные кривые Безье<sup>1</sup>, так как даже при достаточно

---

<sup>1</sup> Кривая Безье - параметрическая кривая, строящаяся по нескольким опорным точкам и используемая в компьютерной графике [12].



хорошем приближении прямыми линиями угловатость их соединений остается сильно заметна.

Алгоритм построения произвольной фигуры (нижним подчеркиванием обозначены переменные экземпляра класса):

```
1. _previousPoint2 = _previousPoint1;
2. _previousPoint1 = _currentPoint;
3. _currentPoint = endPoint;
4. CGPoint mid1 = getMidPoint(_previousPoint1, _previousPoint2);
5. CGPoint mid2 = getMidPoint(_currentPoint, _previousPoint1);
6. CGMutablePathRef subpath = CGPathCreateMutable();
7. CGPathMoveToPoint(subpath, NULL, mid1.x, mid1.y);
8. CGPathAddQuadCurveToPoint(subpath, NULL, _previousPoint1.x,
    _previousPoint1.y, mid2.x, mid2.y);
9. CGRect boxNeedsToDisplay = CGPathGetBoundingBox(subpath);
10. boxNeedsToDisplay = CGRectInset(boxNeedsToDisplay, -
    self.lineWidth / 2, -self.lineWidth / 2);
11. _boxNeedsToDisplay = boxNeedsToDisplay;
12. CGPathAddPath(_path, NULL, subpath);
13. CGPathRelease(subpath);
```

Необходимо отметить, что если пользователь будет долго перемещать палец по экрану, то возможна ситуация, когда обновление и перерисовка всей кривой будут слишком ресурсозатратными операциями. Поэтому при очередном вызове функции `moveFromPoint(start point) toPoint(end point)` вычисляется минимальная рамка, в которую вмещается новый участок линии, который необходимо дорисовать.

Также библиотека предоставляет инструменты для управления состоянием полотна:

- 1) стирательная резинка. Использует тот же алгоритм, что и при произвольном рисовании, только у пера выставляется режим очистки содержимого холста;

- 2) палитра. При нажатии открывается экран выбора различных параметров текущего инструмента. В качестве параметров доступны толщина, прозрачность и цвет линии. После выставления параметра он сохраняется в настройках и соответствующая переменная у текущего инструмента устанавливается в это значение;
- 3) инструменты отмены и возврата изменений. Для ведения учёта вносимых изменений и их последующей обработки инициализируются два пустых массива указателей на инструменты, участвовавшие в рисовании: массив текущих инструментов `pathArray`, и массив буфера `bufferArray`, используемый для возврата отменённых действий. При добавлении аннотаций пользователем указатель на инструмент, которым он совершал это действие, добавляется в конец массива текущих инструментов.

Алгоритм отмены изменения:

1. `if (_pathArray.isEmpty) { return; }`
2. `_currentTool = nil;`
3. `id <MNKPDFDrawingTool>tool = _pathArray.lastObject;`
4. `_bufferArray.addObject(tool)`
5. `_pathArray.removeLastObject();`
6. `setNeedsDisplay();`

Алгоритм возврата изменения:

1. `if (_bufferArray.isEmpty) { return; }`
2. `_currentTool = nil;`
3. `id<MNKPDFDrawingTool>tool = [_bufferArray lastObject];`
4. `[_pathArray addObject:tool];`
5. `[_bufferArray removeLastObject];`
6. `[self updateImageWithRedrawing:YES];`
7. `[self setNeedsDisplay];`

- 4) Отмена всех изменений. Производит очистку массива текущих инструментов и массива буфера, после которой содержимое холста становится пустым, и он обновляется.

## 2.2. Изменение порядка страниц в документе

Перестановка страниц осуществляется путём долгого нажатия на плитку страницы на экране просмотра коллекции страниц документа и последующего переноса страницы в нужное место в коллекции. Стоит упомянуть, что iOS SDK<sup>1</sup> не предоставляет функции для анимированного перемещения элементов в коллекции. Ниже представлен алгоритм реализации перемещения:

- 1) произвести определение жеста долгого нажатия на элемент в коллекции, используя функции, предоставляемые iOS SDK;
- 2) сделать снимок (snapshot<sup>2</sup>) перемещаемого элемента;
- 3) создать на основе снимка новый вид и разместить его в координатной плоскости поверх этого элемента;
- 4) сделать выбранный элемент в коллекции полностью прозрачным, установив ему цвет с параметром альфа, равным нулю;
- 5) пока пользователь не отрывает палец от объекта на экране, производить следующие действия:
  - если центр перемещаемого элемента пересекает границу одного из элементов в коллекции, запустить анимацию смены скрытого прозрачного и перекрытого элементов;
  - если в момент проведения анимации из пункта А пользователь перекрыл другой элемент, отменить анимацию перемещения первого элемента, также анимированно вернуть его на начальное место. Прозрачный объект обменивается позициями с новым элементом;
  - если элемент был подведён к низу или верху вида с коллекцией, то рассчитывается пересечение границ экрана и этого элемента. Если в коллекции в соответствующем направлении есть контент для

---

<sup>1</sup> iOS SDK – комплект средств разработки для операционной системы iOS [13].

<sup>2</sup> Snapshot (снэпшот) – мгновенный снимок зафиксированного состояния элемента [14].

отображения, инициализируется прокрутка коллекции вверх или вниз со скоростью, обратно пропорциональной высоте прямоугольника пересечения, указанного выше;

После окончания работы с коллекцией страниц при необходимости внесенные изменения сохраняются на устройстве.

### 2.3. Смена ориентации страниц в документе

Как было указано выше, поворот страницы на 90 градусов осуществляется с помощью нажатия на кнопку со знаком закругленной стрелки.

Алгоритм поворота страницы (рисунок 4):

- 1) транслировать холст с контентом так, чтобы положение центра вращения совпало с началом координат. В случае страницы центром вращения является центр её рамки;
- 2) повернуть вид вокруг начала координат на заданный угол;
- 3) транслировать вид так, чтобы центр вращения вернулся в исходное положение;

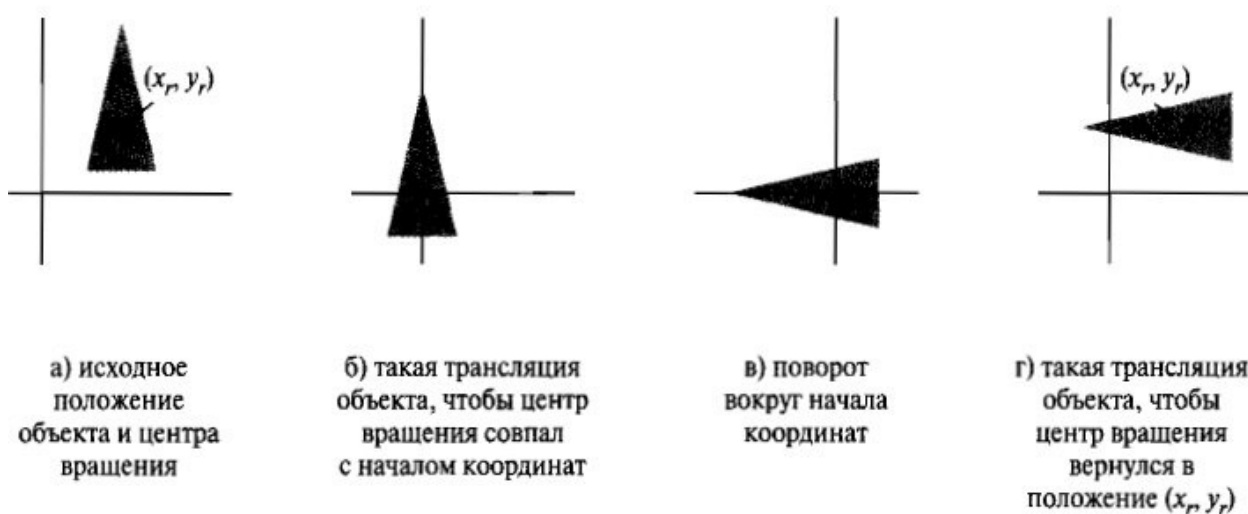


Рис. 4. Поворот объекта вокруг произвольной точки – центра вращения [15]

Математически операции геометрических преобразований представляются в однородных координатах, количество которых всегда на одну больше, чем размерность пространства. В данном случае удобно считать, что третья координата равна 1, тогда полные координаты точки равны  $(x, y, 1)$ . Двухмерные точки представляются трехэлементными векторами-столбцами, а двухмерные операции преобразования выражаются матрицами 3 на 3. Тогда все уравнения геометрических преобразований можно представить в форме матричного умножения, что и является стандартным методом, используемым в графических системах.

На Рисунке 5 проиллюстрировано матричное умножение, позволяющее повернуть объект вокруг его центра.  $(x_r, y_r)$  – координаты центра фигуры, а  $\theta$  – требуемый угол поворота.

$$\begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix} = \\ = \begin{bmatrix} \cos \theta & -\sin \theta & x_r(1 - \cos \theta) + y_r \sin \theta \\ \sin \theta & \cos \theta & y_r(1 - \cos \theta) - x_r \sin \theta \\ 0 & 0 & 1 \end{bmatrix},$$

Рис. 5. Последовательность матричных преобразований для поворота фигуры вокруг своего центра [15]

## Глава 3. Предлагаемые методы и инструменты

В качестве инструмента для реализации поставленной задачи был выбран язык программирования Objective-C, так как он создан компанией Apple специально для создания приложений для операционных систем iOS и Mac OS. Альтернативой Objective-C является новый язык программирования Swift, по сути являющийся логическим продолжением и улучшением первого. Но из-за постоянных обновлений и ошибок становится тяжело поддерживать продукт, и большинство программистов предпочитают использовать уже привычную устоявшуюся технологию, к тому же специализированная интегрированная среда разработки предоставляет возможность перевода практически любых синтаксических конструкций с одного языка на другой.

Для контроля за версиями проекта рассматривались два продукта – SVN и Git [16]. Первый является централизованной системой контроля версий. Есть центральный сервер, на котором находятся все файлы под “версионным” контролем, и ряд клиентов, которые получают от него копии файлов. У такого подхода присутствует значительный недостаток – при повреждении сервера вся система выходит из строя, а при потере данных их невозможно восстановить. В случае с Git клиенты не просто загружают последние версии файлов, а полностью копируют рабочую директорию. Поэтому в случае неисправности сервера, с помощью которого велись все версионные операции, клиенты имеют возможность самостоятельно вести контроль, а затем загружать свое состояние проекта обратно на сервер. После изучения достоинств и недостатков обоих вариантов было принято решение использовать Git.

Следующий шаг - выбор платформы для размещения исходного кода. Основные критерии выбора – бесплатность, интеграция в существующие среды разработки и распространенность. В качестве альтернатив BitBucket и GitHub. Первый вариант больше подходит для небольших команд, так как он предоставляет возможность бесплатно создавать приватные репозитории и

интегрироваться с менеджерами задач. GitHub, в свою очередь, представляет собой некую социальную сеть, предоставляя пользователям обширные возможности для общения, например, выставление звезд репозиториям, открытые обсуждения ошибок или доработок программ. Также, на данный момент, на GitHub зарегистрировано более 14 миллионов человек.

Для удобства и простоты интеграции библиотеки в существующие проекты используется менеджер зависимостей CocoaPods. На данный момент в его базе присутствует более 8000 библиотек, и она ежедневно пополняется. Существует специальный файл (Podfile), в котором прописываются в установленном формате необходимые зависимости проекта от сторонних библиотек. Таким образом, для использования MNKPDFEditor в проекте, нужно лишь вставить строку “pod MNKPDFEditor” в Podfile проекта и запустить процесс установки зависимостей.

## Заключение

Для достижения поставленной цели была изучена литература о программировании на операционной системе iOS и наложении различной графики на изображения. Был проведен анализ существующих библиотек для просмотра и редактирования PDF файлов на iOS, после которого были сформированы преимущества и недостатки этих решений и сформулирован список задач, необходимый для создания альтернативы с вышеуказанным списком преимуществ.

Все поставленные задачи были выполнены, результатом работы является библиотека с открытым исходным кодом, которая размещена на сервисе GitHub. Любой пользователь может абсолютно бесплатно пользоваться всем функционалом, предоставляемым API<sup>1</sup> библиотеки, а также добавлять новые возможности, исправлять найденные ошибки и обсуждать текущие и будущие изменения.

Дальнейшие улучшения включают в себя добавление поиска по документу, возможности работы с документами, защищенными паролем, а также различные улучшения интерфейса и работы с инструментами.

Цель работы достигнута, на защиту выносится следующее положение: Созданная библиотека позволяет просматривать и редактировать PDF файлы на устройствах под управлением операционной системы iOS.

---

<sup>1</sup> API (Application Programming Interface) – набор готовых классов, функций или процедур, предоставляемых приложением, библиотекой или сервисом для использования во внешних программных продуктах.



## Список литературы

- 1) Статистика распространения файловых форматов в интернете.  
[http://rusrim.blogspot.ru/2013/07/blog-post\\_16.html](http://rusrim.blogspot.ru/2013/07/blog-post_16.html)
- 2) Распространенность формата PDF в российских организациях.  
<http://www.itbestsellers.ru/news/detail.php?ID=28011>
- 3) Сравнение библиотек для просмотра и редактирования PDF файлов на iOS. <http://www.binpress.com/blog/2014/07/28/comparing-ios-pdf-libraries/>
- 4) PDF Reader Core. <http://www.vfr.org>
- 5) Lazy PDF Kit. <https://github.com/lazyprogram/LazyPDFKit>
- 6) Fast PDF Kit. <http://fastpdfkit.com>
- 7) PS PDF Kit. <https://pspdfkit.com>
- 8) FOXIT Embedded PDF. <https://www.foxitsoftware.com/products/sdk/pdf-sdk/>
- 9) PDF Touch Kit. <http://www.binpress.com/app/pdftouch-sdk/859>
- 10) CATiledLayer. Документация по использованию.  
[https://developer.apple.com/library/ios/documentation/GraphicsImaging/Reference/CATiledLayer\\_class/](https://developer.apple.com/library/ios/documentation/GraphicsImaging/Reference/CATiledLayer_class/)
- 11) PDF generation on iOS.  
<https://developer.apple.com/library/ios/documentation/2DDrawing/Conceptual/DrawingPrintingiOS/GeneratingPDF/GeneratingPDF.html>
- 12) Роджерс Д., Адамс Дж. Математические основы машинной графики. — М.: Мир, 2001.
- 13) Определение понятия iOS SDK. [https://ru.wikipedia.org/wiki/IOS\\_SDK](https://ru.wikipedia.org/wiki/IOS_SDK)
- 14) Определение понятия Snapshot. <https://ru.wikipedia.org/wiki/Snapshot>
- 15) Компьютерная графика и стандарт OpenGL, 3-е издание. : Пер. с англ. — М. : Издательский дом “Вильямс”, 2005. — 1168 стр., с ил.; ISBN 5-8459-0772-1
- 16) Pro GIT book. <https://git-scm.com/book/ru/v1/>
- 17) Core Graphics API reference. [https://developer.apple.com/library/ios/documentation/CoreGraphics/Reference/CoreGraphics\\_Framework/](https://developer.apple.com/library/ios/documentation/CoreGraphics/Reference/CoreGraphics_Framework/)

## Приложение

В данном приложении приводятся функции рисования, предоставляемые библиотеки Core Graphics [17] и используемые в данной работе:

- 1) `UIGraphicsGetCurrentContext()` используется для получения текущего контекста для рисования;
- 2) `CGContextSetRGBFillColor(context, R, G, B, A)` устанавливает цвет заполнения элементов в контексте;
- 3) `CGContextSetRGBStrokeColor(context, R, G, B, A)` устанавливает цвет контура элементов в контексте;
- 4) `CGContextSetLineWidth(context, width)` устанавливает толщину линии элементов в контексте;
- 5) `CGContextFillRect(context, rect)` заполняет указанный прямоугольник цветом, установленным ранее функцией 2;
- 6) `CGContextStrokeEllipseInRect()` заполняет вписанный в прямоугольник эллипс;
- 7) `CGPointMake(x, y)` создает точку с координатами (x, y);
- 8) `CGContextStrokeLineSegments(context, points, pointsCount)` заполняет контур ломаной линии, образованной точками из массива `points`;
- 9) `CGPathCreateMutable()` инициализирует
- 10) `CGContextBeginPath(context)` начинает произвольную фигуру, которая может состоять из нескольких составных;
- 11) `CGContextMoveToPoint(context, x, y)` перемещает перо в точку с координатами (x, y);
- 12) `CGContextAddCurveToPoint(context, x1, y1, x2, y2, xEnd, yEnd)` – Добавляет к текущей фигуре кривую Безье от текущего положения пера до точки с координатами (xEnd, yEnd), используя в качестве опорных точек точки с координатами (x1, y1), (x2, y2). После выполнения данной функции перо будет находиться в конечной точке;

13) `CGContextStrokePath(context)` заполняет контур текущей фигуры выбранным функцией 3 цветом;

Контекст представляет собой систему координат, которую можно изменять для создания поворота, масштабирования, сдвига (трансляции) и других комплексных эффектов для фигур.

Основные операции двумерной трансформации:

- 1) `CGContextRotateCTM(context, angle)` поворачивает систему координат контекста на заданный угол `angle` в радианах;
- 2) `CGContextScaleCTM(context, sx, sy)` масштабирует систему координат на заданные коэффициенты по оси `x` (`sx`) и `y` (`sy`);
- 3) `CGContextTranslateCTM(context, tx, ty)` сдвигает систему координат на `tx` пунктов по оси `x` и `ty` пунктов по оси `y`;

Более комплексные операции производятся путем последовательных применений различных последовательностей базовых операций трансформации.