**TEMAT:** Sprzętowe zasoby systemu mobilnego.

## TREŚĆ ZADANIA

**Cel:** wykorzystanie wybranych sprzętowych zasobów aplikacji iOS.

*Obsługa zdarzenia potrząśnięcia telefonem*

1. W klasie *ViewController* zaimplementuj następujące metody:

```
- (void)viewDidLoad {
    [super viewDidLoad];
    [self becomeFirstResponder];
}

-(void)motionEnded:(UIEventSubtype)motion withEvent:(UIEvent *)event {
    if (motion == UIEventSubtypeMotionShake) {
        [self showShakeDetectedAlert];
    }
}

-(BOOL)canBecomeFirstResponder {
    return YES;
}
```

2. Zaimplementuj metodę wyświetlającą powiadomienie z pytaniem, czy użytkownik chce zmienić kolor tła widoku. W przypadku wyrażenia zgody (przycisk „Yes") tło powinno zostać zmienione na dowolny kolor. Brak wyrażenia zgody powinien skutkować jedynie zapisaniem do logów informacji o wykrytym zdarzeniu potrząśnięcia. Uzupełnij poniższy kod:

```
-(IBAction)showShakeDetectedAlert {
    UIAlertController *alertController = [UIAlertController alertControllerWithTitle:@"Shake gesture detected"
        message:@"Do you want to change the background color?" preferredStyle:UIAlertControllerStyleAlert];

    UIAlertAction *yesButton = [UIAlertAction actionWithTitle:@"Yes" style:UIAlertActionStyleDefault
        handler:^(UIAlertAction *action) {
    }];

    UIAlertAction *noButton = [UIAlertAction actionWithTitle:@"No" style:UIAlertActionStyleDefault
        handler:^(UIAlertAction *action) {
    }];

    [alertController addAction:yesButton];
    [alertController addAction:noButton];
    [self presentViewController:alertController animated:YES completion:nil];
}
```

*Obsługa gestów*

1. Dodaj w głównym widoku aplikacji etykietę *Label*, która będzie wyświetlała nazwę wykonanego gestu (zapewnij możliwość wyświetlania dłuższego tekstu w dwóch liniach).
2. Umieść na kontrolerze widoku obiekty odpowiednich gestów:
    a. Tap Gesture Recognition
    b. Pinch Gesture Recognition
    c. Swipe Gesture Recognition
    d. Long Press Gesture Recognition
3. Po zaznaczeniu danego gestu można edytować jego ustawienia. Zaznacz więc *Tap Gesture Recognition* i w ustawieniach wpisz liczbę 3 w polu *Taps*.
4. Dla gestu *Long Press* ustaw atrybut czasu przytrzymania na 2 sekundy.
5. Utwórz połączenia typu IBAction dla wszystkich gestów. Po dodaniu wszystkich połączeń plik nagłówkowy kontrolera powinien zawierać następujące wpisy:

```
@property(weak, nonatomic) IBOutlet UILabel *gestureLabel;

- (IBAction) tapGesture: (UITapGestureRecognizer *) sender;
- (IBAction) pinchGesture: (UIPinchGestureRecognizer *) sender;
- (IBAction) swipeGesture: (UISwipeGestureRecognizer *) sender;
- (IBAction) longPressGesture: (UILongPressGestureRecognizer *) sender;
```

6. Następnie zaimplementuj odpowiednie metody obsługujące cztery dodane gesty. Kod każdej metody powinien wyglądać analogicznie do następującego:

```
- (IBAction) tapGesture:(UITapGestureRecognizer *)sender {
    _gestureLabel.text = @"Tap detected";
}
```

*Obsługa GPS*

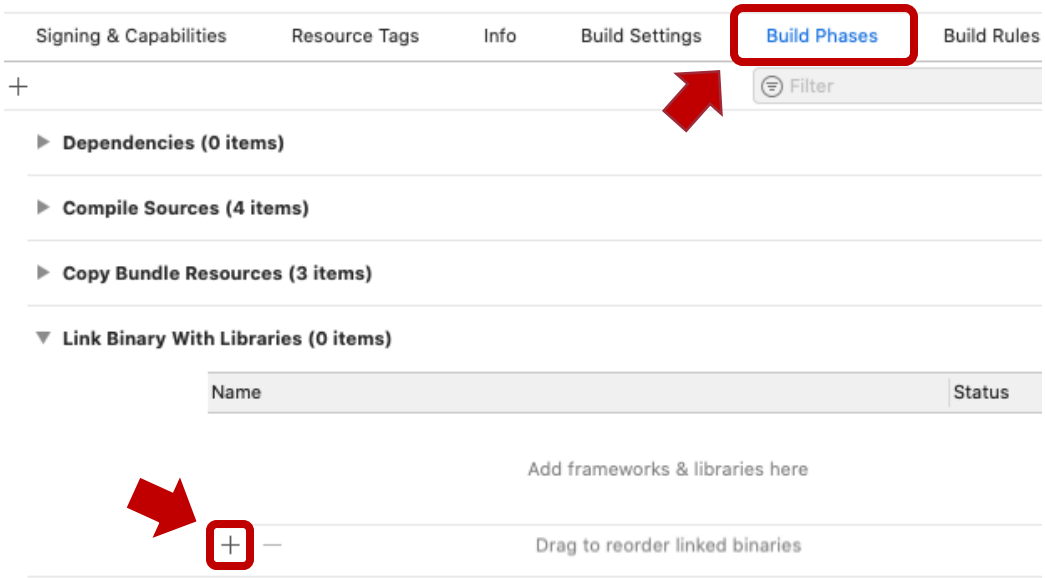
1. Przygotuj kolejny widok, który będzie zawierał trzy pola (etykiety):
    - Latitude:
    - Longtitude:
    - Address:
2. Naprzeciw każdego z tych pól powinno znajdować się puste pole przygotowane do wypełnienia go odpowiednią informacją. Pole, które zostanie wypełnione adresem powinno zostać skonfigurowane w taki sposób, aby wyświetlało informacje w 3 liniach (opcja *lines*).
3. Dodaj przycisk z napisem „My current location”.
4. Utwórz zmienne typu outlet odpowiadające trzem etykietom, które będą przeznaczone do wypełnienia informacjami o aktualnej lokalizacji.
5. Utwórz połączenie typu *IBAction* dla przycisku.

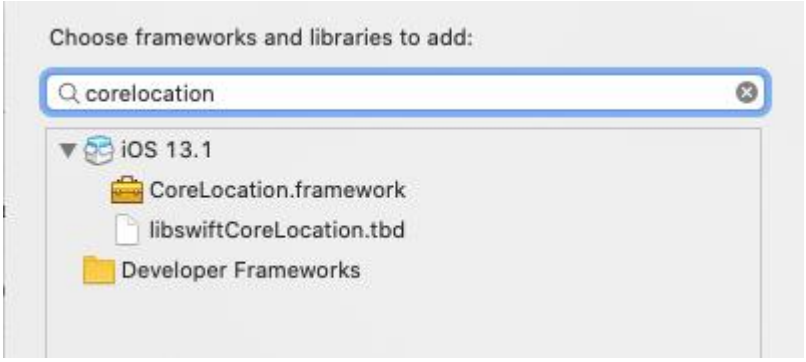
6. Kod pliku interfejsu kontrolera powinien wyglądać podobnie do następującego:

```
14  @interface ViewController : UIViewController <CLLocationManagerDelegate>

    @property(weak, nonatomic) IBOutlet UILabel *latitudeLabel;
    @property(weak, nonatomic) IBOutlet UILabel *longtitudeLabel;
    @property(weak, nonatomic) IBOutlet UILabel *addressLabel;
24
    @property(weak, nonatomic) IBOutlet UITextField *latitudeText;
    @property(weak, nonatomic) IBOutlet UITextField *longtitudeText;
    @property(weak, nonatomic) IBOutlet UITextView *addressText;
    @property(weak, nonatomic) IBOutlet UIButton *currentLocationButton;
29
    -(IBAction)getCurrentLocation:(id)sender;
31
32  @end
```

7. Dodaj do aplikacji framework Core Location. W nawigatorze projektu wybierz aktualny projekt pod etykietą *Targets*. Następnie w oknie, które się pojawi należy wybrać *Build Phases* > *Link Binary with Libraries* i kliknąć przycisk plusa.




8. Po tym kroku można przystąpić do właściwiej implementacji pobrania lokalizacji. Skorzystaj ze wzorca delegata – zastosuj w kontrolerze odpowiedni protokół, jak w przykładzie poniżej.

   Plik *.h kontrolera:

```
#import <UIKit/UIKit.h>
#import <CoreLocation/CoreLocation.h>

NS_ASSUME_NONNULL_BEGIN

@interface ViewController : UIViewController <CLLocationManagerDelegate>
```

9. *CLLocationManager* odpowiada za dostarczenie informacji na temat lokalizacji. *CLGeocoder* odpowiada za pobranie adresu.
10. Dodaj następujące zmienne:

```
@interface ViewController : UIViewController <CLLocationManagerDelegate>
{
    CLLocationManager *locationManager;
    CLGeocoder *geocoder;
    CLPlacemark *placemark;
}
```

11. Zainicjalizuj obiekt *CLLocationManager* i *CLGeocoder* w metodzie *viewDidLoad*:

```
- (void)viewDidLoad {
    [super viewDidLoad];
    locationManager = [[CLLocationManager alloc] init];
    geocoder = [[CLGeocoder alloc] init];
}
```

12. Następnie zaimplementuj metodę, która będzie wywołana po wciśnięciu przycisku:

```
-(void)getCurrentLocation:(id)sender {
    locationManager.delegate = self;
    if ([locationManager respondsToSelector:@selector(requestWhenInUseAuthorization)]) {
        [locationManager requestWhenInUseAuthorization];
    }
    locationManager.desiredAccuracy = kCLLocationAccuracyBest;
    [locationManager startUpdatingLocation];
}
```

13. Dodaj następujące dwa wpisy do pliku Info.plist:

14. Niezbędne jest również zaimplementowanie metod wymaganych przez protokół CLLocationManagera. Jedna z nich będzie odpowiadać za obsługę błędów, druga zaś umożliwi aktualizowanie zawartości widoku kontrolera. Można skorzystać z przykładu poniżej:

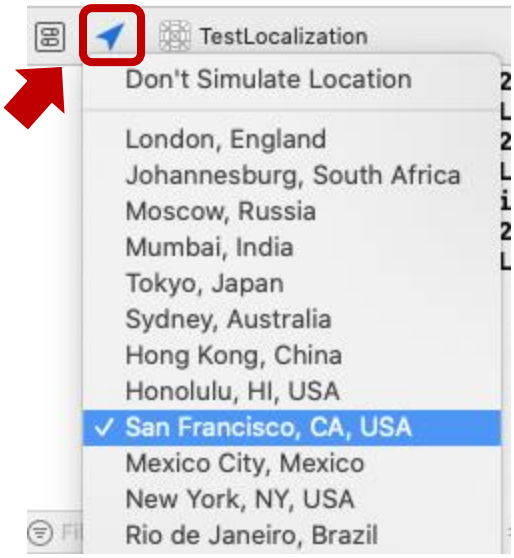
```
-(void)locationManager:(CLLocationManager *)manager didFailWithError:
    (NSError *)error {
    NSLog(@"didFailWithError: %@", error);
    UIAlertController *alertController = [UIAlertController
        alertControllerWithTitle:@"Error" message:@"Failed to get your
        location" preferredStyle:UIAlertControllerStyleAlert];

    UIAlertAction *okButton = [UIAlertAction actionWithTitle:@"Ok"
        style:UIAlertActionStyleDefault handler:^(UIAlertAction *action)
        {
        [self.view setBackgroundColor:[UIColor blueColor]];}];
    [alertController addAction:okButton];
    [self presentViewController:alertController animated:YES
        completion:nil];
}
-(void)locationManager:(CLLocationManager *)manager didUpdateLocations:
    (nonnull NSArray<CLLocation *> *)locations {
    NSLog(@"didUpdateLocations");

    CLLocation *currentLocation = [locations lastObject];
    if (currentLocation != nil) {
        _longtitudeText.text = [NSString stringWithFormat:@"%.8f",
            currentLocation.coordinate.longitude];
        _latitudeText.text = [NSString stringWithFormat:@"%.8f",
            currentLocation.coordinate.latitude];
    }

    [geocoder reverseGeocodeLocation:currentLocation
        completionHandler:^(NSArray<CLPlacemark *> * _Nullable
        placemarks, NSError * _Nullable error) {
        if (error == nil && [placemarks count] > 0) {
            self->placemark = [placemarks lastObject];
            self->_addressText.text = [NSString stringWithFormat:@"%@
                %@\n%@ %@\n%@\n%@", self->placemark.subThoroughfare,
                self->placemark.thoroughfare, self->placemark.postalCode,
                self->placemark.locality, self-
                >placemark.administrativeArea, self->placemark.country];
        } else {
            NSLog(@"%@", error.debugDescription);
        }
    }];
}
```

15. Do weryfikacji poprawności działania kodu możesz wymusić zmianę lokalizacji, na przykład w następujący sposób:


TestLocalization
Don't Simulate Location
London, England
Johannesburg, South Africa
Moscow, Russia
Mumbai, India
Tokyo, Japan
Sydney, Australia
Hong Kong, China
Honolulu, HI, USA
San Francisco, CA, USA
Mexico City, Mexico
New York, NY, USA
Rio de Janeiro, Brazil