

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«Санкт–Петербургский государственный университет
аэрокосмического приборостроения»

Кафедра №43 «Компьютерных технологий и программной инженерии»

ОТЧЁТ ПО ПРАКТИКЕ
ЗАЩИЩЁН С ОЦЕНКОЙ
РУКОВОДИТЕЛЬ

Ст. преподаватель

С.А. Рогачев

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЁТ ПО ПРАКТИКЕ

вид практики	производственная
тип практики	по получению профессиональных умений и опыта профессиональной деятельности
на тему индивидуального задания	«Алгоритм поворота изображения на произвольный угол»

выполнен	Корнющенковым Кириллом Алексеевичем
	фамилия, имя, отчество обучающегося в творительном падеже

по направлению подготовки	09.03.04	Программная инженерия
	код	наименование направления

	наименование направления	
направленности	09.03.04.01	Разработка программно-информационных систем
	код	наименование направленности

наименование направленности

Обучающийся группы №	4831	К.А.Корнющенко
	номер	инициалы, фамилия
	подпись, дата	

Санкт–Петербург 2020

Оглавление

1. ЦЕЛЬ РАБОТЫ:	3
2. ИСХОДНЫЕ ДАННЫЕ.....	3
3. ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ	5
4. ПРАКТИЧЕСКИЙ РАЗДЕЛ	5
5. РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ	5
6. ВЫВОДЫ	6
7. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	7
ПРИЛОЖЕНИЕ А	7

1. Цель работы

Целью данной работы является разработка алгоритма поворота изображения на произвольный угол.

Вариант 16:

16	Алгоритм поворота изображения на произвольный угол
----	--

2. Исходные данные

Исходными данными для разработанного приложения являются изображения, хранящиеся в галереи симулятора телефона.

Было выбрано 3 изображения:



Изображение 1



Изображение 2



Изображение 3

3. Теоретический раздел

Разработанное приложение работает под управлением фреймворка UIKit. При написании программы использовались как библиотеки и ресурсы, предоставляемые языком Swift, так и средства фреймворка UIKit. Для разработки приложения были использованы такие классы фреймворка UIKit, как UIImageView, UISlider, UIButton, UILabel и другие.

Приложение состоит из одного ViewController-a, на котором размещаются UI элементы. Пользователь выбирает необходимое ему изображение из галереи и при помощи слайдера выбирает угол поворота. Слайдер принимает значения от 0 до 360, после поворота пользователь может сохранить полученное изображения при помощи кнопки Save.

4. Практический раздел

Смотреть Приложение А

5. Результаты работы программы



Изображение 1 повернутое на 230 градусов



Изображение 2 повернутое на 5 градусов



Изображение 3 повернутое на 180 градусов

6. Выводы

В ходе выполнения данного проекта был разработан алгоритм поворота изображения на произвольный угол. Разработанное приложение работает под управлением фреймворка UIKit. При написании программы использовались библиотеки и ресурсы, предоставляемые языком Swift.

7. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) Ray Wenderlich. Swift Apprentice: 2018 - 342 стр.
- 2) Ray Wenderlich. IOS Apprentice: 2020 - 656 стр.
- 3) *Василий Усов: Swift. Основы разработки приложений под iOS и macOS: 2019 - 448 стр.*
- 4) Official Swift documentation: <https://www.apple.com/ru/swift/>
- 5) Ray Wenderlich. Design Patterns by Tutorials: 2019 - 364 стр.

Приложение А

```
import UIKit

class ViewController: UIViewController {

    //MARK:Var

    @IBOutlet weak var mainImageView: UIImageView!

    @IBOutlet weak var degreeSlider: UISlider!

    @IBOutlet weak var saveButton: UIButton!

    @IBOutlet weak var degreeLabel: UILabel!

    var mainImageTapGestureRecognizer:UITapGestureRecognizer!

    var startPossitionImageView:CGAffineTransform = CGAffineTransform()

    var flagImageStart:Bool = false


    //MARK:Lifecycle

    override func viewDidLoad() {

        super.viewDidLoad()
```

```
        setup()

    }
}
```

```
//MARK:Action
```

```
@IBAction func saveButton(_ sender: UIButton) {

    let image = makeScreenshot()

    Sharing.share(on: self,image: image)

}
```

```
@objc func EndChangeSlider(){

    UIImageView.animate(withDuration: 1, animations: {

        self.mainImageView.transform =
self.mainImageView.transform.rotated(by: CGFloat(self.degreeSlider.value) * .pi /
180 )

    })

}
```

```
@objc func ChangeSlider(){

    degreeLabel.text = String(Int(degreeSlider.value))

}
```



```
}
```

```
@objc func tapMainImageView(){
```

```
    if flagImageStart{
```

```
        UIImageView.animate(withDuration: 1, animations: {
```

```
            self.mainImageView.transform = self.startPositionImageView
```

```
        })
```

```
        UISlider.animate(withDuration: 1) {
```

```
            self.degreeSlider.value = 0
```

```
        }
```

```
        degreeLabel.text = "0"
```

```
        DispatchQueue.main.asyncAfter(deadline: .now() + .seconds(1)) {
```

```
            self.alertImage()
```

```
        }
```

```
    }else{
```

```
        alertImage()
```

```
        flagImageStart = true
```

```
    }
```

```
}
```

```
}
```

```
//MARK: Helpers func
```

```
extension ViewController{
```

```
    private func setup(){
```

```
        degreeLabel.text = "\(Int(degreeSlider.value))"
```

```
        degreeSlider.addTarget(self, action: #selector(EndChangeSlider), for:  
        .touchUpInside)
```

```
        degreeSlider.addTarget(self, action: #selector(ChangeSlider), for:  
        .valueChanged)
```

```
        mainImageTapGestureRecognizer = UITapGestureRecognizer(target:  
        self, action: #selector(tapMainImageView))
```

```
        mainImageView.addGestureRecognizer(mainImageTapGestureRecognizer)
```

```
        startPossitionImageView = self.mainImageView.transform
```

```
        saveButton.alpha = 0
```

```

        saveButton.isHidden = false

    }

    private func makeScreenshot() -> UIImage{

        UIGraphicsBeginImageContextWithOptions(CGSize(width:
view.bounds.width,height: view.bounds.height),false,0)

        self.view.drawHierarchy(in: view.bounds, afterScreenUpdates: true)

        let image:UIImage = UIGraphicsGetImageFromCurrentImageContext()!

        UIGraphicsEndImageContext()

        return image

    }

}

//MARK: Load Photo

extension ViewController: UIImagePickerControllerDelegate,
UINavigationControllerDelegate{

    private func alertImage(){

        let alert = UIAlertController(title: nil, message: "Выберите способ
загрузки изображения", preferredStyle: .alert)

```

```

        let cameraAction = UIAlertAction(title: "Camera", style: .default) {
(action) in

            self.chooseImage(source: .camera)

        }

        let photoLibAction = UIAlertAction(title: "Photo", style: .default) {
(action) in

            self.chooseImage(source: .photoLibrary)

        }

        let cancelAction = UIAlertAction(title: "Cancel", style: .cancel, handler:
nil)

        alert.addAction(cameraAction)

        alert.addAction(photoLibAction)

        alert.addAction(cancelAction)

        self.present(alert,animated: true,completion: nil)

    }

```

```

private func chooseImage(source:UIImagePickerController.SourceType){

    if UIImagePickerController.isSourceTypeAvailable(source){

        let imagePicker = UIImagePickerController()

        imagePicker.delegate = self
    }
}

```

```

        imagePicker.allowsEditing = true

        imagePicker.sourceType = source

        self.present(imagePicker,animated: true,completion: nil)

    }

}

```

```

func imagePickerController(_ picker: UIImagePickerController,
didFinishPickingMediaWithInfo info: [UIImagePickerController.InfoKey : Any])
{

    mainImageView.image =
info[UIImagePickerController.InfoKey.editedImage] as? UIImage

    mainImageView.contentMode = .scaleAspectFit

    mainImageView.clipsToBounds = true

    dismiss(animated: true, completion: nil)


    DispatchQueue.main.asyncAfter(deadline: .now() + .seconds(1)) {

        UIButton.animate(withDuration: 1) {

            self.saveButton.alpha = 1

            self.saveButton.isHidden = false

        }
    }
}

```

```
    }  
  }  
}
```

```
class Sharing {  
  
    static func share(on viewController: UIViewController,image: UIImage?) {  
  
        var toShare = [Any]()  
  
        if let image = image { toShare.append(image) }  
  
        guard toShare.count > 0 else { return }  
  
        let activityViewController = UIActivityViewController(activityItems:  
toShare, applicationActivities: nil)  
  
        activityViewController.popoverPresentationController?.sourceView =  
viewController.view  
  
        viewController.present(activityViewController, animated: true,  
completion: nil)  
  
    }  
  
}
```