**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики**

**Кафедра информационных систем управления**

**Отчет**

**По технологиям программирования**

Выполнил студент группы №13

*Веренич Владислав Николаевич*

**Минск 2022**

## 1.1 Рисование (приложение на языке objective-C)

**ЗАДАЧИ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

Разобрать пример рисования на языке Objective-C

**Входные данные**

Цвет, размер и форма кисти, использующейся для рисования.

**Выходные данные**

Рисунок, состоящий из линий произвольной формы.

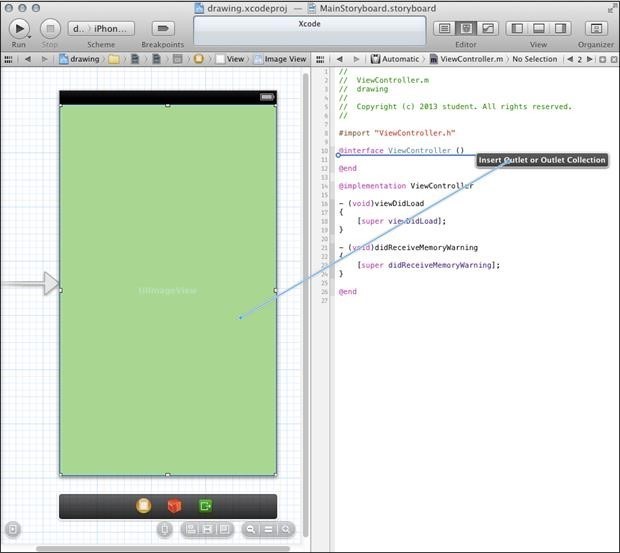
## Указания

Создайте новый Single View Application проект. Добавьте в интерфейс приложения компонент UIImageView (Рис. 1)



**Рис. 1.** Компонент холста

Создайте IBOutlet области рисования в контроллере. Удерживая нажатой клавишу Ctrl перетащите UIImageView в область interface файла ViewController.m используя режим Assistant editor (Рис. 2).



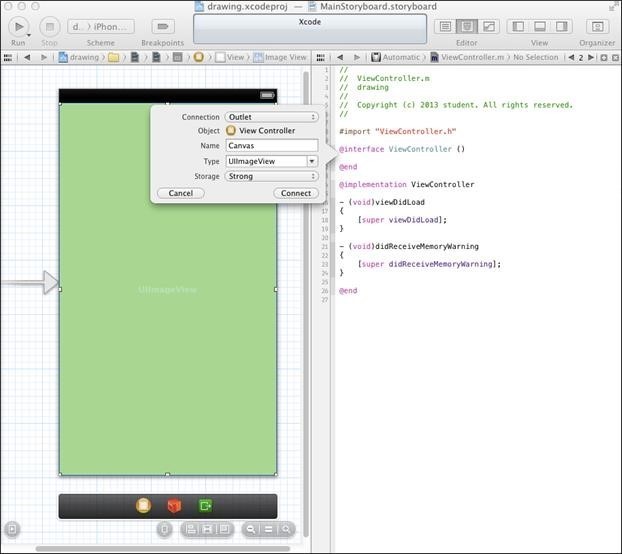
**Рис. 2.** Добавление связи

Введите canvas и нажмите Connect (Рис. 3).

В область interface добавиться объявление холста (Листинг 1).

@property (strong, nonatomic) IBOutlet UIImageView \*canvas;

### **Листинг 1.** Объявление холста



**Рис. 3.** Создание IBOutlet

Рисование будет осуществляться при помощи обработки жестов. Для этого в части implementation объявить методы отвечающие за обработку жестов (Листинг 2).

* (void)touchesBegan:(NSSet \*)touches withEvent:(UIEvent \*)event;
* (void)touchesMoved:(NSSet \*)touches withEvent:(UIEvent \*)event;

### **Листинг 2.** Объявление методов обработки жестов

Для рисования линии необходимо хранить предыдущую точку маршрута. Добавим свойство lastPont типа CGPoint в интерфейc контролера. CGPoint это обычная Си-шная структура хранящая координаты x и y (Листинг 3).

@property CGPoint lastPoint;

### **Листинг 3.** Объявление свойства последней точки касания

Метод touchesBegan:withEvent: будет отвечать за сохранение начальной точки рисуемой линии в свойстве lastPoint (Листинг 4).

UITouch \*touch = [touches anyObject];

[self setLastPoint:[touch locationInView:self.view]];

### **Листинг 4.** Сохранение позиции касания

Метод touchesMoved:touchesMoved: будет отвечать за рисование линии на экране. Для рисования необходимо установить контекст рисования, установить параметры линии такие как цвет и размер, нарисовать линию и удалить контекст (Листинг 5).

UITouch \*touch = [touches anyObject];

CGPoint currentPoint = [touch locationInView:self.view];

UIGraphicsBeginImageContext(self.view.frame.size);

CGRect drawRect = CGRectMake(0.0f, 0.0f, self.view.frame.size.width, self.view.frame.size.height);

[[[self canvas] image] drawInRect:drawRect];

CGContextSetLineCap(UIGraphicsGetCurrentContext(), kCGLineCapRound);

CGContextSetLineWidth(UIGraphicsGetCurrentContext(), 5.0f);

CGContextSetRGBStrokeColor(UIGraphicsGetCurrentContext(), 1.0f, 0.0f, 0.0f, 1.0f);

CGContextBeginPath(UIGraphicsGetCurrentContext());

CGContextMoveToPoint(UIGraphicsGetCurrentContext(), \_lastPoint.x,

\_lastPoint.y);

CGContextAddLineToPoint(UIGraphicsGetCurrentContext(), currentPoint.x, currentPoint.y);

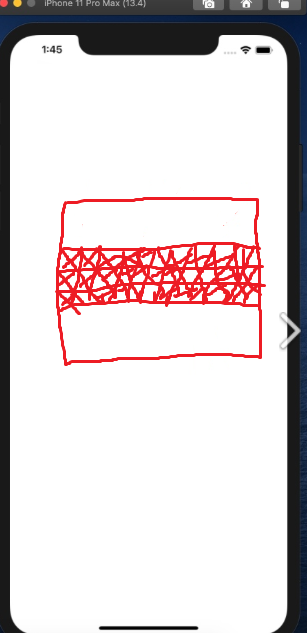
CGContextStrokePath(UIGraphicsGetCurrentContext());

[[self canvas] setImage:UIGraphicsGetImageFromCurrentImageContext()];

UIGraphicsEndImageContext();

\_lastPoint = currentPoint;

**Листинг 5.** Рисование линии на экране



#### 1.2. Создание iOS приложения с простыми графическими объектами на языке Swift

**Цель**: изучить примеры рисования графических объектов и получить навыки разработки приложений

1. Просмотреть видео: <https://www.youtube.com/watch?v=wmfecAPy_SM>
2. Изучить статьи:

◦ [http://sketchytech.blogspot.com.by/2016/03/swift-giving-context-to-cgcontext- part-i.html](http://sketchytech.blogspot.com.by/2016/03/swift-giving-context-to-cgcontext-part-i.html)

◦ [https://www.raywenderlich.com/128614/core-graphics-os-x-tutoria l](https://www.raywenderlich.com/128614/core-graphics-os-x-tutorial)

◦ [https://www.raywenderlich.com/162315/core-graphics-tutorial-part-1-getting- started](https://www.raywenderlich.com/162315/core-graphics-tutorial-part-1-getting-started)

◦ [https://www.raywenderlich.com/162313/core-graphics-tutorial-part-2-gradients- contexts](https://www.raywenderlich.com/162313/core-graphics-tutorial-part-2-gradients-contexts)

◦ [https://www.raywenderlich.com/167352/core-graphics-tutorial-part-3-patterns- playgrounds](https://www.raywenderlich.com/167352/core-graphics-tutorial-part-3-patterns-playgrounds)

◦<http://proswift.ru/calayer-ili-kak-zakruglit-ugly-sdelat-ten-i-gradient-na-swift/>

◦ [https://developer.apple.com/library/content/documentation/GraphicsImaging/ Conceptual/drawingwithquartz2d/Introduction/Introduction.html#//apple\_ref/doc/ uid/TP40007533-SW1](https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/drawingwithquartz2d/Introduction/Introduction.html#//apple_ref/doc/uid/TP40007533-SW1)

#### 1.3. Создание iOS приложения на языке Swift с использованием анимации

**Цель**: изучить примеры рисования графических объектов и получить навыки разработки приложений.

1. Изучить статьи:

◦<https://www.raywenderlich.com/173544/ios-animation-tutorial-getting-started-3>

◦ [https://www.raywenderlich.com/173576/ios-animation-tutorial-custom-view- controller-presentation-transitions-3](https://www.raywenderlich.com/173576/ios-animation-tutorial-custom-view-controller-presentation-transitions-3)

◦ [https://developer.apple.com/library/content/documentation/WindowsViews/ Conceptual/ViewPG\_iPhoneOS/AnimatingViews/AnimatingViews.html](https://developer.apple.com/library/content/documentation/WindowsViews/Conceptual/ViewPG_iPhoneOS/AnimatingViews/AnimatingViews.html)

◦ [https://developer.apple.com/library/content/documentation/Cocoa/Conceptual/ CoreAnimation\_guide/CoreAnimationBasics/CoreAnimationBasics.html#// apple\_ref/doc/uid/TP40004514-CH2-SW3](https://developer.apple.com/library/content/documentation/Cocoa/Conceptual/CoreAnimation_guide/CoreAnimationBasics/CoreAnimationBasics.html#//apple_ref/doc/uid/TP40004514-CH2-SW3)

#### 1.4. Создание iOS приложения на языке Swift с использованием жестов

**Цель работы:** Научиться обрабатывать события, которые будут происходить при использовании различных жестов.

Создать объект UILabel, который будет отображать используемый пользователем жест и менять цвет своего фона, в зависимости от используемого жеста. Реализовать работу следующих жестов:

1. Вращение (цвет фона: синий)
2. Масштабирование (цвет фона: красный)
3. Касание (цвет фона: зеленый)
4. Долгое нажатие (цвет фона: оранжевый)
5. Смахивание (цвет фона: серый) **Ход работы:**

##### Создание проекта Xcode

Создаем проект Xcode, аналогично предыдущим лабораторным работам. Убираем галочку Use Core Data и выбираем iOS – Application – Single View

Application.

##### Добавление элементов интерфейса

Открываем файл Main.storyboard и на экран контроллера View добавляем объект Label. Располагаем его посередине экрана и растягиваем (рис.11).

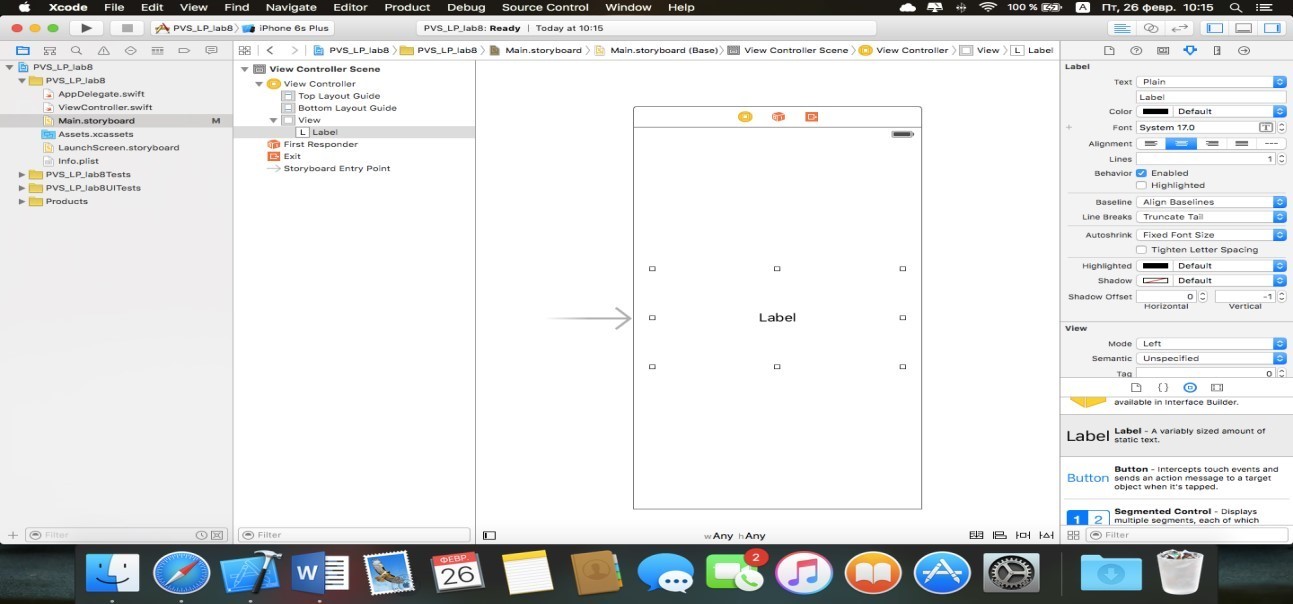


Рис. 11. Добавление объекта Label

Далее, находим объекты: Tap Gesture Recognizer, Long Press Gesture Recognizer, Pinch Gesture Recognizer, Rotation Gesture Recognizer и Swipe Gesture Recognizer. Перетаскиваем их в область объекта Label (рис. 12).

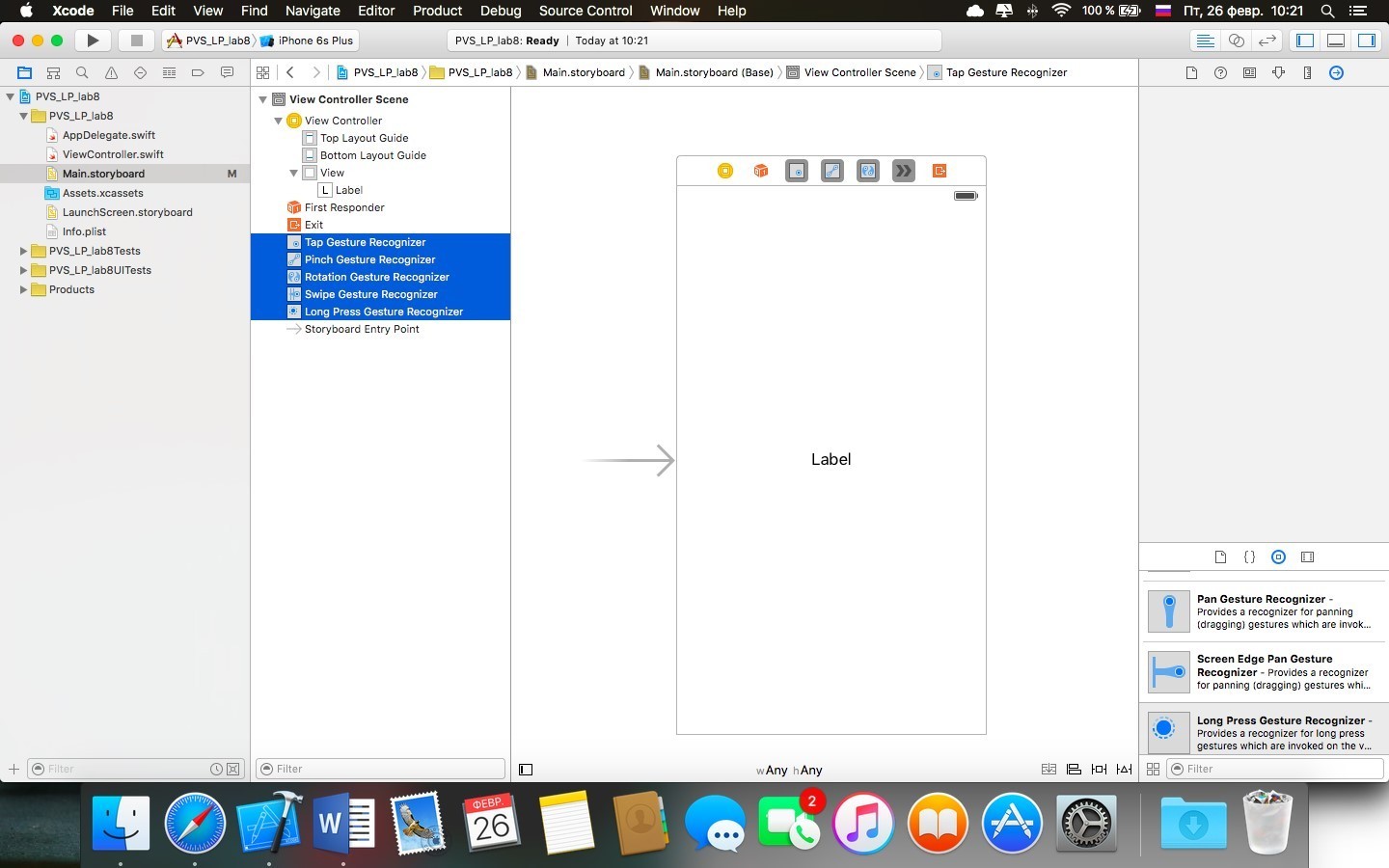


Рис.12. Область объекта Label

##### Создание связей объектов интерфейса с кодом приложения

В описание класса контроллера View для объекта Label добавляем связь типа Outlet:

@IBOutlet weak var gestureIndicator: UILabel!

Для объектов Gesture Recognizer добавляем пять связей типа Action: @IBAction func tap(sender: AnyObject)

{

}

@IBAction func pinch(sender: AnyObject)

{

}

@IBAction func rotation(sender: AnyObject)

{

}

@IBAction func swipe(sender: AnyObject)

{

}

@IBAction func longPress(sender: AnyObject)

{

}

##### Описание событий в коде приложения

Для того, чтобы использование жестов в области объекта Label стало возможным, присвоим параметру userInteractionEnabled (взаимодействие с пользователем включено) значение true в методе ViewDidLoad(): gestureIndicator.userInteractionEnabled = true

В этом же методе, задаем выравнивание текста, количество строк текста, текст и цвет фона:

gestureIndicator.textAlignment = NSTextAlignment.Center

gestureIndicator.numberOfLines = 2 gestureIndicator.text = "Используйте жесты в этой области" gestureIndicator.backgroundColor = UIColor.yellowColor()

Далее, опишем события для всех пяти жестов в созданных ранее функциях. Для жеста касания:

@IBAction func tap(sender: AnyObject)

{

gestureIndicator.text = "Жест: касание\n Цвет фона: зеленый" gestureIndicator.backgroundColor = UIColor.greenColor() } Для жеста масштабирования:

@IBAction func pinch(sender: AnyObject)

{

gestureIndicator.text = "Жест: масштабирование\n Цвет фона: красный" gestureIndicator.backgroundColor = UIColor.redColor()

}

Для жеста вращения:

@IBAction func rotation(sender: AnyObject)

{

gestureIndicator.text = "Жест: вращение\n Цвет фона: синий"

gestureIndicator.backgroundColor = UIColor.blueColor()

}

Для жеста смахивания:

@IBAction func swipe(sender: AnyObject)

{

gestureIndicator.text = "Жест: смахивание\n Цвет фона: серый"

gestureIndicator.backgroundColor = UIColor.lightGrayColor()

}

Для жеста долгого нажатия:

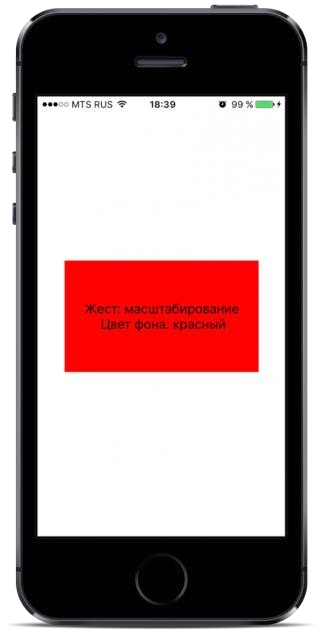
@IBAction func longPress(sender: AnyObject)

{

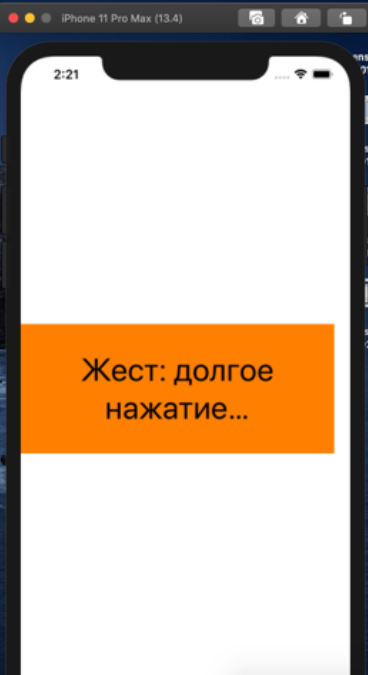
gestureIndicator.text = "Жест: долгое нажатие\n Цвет фона: оранжевый" gestureIndicator.backgroundColor = UIColor.orangeColor()

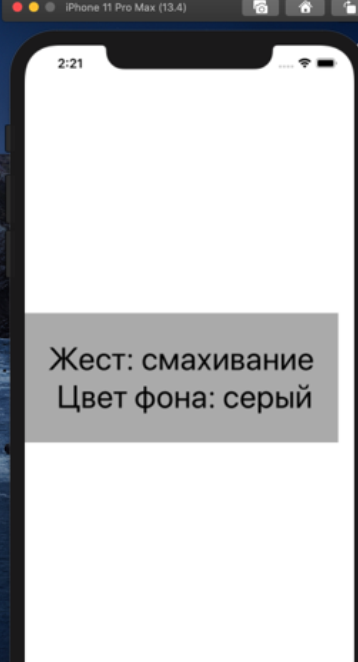
}

##### Сборка проекта и запуск приложения



Результат:





# 2. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

## ВАРИАНТЫ ЗАДАНИЙ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

### **2.1 Рисование (приложение на языке objective-C)**

Дополнить функционал приложения, созданного в 1.1. Требуется разработать программу рисования. Интерфейс программы должен поддерживать рисование линиями 5 размеров и разными цветами.

Реализовать возможность сохранения нарисованного изображения в файл.

Исходный код приложений на языке Objective-C должен соответствовать модели КИС для Objective-C.

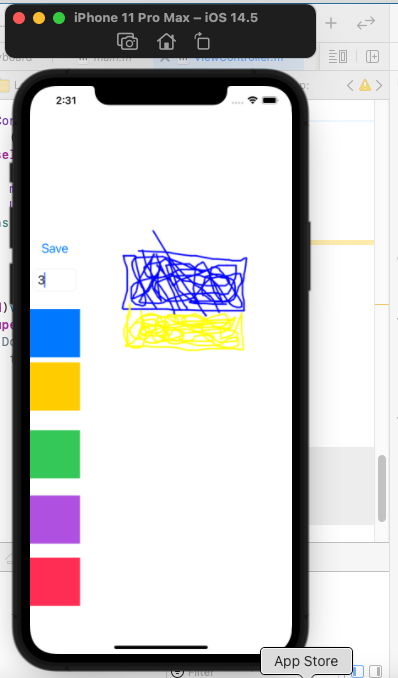
**Входные данные**

Цвет, размер и форма кисти, использующейся для рисования.

## Выходные данные

Рисунок, состоящий из линий произвольной формы, сохранённый в библиотеку.

Результат:



**Вариант 16. Декагон и равнобедренная трапеция**

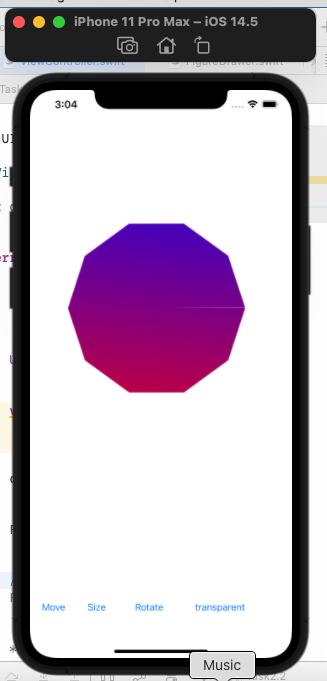
### **2.2. Создание iOS приложения с простыми графическими объектами на языке Swift**

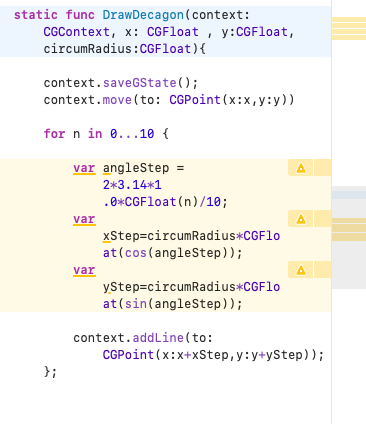
Создать iOS приложение, иллюстрирующее объекты двумерной графики с однотонным фоном, градиентом, согласно вариантам. Проиллюстрировать различные варианты создания новых фигур за счет вычитания или объединения 2 разных фигур.

Код проекта распределить по группам (каталогам) согласно концепции MVC

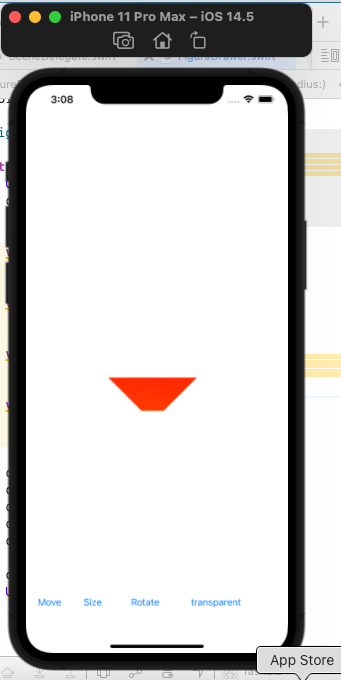
(Model, View, Controller).

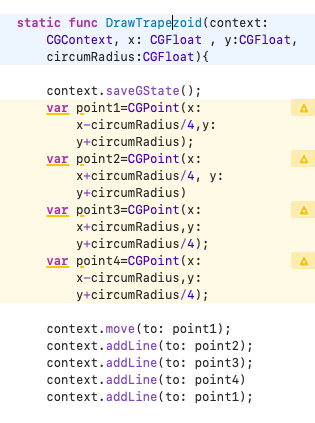
**Decagon:**



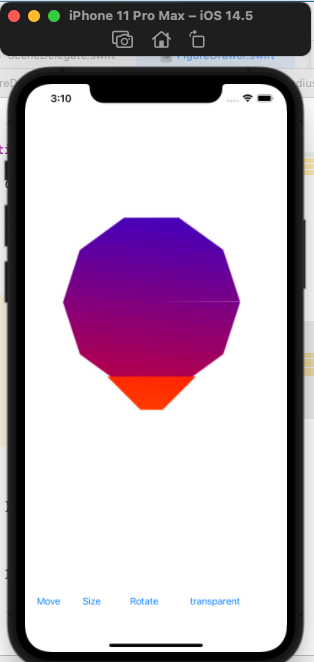


**Trapezoid:**





**Combination:**



### **2.3. Создание iOS приложения с использованием анимации**

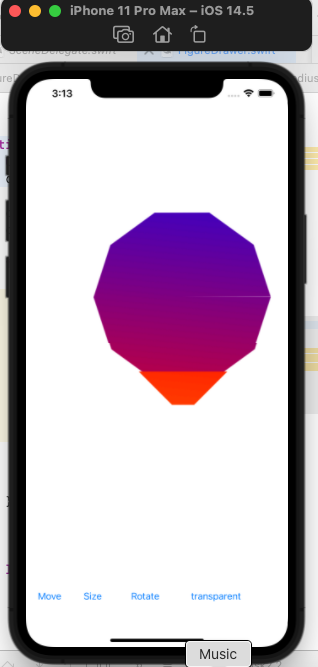
Дополнить iOS приложение, созданное в задании 2.2, анимационными эффектами. Проиллюстрировать следующие эффекты:

* Перемещение
* Вращение
* Масштабирование
* Прозрачность
* Наложение как минимум двух эффектов, например перемещение и вращение

Код проекта распределить по группам (каталогам) согласно концепции MVC

(Model, View, Controller).

**Move:**



**Rotation:**



**Size:**

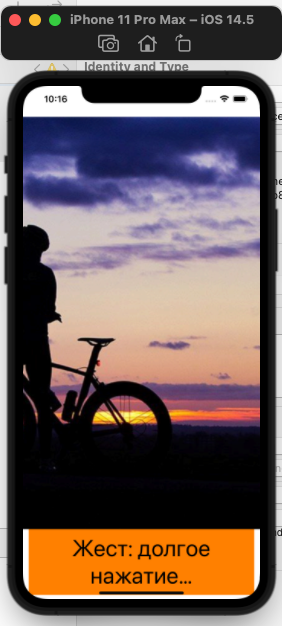


### **2.4. Создание iOS приложения на языке Swift с использованием жестов**

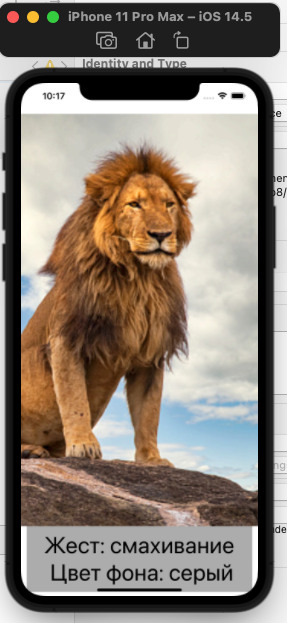
Изменять фон изображения, в зависимости от жеста. Проиллюстрировать следующие жесты:

1. Вращение (изображение фона 1)
2. Масштабирование (изображение фона 2)
3. Касание (изображение фона 3)
4. Долгое нажатие (изображение фона 4)
5. Смахивание (изображение фона 5)

Для выполнения задания подобрать 5 изображений фона. Код проекта распределить по группам (каталогам) согласно концепции MVC (Model, View, Controller).







## КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Как нарисовать контур?

***ОТВЕТ:***

CGContextAddLineToPoint()

StrokePath();

1. Как нарисовать прямоугольник?

***ОТВЕТ:***

CGContext.addRect();

CgContext.drawPath();

1. Как заполнить контур цветом? Градиентом?

***ОТВЕТ:***

CGContext.FillPath();

DrawLinearGradient();

1. Как заполнить прямоугольник цветом, сохранив цвет контура?

***ОТВЕТ:***

Fill() -> Stroke();

1. Как нарисовать круг / эллипс?

***ОТВЕТ:***

var ovalPath = UIBezierPath(ovalInRect: CGRectMake(x, y, width, height))

UIColor.grayColor().setFill()

ovalPath.fill()

1. Как добавить тень к рисункам?

***ОТВЕТ:***

CGContextSetShadowWithColor(context, shadowOffset, shadowBlurRadius, (shadow as UIColor).CGColor)

1. Как нарисовать изображение на контроллере View в прямоугольнике?

***ОТВЕТ:***

Draw(in: rect)

1. Как перерисовать представление View с помощью метода SetNeedsDisplay

***ОТВЕТ:***

Данный метод делает запрос на перерисовку при следующем цикле рисования.

1. Как вычесть одну фигуру из другой?

***ОТВЕТ:***

Работа с ImageView.mask

1. Какое компонент Cocoa отвечает за обработку жестов?

***ОТВЕТ:***

Gesture recognizer

11) Как добавить и обработать жест касания (нажатия) (tap gesture)?

***ОТВЕТ:***

Добавить tap gesture recognizer в view. Для обработки действия добавить макрос action

12) Как добавить и обработать долгое нажатие (long press gesture)?

***ОТВЕТ:***

Добавить long press gesture recognizer в view. Для обработки действия добавить action в код

13) Как добавить и обработать жест перелистывания (смахивания) (swipe gesture)?

***ОТВЕТ:***

Добавить swipe gesture recognizer в storyboard. Для обработки действия добавить action в код

14) Как обрабатывается жест стягивания (щипка) (pinch gesture)?

***ОТВЕТ:***

Аналогично

15) Как обрабатывается жест растягивания (spread gesture)?

***ОТВЕТ:***

Аналогично

16) Как добавить жесты непосредственно на Storyboard?

***ОТВЕТ:***

Открыть object library (command+shift+L, там найти соответствующий recognizer и перетащить его на storyboard.

17) Какие примитивы рисования Вы знаете?

***ОТВЕТ:***

Графический примитив — это простейший геометрический объект, отображаемый на экране дисплея: точка, отрезок прямой, прямоугольник, дуга, окружность и т. д.

18) Что такое CGContext?

***ОТВЕТ:***

Он содержит параметры рисования и всю нужную для рисования на устройстве информацию.

19) Что такое UIBezierPath? Чем отличается от CGContext?

***ОТВЕТ:***

Вы изначально используете UIBezierPath, чтобы указать только геометрию вашего пути. Контуры могут определять простые формы, такие как прямоугольники, овалы и дуги, или они могут определять сложные многоугольники, которые включают в себя смесь прямых и изогнутых отрезков. После определения формы вы можете использовать дополнительные методы этого класса, чтобы отобразить путь в текущем контексте рисования.  
Тип CGContext представляет пункт назначения 2D-рисования Quartz. Графический контекст содержит параметры рисования и всю специфическую для устройства информацию, необходимую для визуализации краски на странице до места назначения, независимо от того, является ли место назначения окном в приложении, растровым изображением, документом PDF или принтером.

может быть немного быстрее, поскольку он не должен проходить через Obj-C и имеет более продвинутые функции, такие как CGPathApply

20) Что такое CGImage?

***ОТВЕТ:***

Растровое изображение

21) Что такое CGPath?

***ОТВЕТ:***

Описание фигур или линий, которые будут нарисованы в контексте.

22) В чем заключаются трудности использования двумерной графики?

***ОТВЕТ:***

Для создания различных кривых необходима высокая вычислительная мощность.

23) Что такое OpenGL?

***ОТВЕТ:***  
Open Graphics Library (OpenGL) - это кросс-языковой, кросс-платформенный интерфейс прикладного программирования (API) для рендеринга 2D и 3D векторной графики. API обычно используется для взаимодействия с графическим процессором (GPU), чтобы добиться аппаратно-ускоренного рендеринга.

24) Что такое шейдер?

***ОТВЕТ:***

Шейдер – это программа выполняющаяся на видеокарте. Ее задача – обработать входной набор пикселей и вернуть новый набор пикселей.

25) Чем вершинный шейдер отличается от пиксельного?

***ОТВЕТ:***

Вершинные шейдеры

Вершинные шейдеры - это программы, выполняемые видеочипами, которые производят математические операции с вершинами (vertex, из них состоят 3D объекты в играх), иначе говоря, они предоставляют возможность выполнять программируемые алгоритмы по изменению параметров вершин

Пиксельные шейдеры - это программы, выполняемые видеочипом во время растеризации для каждого пикселя изображения, они производят выборку из текстур и/или математические операции над цветом и значением глубины (Z-buffer) пикселей. Все инструкции пиксельного шейдера выполняются попиксельно, после того, как операции с трансформированием и освещением геометрии завершены.

26) Что такое Metal?

***ОТВЕТ:***

Это framework, который обеспечивает почти прямой доступ к графическому процессору (GPU) устройства. Используется в основном для 3d графики.