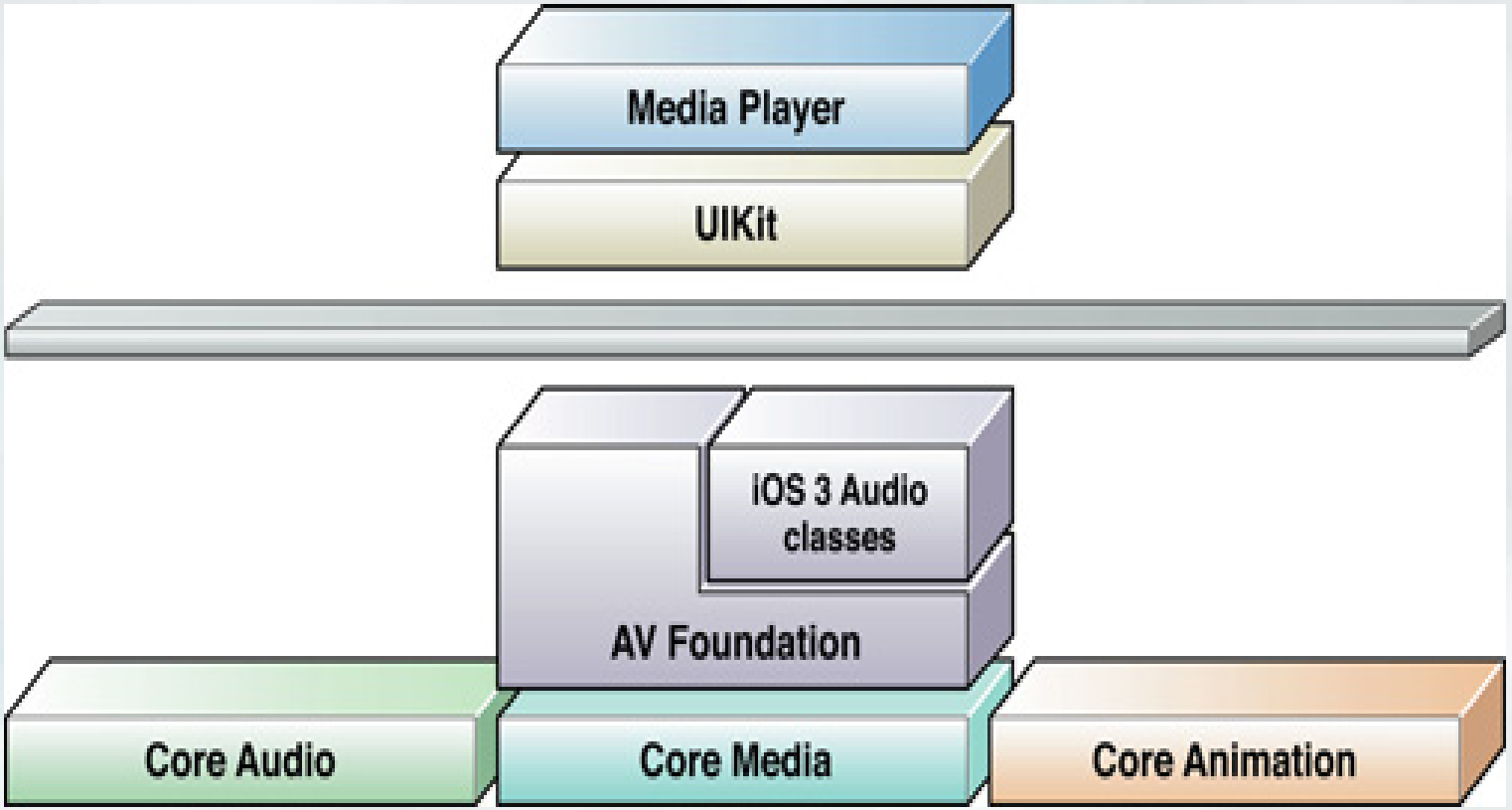
Работа с мултимедия. Използване на възможности за геолокация с MapKit

Боян Лазов

За какво ще става дума?

- Работа с мултимедия - възпроизвеждане и записване на видео/аудио
- Работа с MapKit и CoreLocation

Архитектура на свързаните с медиа framework-и в iOS

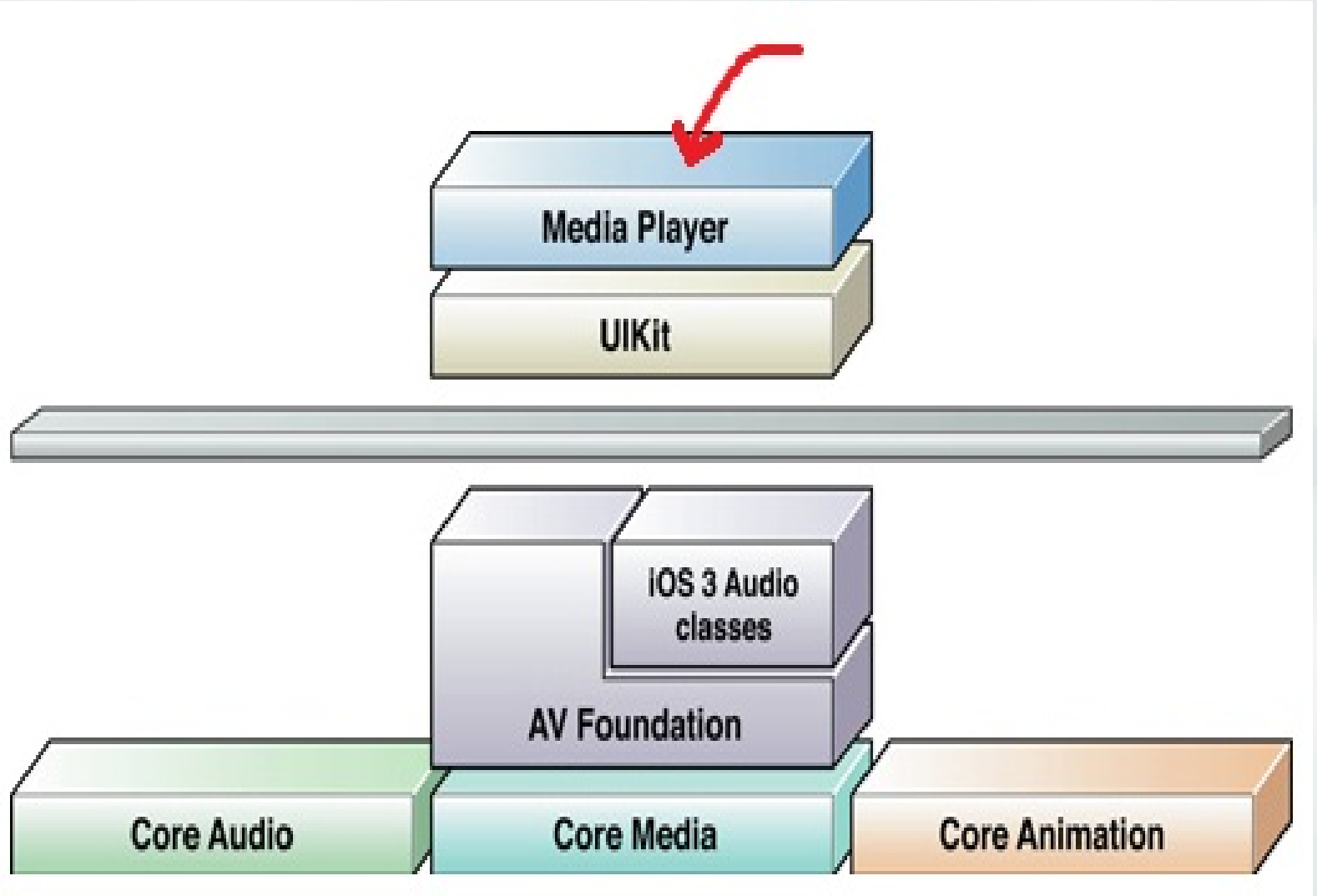


Кои от тях ще разглеждаме?

- Media Player
- iOS3 Audio classes

Както съветват Apple,
"You should typically use the highest-level abstraction available that allows you to perform the tasks you want"

Media Player



Media Player

- Позволява playback на видео и аудио по лесен начин, както и има интерфейс с вградени контроли за playback
- Дава достъп до iPod library
- Не дава много контрол (напр. не можем да изключим аудио пътека от видео файл), но е достатъчно добър в повечето случаи
- Ще разгледаме главно `MPMoviePlayerController`

MPMoviePlayerController

- Инициализира се с URL на ресурса
  `-initWithContentURL:(NSURL *)url`
- Стандартни функции като `play, pause, stop`
- Функциите за seek нямат ефект ако съдържанието се stream-ва
- Playback-а е във view, притежавано от movie player-а

- Можем да добавяме subview-та, за да показваме съдържание върху видео-то

- Можем да създаваме много обекти от този тип, но **само един movie player може да бъде активен**

.. contd

- По-интересни свойства са `contentURL, duration, playbackState`
- Позволява генерирането на thumbnail-и
- Позволява playback към устройства, които поддържат AirPlay
- Поддържаните (видео) формати са относително малко - `.mov, .mp4, .mpv, .3gp`
- За аудио - AAC-LC и MP3

Player notifications

- Player-ът може да ни уведомява за събития относно playback-а на видеото
- Регистрирането става през `NSNotificationCenter`
- Всяко събитие се идентифицира с някакъв низ (константа), а player-а се подава като `object` в съобщението, която получаваме
- За пълен списък от константите и какво се подава, погледнете в документацията :)

.. например

```
MPMoviePlayerController* player = ...
[[NSNotificationCenter defaultCenter]
    addObserver:self
       selector:@selector(playbackStateChanged:)
           name:MPMoviePlayerPlaybackStateDidChangeNotification
         object:player];


...
- (void)playbackStateChanged:(NSNotification *)note {
    MPMoviePlayerController* player = [note object];
    //.. other stuff
}
```


Demo

Генериране на thumbnails

- `thumbnailImageAtTime:timeOption:`
- `requestThumbnailImagesAtTimes:timeOption:`

Приемат параметри време(на) и time option
Time option определя дали да се стремим да вземем точно фрейма на даденото време или най-близкия key frame

Втория метод е асинхронен и комуникацията става с notification
(`MPMoviePlayerThumbnailImageRequestDidFinishNotification`)

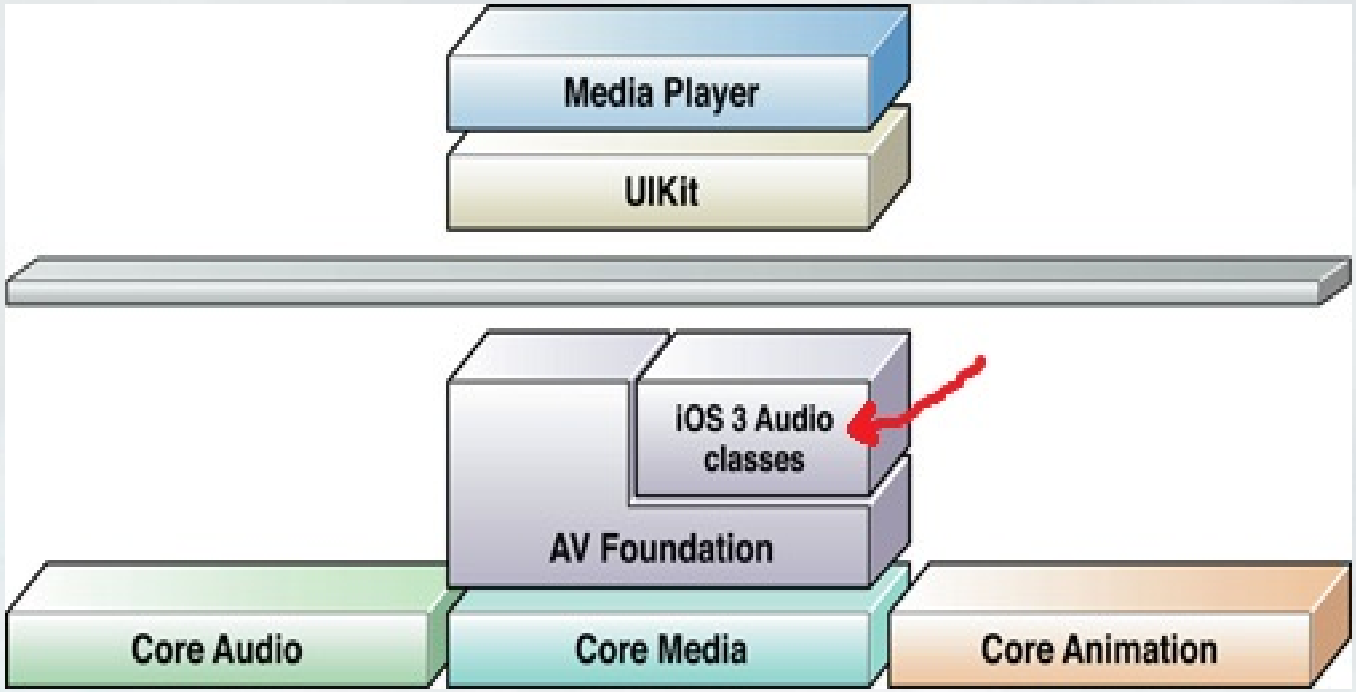
MPMoviePlayerViewController

- Прост view controller, който wrap-ва `MPMoviePlayerController` за показване на видео **full-screen**
- Вместо да показваме видеото веднага, можем да ползваме всякакви начини да представим view controller - tab bar, navigation bar, модално.

- Има `presentMoviePlayerViewControllerAnimated:` в категория на `UIViewController`


Demo

iOS3 Audio Classes



AV Foundation Audio-Related Classes

- `AVAudioPlayer` - за възпроизвеждане на аудио
- `AVAudioRecorder` - за записване на аудио
- `AVAudioSession` - конфигурация на поведението на приложението

Възпроизвеждане на аудио

`AVAudioPlayer`

- Playback на аудио в паметта или от файл.

- Отново стандартни функции като `play, pause, stop`
- Random seek

- Можем да имаме > 1 активен audio player
- Не съдържа UI компоненти, за разлика от `MPMediaPlayerController`

- За съжаление, не можем да stream-ваме с него

AVAudioPlayer misc

- Може да play-ва всички аудио формати, поддържани в iOS (`.aac, .aiff, .alac, .mp3, .wav, ...`)
- Синхронизация на няколко player-а с използването на `playAtTime:` и `deviceCurrentTime`
- `AVAudioPlayerDelegate` - информира ни кога аудиото е свършило/възникнала е грешка
- `settings` свойството дава най-различна информация за аудиото като bitrate, брой канали и т.н.

Примери

```
//Init with file
NSError* error = nil ;
NSURL* url = ...

AVAudioPlayer* player = [[AVAudioPlayer
alloc]                                                    i
nitWithContentsOfURL:url
                                        error:&error];

//Init with data from memory
NSData* data = ...
player = [[AVAudioPlayer alloc] initWithData:data error:&error];

//play
[player play];

//stop
[player stop];
//seek
player.currentTime = seekTime;
```


Demo

Записване на аудио

`AVAudioRecorder`

Можем да

- Записваме, докато потребителя не спре записа
- Записваме с определена продължителност
- pause/resume на recording
- подаваме най-различни опции (като формат на резултатния файл, bit depth, bit rate и т.н.)
- (!) Трябва да настроим аудио сесията, за да можем да записваме (за това по-подробно при `AVAudioSession`)

AVAudioRecorder misc

- Поддържани формати:

Audio encoder/recording format	Hardware-assisted encoding	Software-based encoding
AAC (MPEG-4 Advanced Audio Coding)	Yes, starting in iOS 3.1 for iPhone 3GS and iPod touch (2nd generation) Yes, starting in iOS 3.2 for iPad	Yes, starting in iOS 4.0 for iPhone 3GS and iPod touch (2nd generation)
ALAC (Apple Lossless)	-	Yes
iLBC (internet Low Bitrate Codec, for speech)	-	Yes
IMA4 (IMA/ADPCM)	-	Yes
Linear PCM (uncompressed, linear pulse-code modulation)	-	Yes
μ-law and a-law	-	Yes

- `AVAudioRecorderDelegate` - съобщава кога е свършило записването (дали е било успешно) /евентуално при грешка

Примери

```
//Setup recorder
NSNumber* format = [NSNumber numberWithInt:kAudioFormatAppleIMA4];
NSDictionary* settings =
[NSDictionary                                    dictionaryWithOb
jectsAndKeys:format, AVFomatIDKey, nil];
NSString* filename = [NSTemporaryDirectory()
                                              stringByAppendingPathComp
onent:@"temp.caf"];
NSURL* location = [NSURL fileURLWithPath:filename];
AVAudioRecorder* recorder = [[AVAudioRecorder alloc]
                        initWithURL:location settings:settings
                        error:&error];
//record
[recorder record];

//stop recording
[recorder stop];
//pause & resume
[recorder pause];
...
[recorder record];
```


Demo

Настойки на поведението

`AVAudioSession`

- Singleton, създава се при стартиране
- Занимава се с аудио "поведението" на приложението - напр. дали нашето аудио може да се смесва с музиката от iPod, какво да прави при аларми, и т.н.

- Като цяло служи като посредник между нашето приложение и iOS

- Описва почти всичко - дали имаме право на audio input/output въобще, mixing, ...

- Има поведение по default - playback, без записване, без миксиране

AVAudioSession more

- Но Apple казват че не е добре да минаваме въобще без конфигуриране, понеже няма как така да се възстановяваме от прекъсвания (напр. обаждане)

- Сесията трябва да бъде активна за да са в сила правилата, които описва (кога може да се деактивира - при прекъсвания)

- Audio session category - описва набор от поведения за приложението; ползва се със `setCategory:error:`
- Категориите са `NSString` константи

AVAudioSession even more

- Категориите описват цялостно поведение на приложението, а понякога не са напълно подходящи
- Например - категорията `AVAudioSessionCategoryPlayAndRecord`не позволява миксиране с други приложения (а бихме могли да искаме ..)

- Има и C API - Audio Session Services
- set/get на property-та - `AudioSessionSetProperty`, `AudioSessionGetProperty`
- Позволява по-фин контрол
- Можем да мешаме обръщения към него с ползване на `AVAudioSession`

Примери

```
NSError* error = nil;

//Category usage
AVAudioSession* session = [AVAudioSession sharedInstance];
[session setCategory:AVAudioSessionCategoryPlayAndRecord
               error:&error];

//Allow mixing with other apps, use C API
UInt32 mixWithOthers = 1; //YES
OSStatus status = AudioSessionSetProperty (
 kAudioSessionProperty_OverrideCategoryMixWithOthers,
 sizeof(mixWithOthers),
 &mixWithOthers);


[session setActive:YES error:&error];
```


Demo

Мултимедия чрез HTML5

- HTML5 предлага поддръжка за аудио и видео с <audio> и <video> елементите
- Има възможност за стандартни контроли
- Не специфицира кои са поддържаните формати - всеки browser има възможност да си избира (но главно Ogg, H. 264)

MapKit + CoreLocation

- MapKit съдържа главно view за карта + начини за overlay-ването и, както и някои типове и функции за по-лесна работа с точки от картата

  Има и вграден reverse geocoder

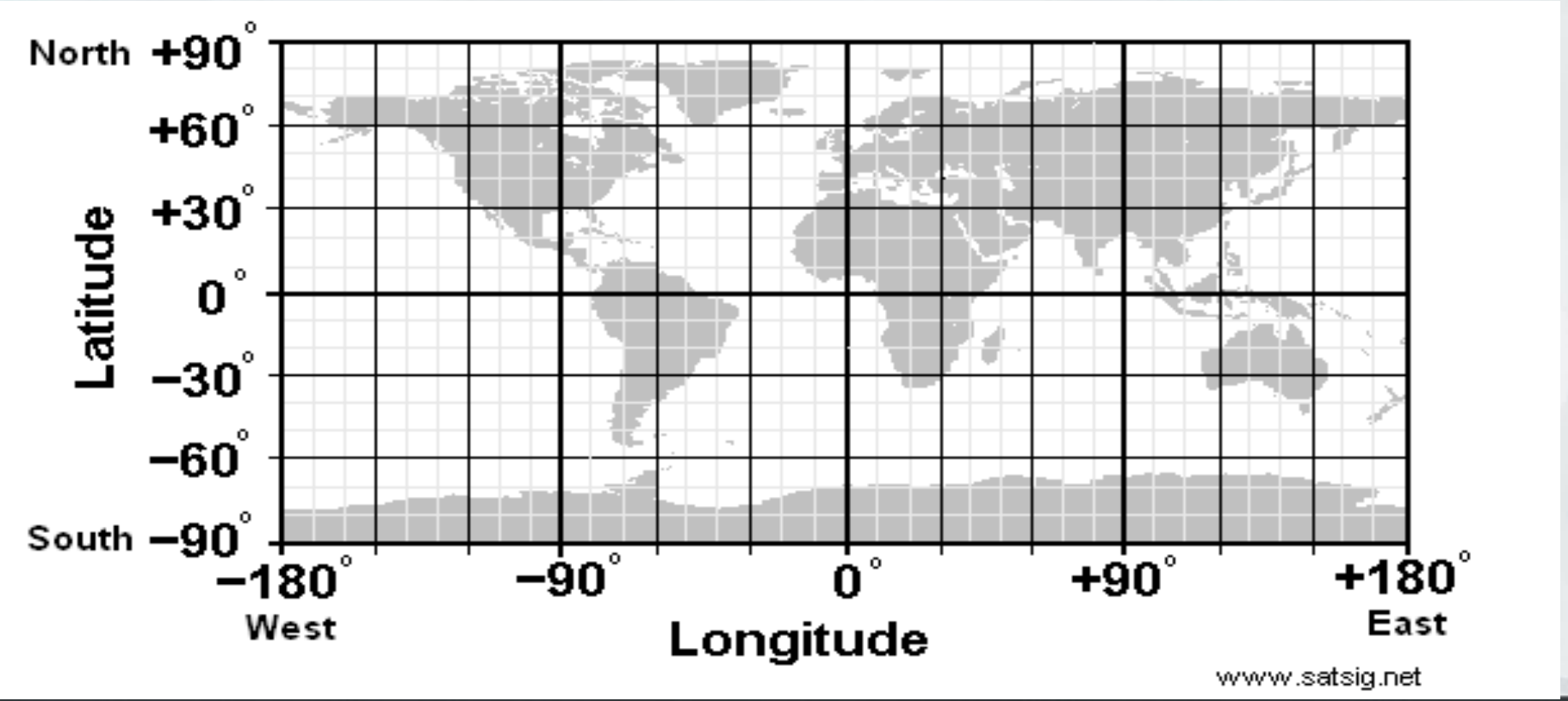
- CoreLocation е framework за определяне на текущата позиция на потребителя.

- Доста често MapKit се използва заедно с CoreLocation

Map basics

- Една точка от повърхността на Земята се определя от 2 координати - ширина (latitude) и дължина (longitude). И двете са в градуси

- latitude ∈ [-90.0, 90.0]
- longitude ∈ [-180.0, 180.0]



MapKit basic data types

- lat/lng се представят с `CLLocationDegrees`, което е typedef на `double`
- `CLLocationCoordinate2D` е двойка (lat, lng) - описва точка
- `MKCoordinateSpan` е двойка (latDelta, lonDelta) - същата структура като по-горе, но различна семантика
- `MKCoordinateRegion` е двойка централна точка + coordinate span. Описва регион от картата

Всички изброени са прости структури и имат методи, които ги създават

Map view basics

`MKMapView`

- View, което показва карта;
- Ползва услуги на Google, съответно има някакви terms of service

- Може да бъде добавяна допълнителна информация върху картата
- Има стандартни жестове за интеракция - flick за преместване, pinch за zoom in/out

- Може да показва позицията на потребителя

MKMapView more

- Главните свойства за действия с позицията на картата са `centerCoordinate` и `region`
- Има делегат - `MKMapViewDelegate`, който ни информира за събития (например смяна на региона, зареждане на данни, събития с анотациите и др.)
- Показването на позицията на user-а е с `showsUserLocation`
- За съжаление без адекватна поддръжка на (дискретни) zoom levels - трябва да си правим сметките сами (но има подходящи за това категории на `MKMapView` в интернет)

Анотации по картата

- `MKAnnotation` protocol - описва "модела" на анотацията (стандартно координата, опционално title, subtitle)
- Добавяме към картата с `addAnnotation:` (съответно махаме с `removeAnnotation:`)
- Делегата ни "пита" за view-то, съответстващо на анотация с `mapView:viewForAnnotation:` - иска се да върнем `MKAnnotationView`

MKAnnotationView

- Грижи се как ще изглежда анотацията на картата, т.е. свързано е с конкретна анотация
- Най-често се слага картинка - свойството `image`

- Обектите от този тип могат да бъдат преизползвани (като клетките в таблица) - картата просто ни пита за view-тата на анотациите, които са видими в момента

- Може да бъде select-вано - и да показва т.нар. callout view


Demo

CoreLocation

- Главно функционалност за определяне на местоположението на устройството
- Има опции за следене на региони
- От iOS 5 нататък, има и geocoder (замества предишния `MKReverseGeocoder`)

Определяне на местоположение

`CLLocationManager`

- Следи за промени в положението
- Два основни начина - стандартен и следене само на големи промени. Втория е по-неточен, но пък използва много по-малко батерия

- Ако ползваме стандартното следене, можем да му даваме желана точност и distance filter
- Пази и последно получения location

- Получаваме резултатите чрез делегат - `CLLocationManagerDelegate`

CLLocationManager more

- "Искаме" update-и със `startUpdatingLocation/ startMonitoringSignificantLocationChanges`
- Спираме със съответните stop методи

- Резултатите ги получаваме в делегата - `locationManager: didUpdateToLocation:fromLocation:`

- Можем да проверим дали location services са позволени - `locationServicesEnabled` (потребителя може да ги изключи глобано или за приложение)

Следене на региони

- Възможно е от iOS4 нататък
- Региона се описва с `CLRegion`. Идентифицира се с `identifier`
- "Регистрираме" региона с `startMonitoringForRegion:desiredAccuracy:`
- (!) Регионите, които нашето приложение следи, са в `monitoredRegions` на (който и да е) location manager. Те се запазват между стартиранията на приложението
- (!) Ако потребителя влезе/излезе в даден регион, който следим, и нашето приложение не е пуснато, то се стартира

.. cont'd

- Важно е да сравняме обектите от `CLRegion` по идентификатор, понеже не е гарантирано че ще получим същите (като адреси)
- Можем да проверяваме дали можем да следим региони - `regionMonitoringAvailable`, `regionMonitoringEnabled`

CLGeocoder

- От iOS5; замества по-стария `MKReverseGeocoder`
- Може да извършва и forward, и reverse geocoding (forward = адрес -> координати, reverse = координати -> адрес)
- Приема completion блокове, а не използва делегат


Demo

Q&A

Благодаря Ви за вниманието!