Praktikum iOS-Entwicklung

Aufgabe 6a

Schattierungen mit Core Graphics (in Objective-C)

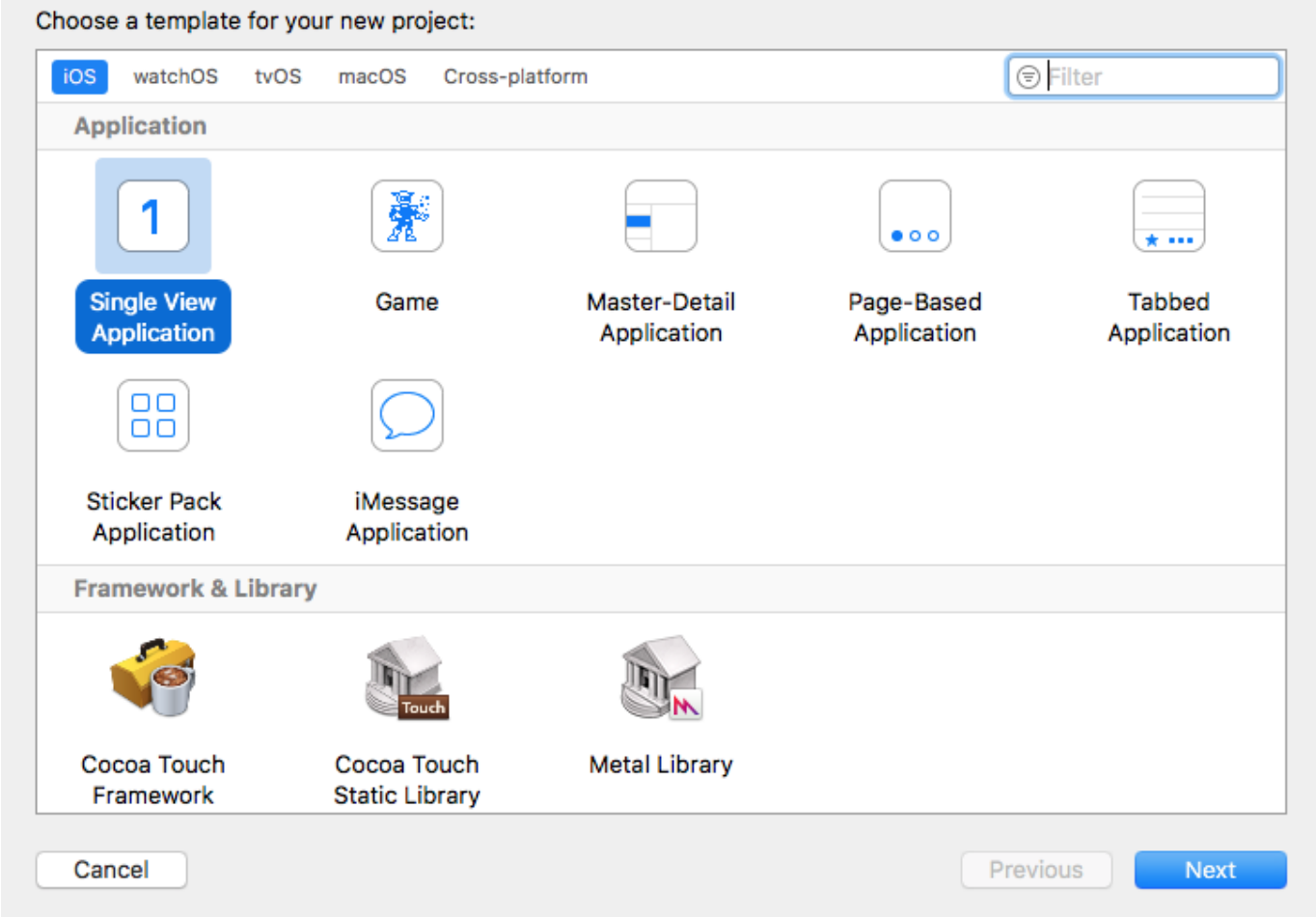
Isabel Schönewald

Aufgabe

- Es sollen zufallsbasiert zwischen 1 und 50 Rechtecke generiert werden
- Jedes Rechteck hat eine zufallsbasierte Position, eine zufallsbasierte Größe und eine zufallsbasierte Füllfarbe
- Jedes zweite Rechteck soll einen Schatten generieren

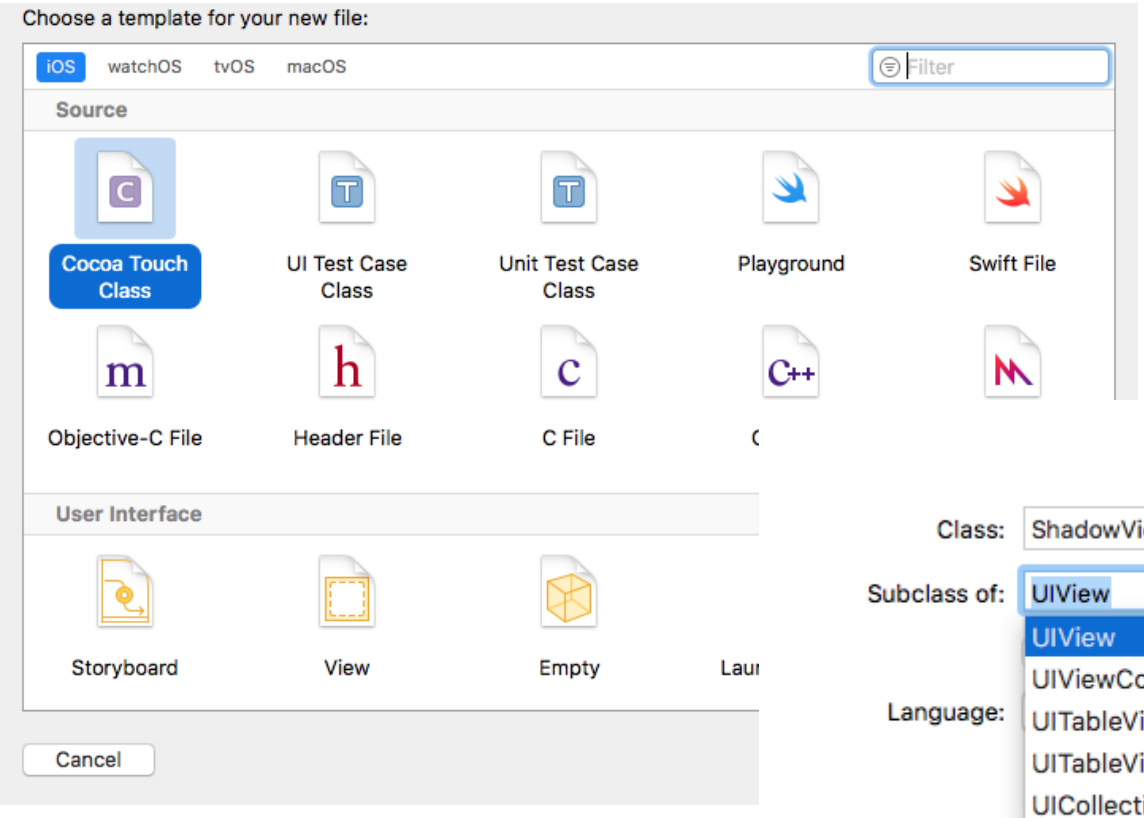
Schritt 1: Projekt vom Typ „Single View Application“ erstellen

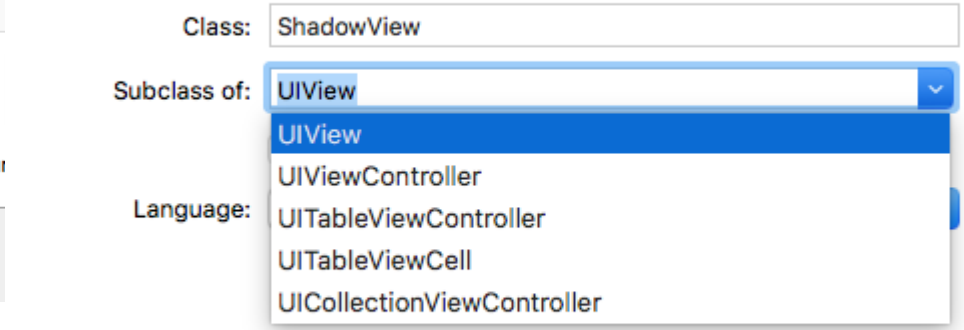
File → New → Project



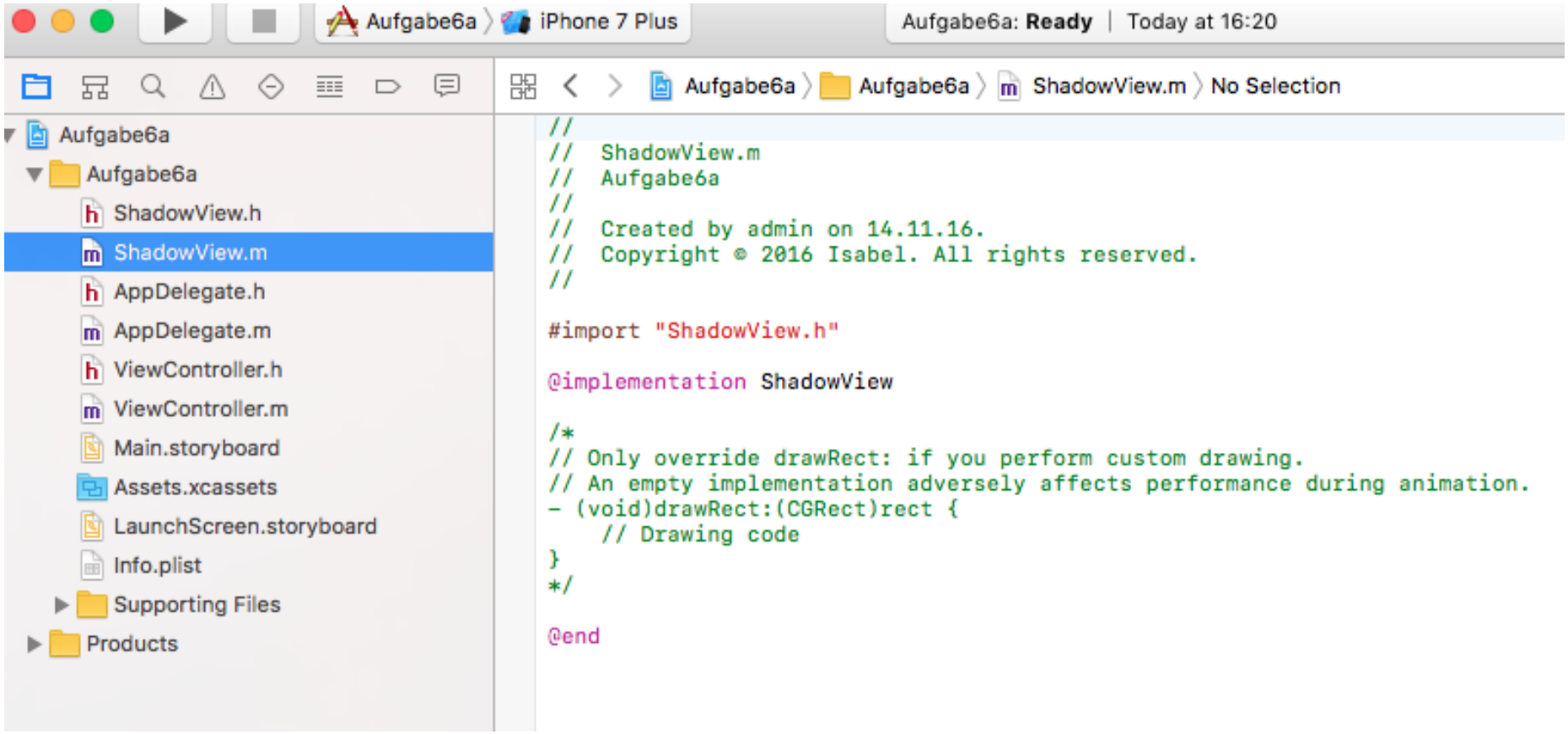
Schritt 2: Klasse „ShadowView“ erstellen, die von der Klasse „UIView“ erbt

File → New → File





ShadowView.m



Schritt 3: DrawRect-Methode überschreiben

```
- (void)drawRect:(CGRect)rect {

    // Referenz auf den Grafikkontext der View erzeugen
    CGContextRef ctx = UIGraphicsGetCurrentContext();

    // Zeichenfläche speichern
    CGRect bounds = self.bounds;

    // Zeichenfläche weiß färben
    CGContextSetRGBFillColor(ctx, 1.0, 1.0, 1.0, 1.0);
    CGContextFillRect(ctx, bounds);

    // Schattenwert für später setzen
    Boolean shadow = true;

    // Anzahl der Rechtecke durch Zufallsfunktion bestimmen
    int number_rectangles = [self randomNumber: 50] + 1;

    // Für jedes Rechteck
    for (int i=0; i< number_rectangles; i++) {

        // Zufällige Breite, Höhe, x-Koordinate und y-Koordinate bestimmen
        // Bedingung: Rechtecke müssen komplett zu sehen sein
        unsigned random_width = [self randomNumber: bounds.size.width] + 1;
        unsigned random_height = [self randomNumber: bounds.size.height] + 1;
        unsigned random_x = [self randomNumber: bounds.size.width - random_width];
        unsigned random_y = [self randomNumber: bounds.size.height - random_height];
        // Rechteck mit den zufälligen Werten erstellen
        CGRect rectangle = CGRectMake (random_x, random_y, random_width, random_height);
```

```
        // Für jede Farbe zufälligen Wert bestimmen
        float random_red = [self randomColor];
        float random_green = [self randomColor];
        float random_blue = [self randomColor];

        // Wenn Schatten gesetzt werden soll
        if (shadow == true) {

            //Grafikkontext zwischenspeichern
            CGContextSaveGState(ctx);
            // Wert für Abstand zwischen Schatten und Rechteck festlegen
            CGSize shadow_offset = CGSizeMake(-15,20);
            // Schatten setzen mit Kontext, Abstand und Wert für die Unschärfe
            CGContextSetShadow(ctx, shadow_offset, 5);

            // Zufällig bestimmte Werten für Farben einsetzen & Rechteck zeichnen
            CGContextSetRGBFillColor(ctx, random_red, random_green, random_blue, 1.0);
            CGContextFillRect(ctx, rectangle);

            // Grafikkontext wiederherstellen
            CGContextRestoreGState(ctx);

        } else {

            // Sonst Rechteck ohne Schatten zeichnen
            CGContextSetRGBFillColor(ctx, random_red, random_green, random_blue, 1.0);
            CGContextFillRect(ctx, rectangle);

        }

        // Schattenwert wechseln
        shadow = !shadow;
    }

}
```

Hilfsfunktionen in ShadowView.m

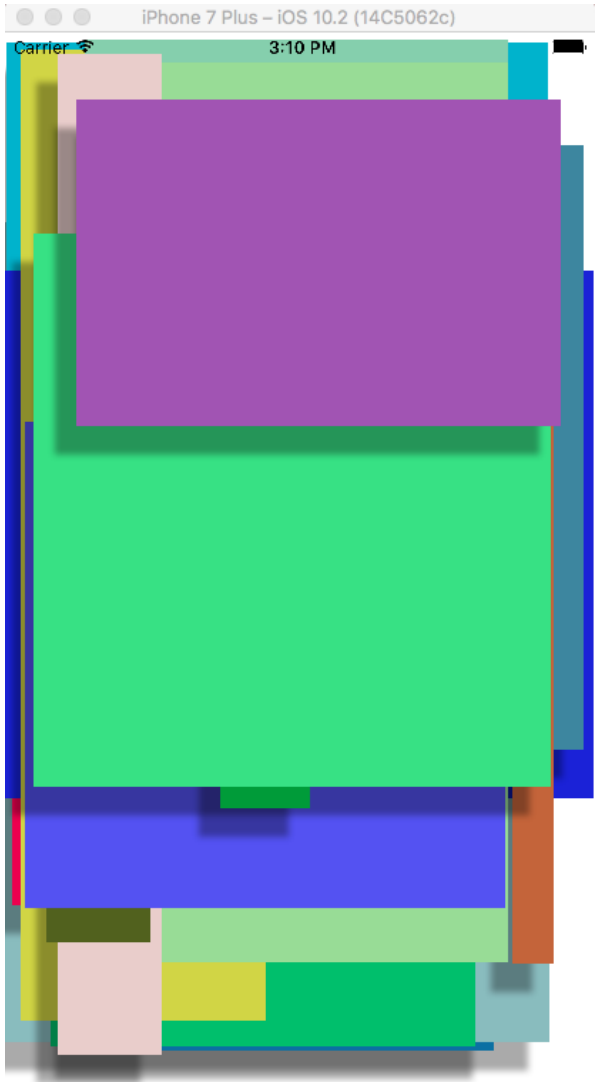
Erstellen einer Zufallszahl zwischen 0 und upper_bound:

```
- (unsigned) randomNumber: (float) upper_bound {
    return arc4random_uniform(upper_bound);
}
```

Erstellen eines zufälligen Farbwerts:

```
- (float) randomColor {
    return arc4random() % 256 / 256.0;
}
```

ViewController.m

```
#import "ViewController.h"
// Einbinden von ShadowView
#import "ShadowView.h"

@interface ViewController ()

@end

@implementation ViewController

- (void)viewDidLoad {

    // Initialisierung der View
    CGRect frame = [[UIScreen mainScreen] bounds];
    ShadowView *view = [[ShadowView alloc] initWithFrame:frame];
    self.view = view;

}


- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}


@end
```

Mögliche Ausgaben auf dem iPhone