**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра MO ЭВМ**

**КУРСОВАЯ РАБОТА**

**по дисциплине «Разработка приложений**

**для мобильных платформ»**

**Тема: Приложение-помощник для фотографа**

|  |  |  |
| --- | --- | --- |
| Студент гр. 4303 |  | Азаров А.И. |
| Преподаватель |  | Заславский М.М. |

Санкт-Петербург

2018

**Задание**

Студент

Азаров А.И.

Группа 4303

Тема проекта: Разработка приложения – помощника для фотографа.

Исходные данные:

Необходимо разработать приложение – помощник для фотографа, облегчающее его работу, беря на себя часть работы, которую должен сделать фотограф, чтобы получить хороший снимок.

Содержание пояснительной записки:

«Содержание»

«Введение»

«Качественные требования к решению»

«Сценарий использования»

«Модель данных»

«Разработка приложения»

«Вывод»

«Приложение»

Предполагаемый объём пояснительной записки:

Не менее 10 страниц.

Дата выдачи задания:

Дата сдачи реферата:

Дата защиты реферата:

|  |  |  |
| --- | --- | --- |
| Студент гр. 4303 |  | Азаров А.И. |
| Преподаватель |  | Заславский М.М. |

**Аннотация**

В рамках данного курса предполагалось разработать какое-либо приложение на одну из поставленных тем, или выбрать собственную тему. Была выбрана тема создания приложения-помощника для фотографа, облегчающее его работу, беря на себя часть работы, которую должен сделать фотограф, чтобы получить хороший снимок.

Найти исходный код и всю дополнительную информацию можно на github[[1]](https://github.com/moevm/adfmp18-photo-assistant) – странице проекта.

**ANNOTATION**

In the course an application was planned to develop for one of the themes or choose own individual theme. Photo assistant application has been chosen to develop. This application makes a work of a photographer easier, taking responsibilities photographer has to do in order to make a great photo for itself.

Source code and all the additional information can be found on the github page[[2]](https://github.com/moevm/adfmp18-photo-assistant) of the project.

**Оглавление**

[1. Введение 6](#_Toc512039360)

[2. Качественные требования к решению 6](#_Toc512039361)

[3. Сценарии использования 7](#_Toc512039362)

[4. Архитектура приложения 11](#_Toc512039363)

[5. Разработанное приложение 13](#_Toc512039364)

[6. Вывод 15](#_Toc512039365)

[7. Приложения 16](#_Toc512039366)

[8. Ссылки на внешние ресурсы 16](#_Toc512039367)

### Введение

Цель работы – создать приложение-помощник для фотографа, облегчающее его работу, беря на себя часть работы, которую должен сделать фотограф, чтобы получить хороший снимок.

### Качественные требования к решению

Должны быть реализованы следующие функции:

1. Выбрать один из фильтров для фотографии (сетка / спираль Фибоначчи).
2. Сделать фотографию.
3. Изменить вид камеры с передней на заднюю и наоборот.
4. Включить / выключить вспышку.
5. Показ линии горизонта.
6. Просмотр сделанной фотографии.

### Сценарии использования

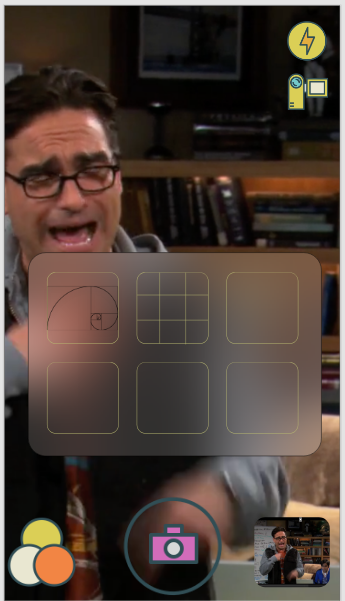
**Макеты UI**

1. Начальный экран (Рис. 1).

****

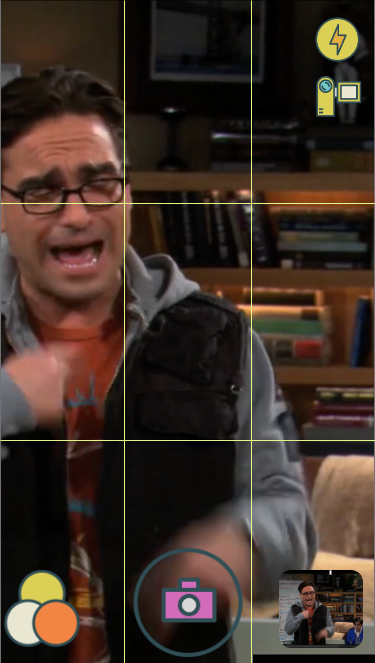
*Рисунок 1. Начальный экран.*

1. Экран выбора фильтра (Рис. 2).

****

*Рисунок 2. Экран выбора фильтров.*

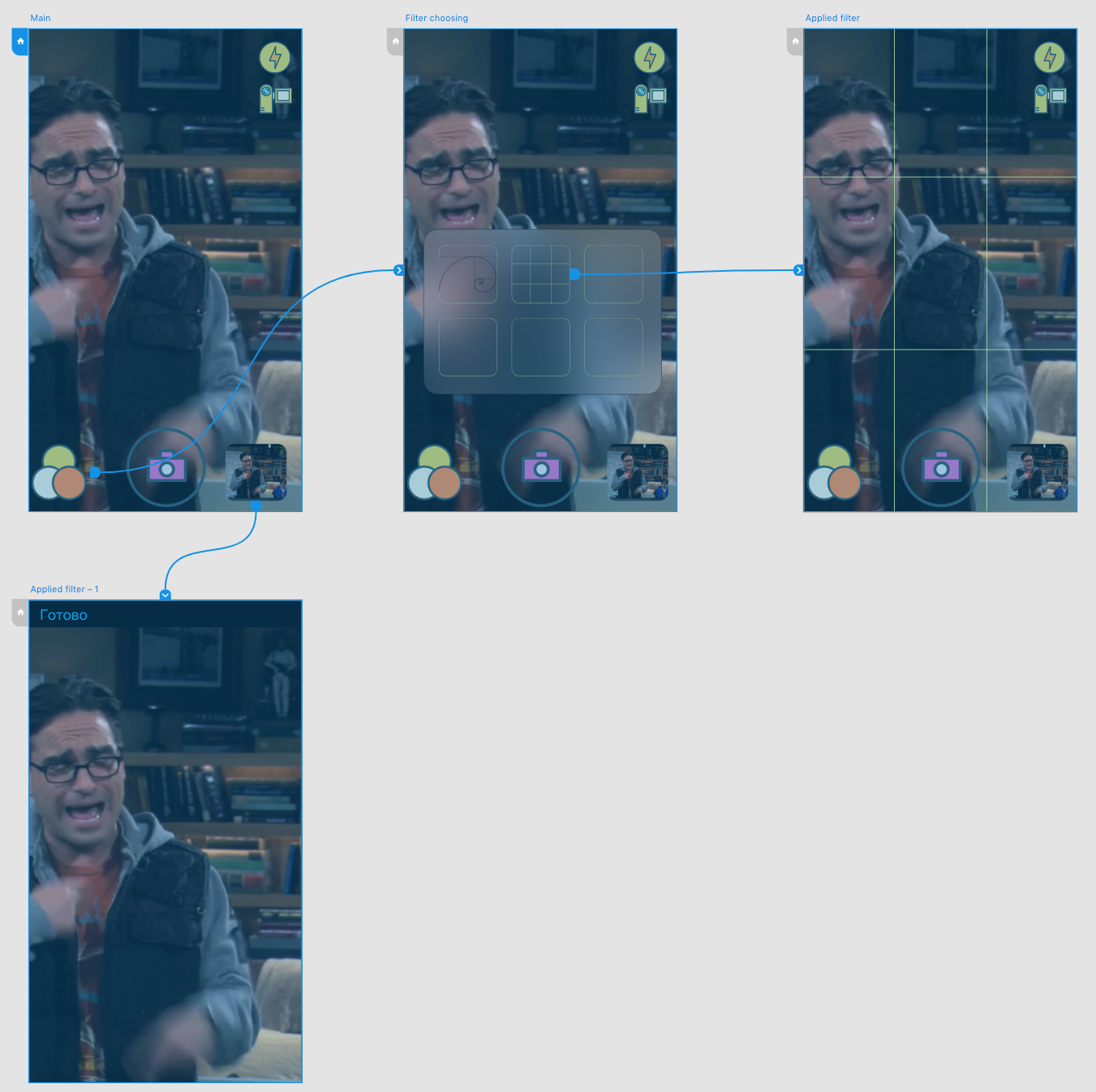
1. Экран с применённым фильтром (Рис. 3).

****

*Рисунок 3. Экран с применённым фильтром.*

**Схема экранов приложения**

Экраны приложения и переходы между ними отображены на рисунке 4.



*Рисунок 4. Схема экранов приложения.*

**Описание сценариев использования**

Следующие сценарии использования предполагают, что пользователь находится на начальном экране.

1.Сделать фото:

Нажать на кнопку «Сделать фото» (Иконка с изображением фотоаппарата)

2. Выбрать фильтр:

1. Нажать на иконку выбора фильтров

2. Выбрать интересующий фильтр.

3. Изменить вид камеры:

Нажать на кнопку с изображением видеокамеры.

4. Включить / выключить вспышку:

Нажать на кнопку с изображением вспышки.

5. Просмотреть сделанное фото:

Нажать на уменьшенное сделанное фото в правом нижнем углу экрана.

1. Архитектура приложения

В основе архитектуры приложения заложен шаблон проектирования VIP (Clean Swift), называемый так из-за первых букв компонентов, выполняющих ключевую роль ViewController – Interactor – Presenter. Общий вид данного шаблона проектирования представлен на рисунке 5.

*Рисунок 5. Схема VIP.*

В данном архитектурном шаблоне всё происходит вокруг так называемых сцен(экранов). Для каждой сцена создаётся 4 класса:

1. View Controller – отвечает за отображение данных и реагирует на действия пользователей.
2. Interactor – отвечает за бизнес логику приложения и хранение данных. Чтобы Interactor не получался слишком большим, за каждый сценарий использования отвечают специальные классы – Workers. Каждый из данных классов выполняет обработку данных для одного сценария использования.
3. Presenter – отвечает за форматирование данных для последующего отображения.
4. Router – отвечает за переход и передачу данных между экранами.

Общение между компонентами происходит посредством интерфейсов(далее протоколов) DisplayLogic, под который подписывается ViewController, BusinessLogic, под который подписывается Interactor и PresentationLogic, под который подписывается Presenter. Общение происходит в виде цикла, называемого VIP – циклом: VC конструирует Request модель с необходимыми данными и вызывает метод у Interactor. Interactor выполняет необходимые преобразования с данными, конструирует Response модель и вызывает метод у Presenter. Presenter форматирует данные, чтобы их было удобно отображать, конструирует ViewModel объект и вызывает метод у VC. VC отображает данные.

В приложении 2 экрана. Для них было создано 2 сцены: Photo и Camera

Структура сцена Camera:

* CameraViewController
* CameraInteractor

Workers:

* CameraConfigurator
* CaptureImageWorker
* DrawFiltersWorker
* KeepHorizonLineWorker
* ShowPreviewWorker
* SwitchCamerasWorker
* ToggleFlashlightWorker
* UpdateOrientationWorker
* CameraPresenter
* CameraRouter
* CameraModels

Структура сцена Photo:

* PhotoViewController
* PhotoInteractor
* PhotoPresenter
* PhotoModels
* PhotoRouter

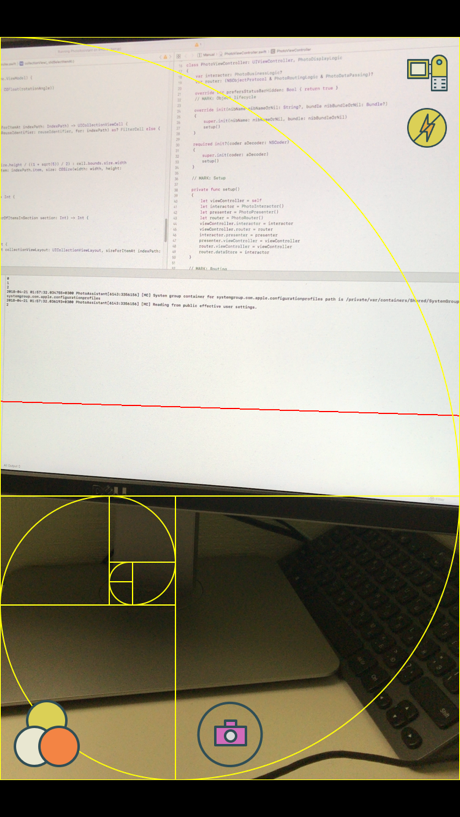
### Разработанное приложение

**Краткое описание**

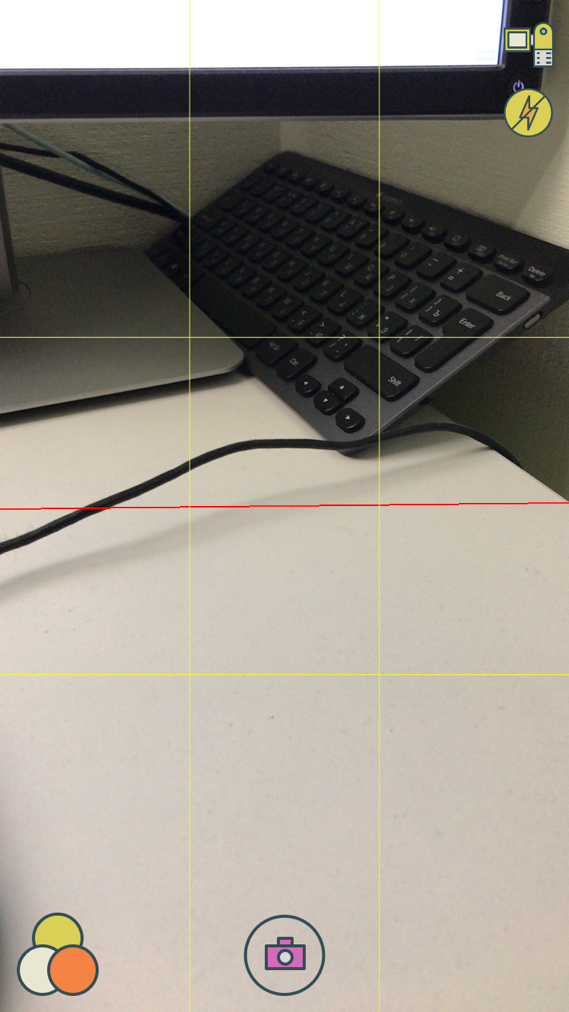
Приложение представляет из себя мобильное приложение-помощник для фотографа, для платформы iOS, написанное на языке программирования Swift. Приложение позволяет делать снимки, изменять вид камеры, включать вспышку, просматривать сделанную фотографию и выбирать фильтр для облегчения выбора композиции снимка. Также приложение показывает линию горизонта, чтобы сделать ровный снимок было просто.

**Скриншоты приложения**

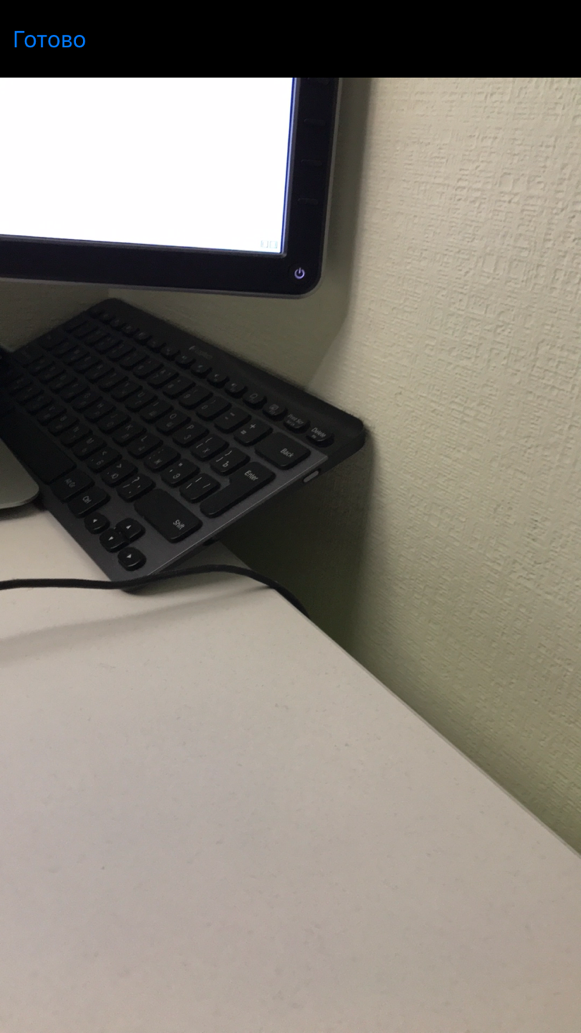
Применённый фильтр «Спираль Фибоначчи»



Применённый фильтр «Сетка»



Просмотр фотографии



Горизонтальный режим



**Использованные технологии**

1. Swift – язык программирования под iOS.
2. UIKit – библиотека графического интерфейса.
3. AVFoundation – библиотека для работы с камерой и фотографиями.
4. CoreMotion – библиотека для работы с сенсорами устройства.
5. CoreGraphics – библиотека для рисования.

### Вывод

**Результаты**

В ходе работы было разработано приложение-помощник для фотографа, позволяющее делать более качественные снимки, используя фильтры и линию горизонта для выстраивания хорошей композиции.

**Недостатки и пути для улучшения полученного решения**

На данный момент в приложении нет недостатков с технической точки зрения, но есть вещи, которые можно улучшить. Вот некоторые из них:

1. С помощью анализа изображения сделать так, чтобы фильтр «сетка» подсвечивалась зелёным цветом при удачном ракурсе.

\* удачный ракурс - такой ракурс, при котором в прямоугольник фильтра «сетка» попадает целый объект,а не его часть.

2. Добавить возможность фокусироваться в определённом месте по нажатию на это место на экране

3. Добавить возможность просматривать фотографию, используя «3D touch» на иконку сделанной фотографии.

Данное нововведение сократит количество действий, выполняемых для просмотра сделанной фотографии: раньше нужно было нажать на иконку с изображением, а затем на кнопку «готово», чтобы закрыть изображение. Теперь же нужно «сильно нажать» на иконку изображения, чтобы просмотреть изображение, и убрать палец с экрана, чтобы изображение исчезло.

4. Добавить возможность использовать кнопки регулировки громкости для того, чтобы сделать снимок.

5. Добавить возможность пользователю самому настраивать фильтр «сетка» (менять размер, количество «прутьев» сетки).

6. Добавить возможность автоматического включения вспышки в зависимости от освещённости в помещении/на улице.

**Будущее развитие решения**

Планируется реализовать больше фильтров.

### Приложения

**Документация по сборке и развертыванию приложения**

1. Скачать проект из репозитория (указан в ссылках на внешние ресурсы)
2. Запустить проект внутри XCode.

### Ссылки на внешние ресурсы

1. Исходники проекта: [https://github.com/moevm/adfmp18-photo-assistant](https://github.com/moevm/adfmp18-poker)