

# Документация по проекту **AR Navigation**

Сопрачёв Андрей

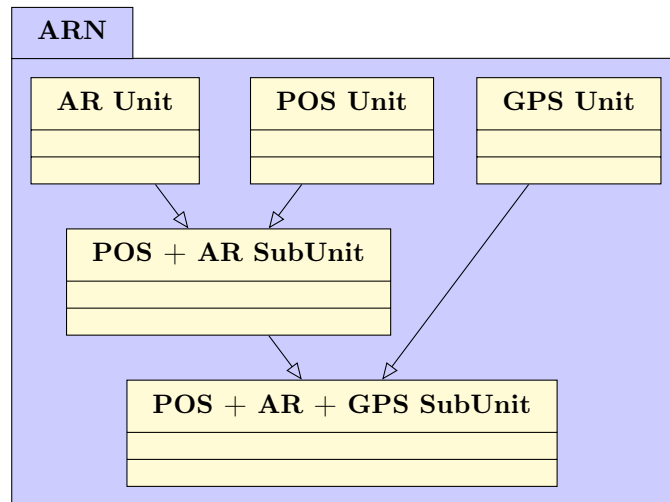
Версия: 0.1.4  
March 9, 2020

# Contents

<b>I</b>	<b>Основное описание проекта</b>	<b>3</b>
<b>1</b>	<b>Система Event</b>	<b>3</b>
<b>II</b>	<b>Деление на модули</b>	<b>3</b>
<b>2</b>	<b>Вступление</b>	<b>3</b>
<b>3</b>	<b>Независимые</b>	<b>4</b>
3.1	AR Unit . . . . .	4
3.1.1	Некоторые примитивы . . . . .	5
3.1.2	ARInterface . . . . .	5
3.1.3	Эмуляция AR сессии . . . . .	6
3.1.3.1	AREventRecorder . . . . .	6
3.1.4	Алгоритм создания AR сцены в Unity . . . . .	6
3.1.5	Пример AR сцены . . . . .	7
3.2	GPS Unit . . . . .	7
3.2.1	GPSInterface . . . . .	7
3.2.2	Алгоритм создания GPS сцены в Unity . . . . .	7
3.2.3	Пример GPS сцены . . . . .	7
3.3	POS Unit . . . . .	7
3.3.1	Задача позиционирования . . . . .	7
<b>4</b>	<b>Надстройки</b>	<b>8</b>
4.1	Pos + AR subUnit . . . . .	8
4.1.1	Описание . . . . .	8
4.1.2	ARMapTool . . . . .	8
4.1.3	Использование . . . . .	8
4.1.3.1	Создание ARMapScriptable . . . . .	8
4.2	Pos + GPS + AR subUnit . . . . .	9
4.2.1	Описание . . . . .	9
4.2.2	GPSMapTool . . . . .	9
4.2.3	Использование . . . . .	9
4.2.3.1	Создание GPSMapScriptable . . . . .	9

## Part I

# Основное описание проекта



## 1 Система Event

## Part II

# Деление на модули

## 2 Вступление

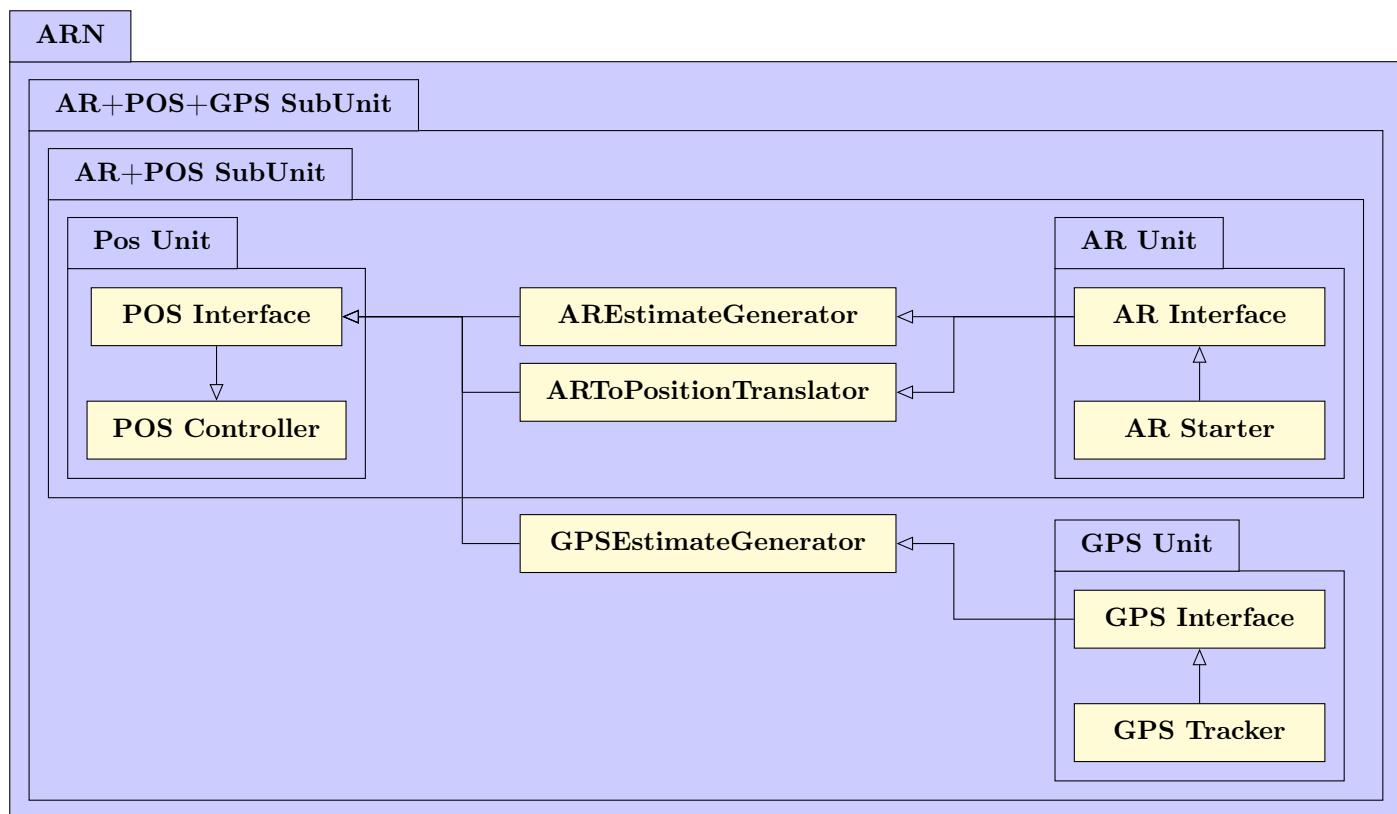
Работа приложения разделена на несколько функциональных модулей, некоторые из которых являются независимыми, а остальные – надстройками.

Независимые:

1. AR Unit — модуль дополненной реальности, прослойка между нативными плагинами и общей системы
2. GPS Unit — модуль позиционирования по GPS
3. Pos Unit — модуль решающий задачу перевода координат из локальной в глобальную системы координат

Надстройки:

1. Pos + AR SubUnit — надстройка над **Pos Unit** и **AR Unit** для проброски ивентов между ними
2. Pos + GPS + AR SubUnit — надстройка над **Pos + AR SubUnit** и **GPS Unit** для проброски ивентов между ними



### 3 Независимые

#### 3.1 AR Unit

Задача AR Unit — предоставить приложению уровень абстракции над ARKit и ARCore плагинами Unity.

Является независимым модулем.

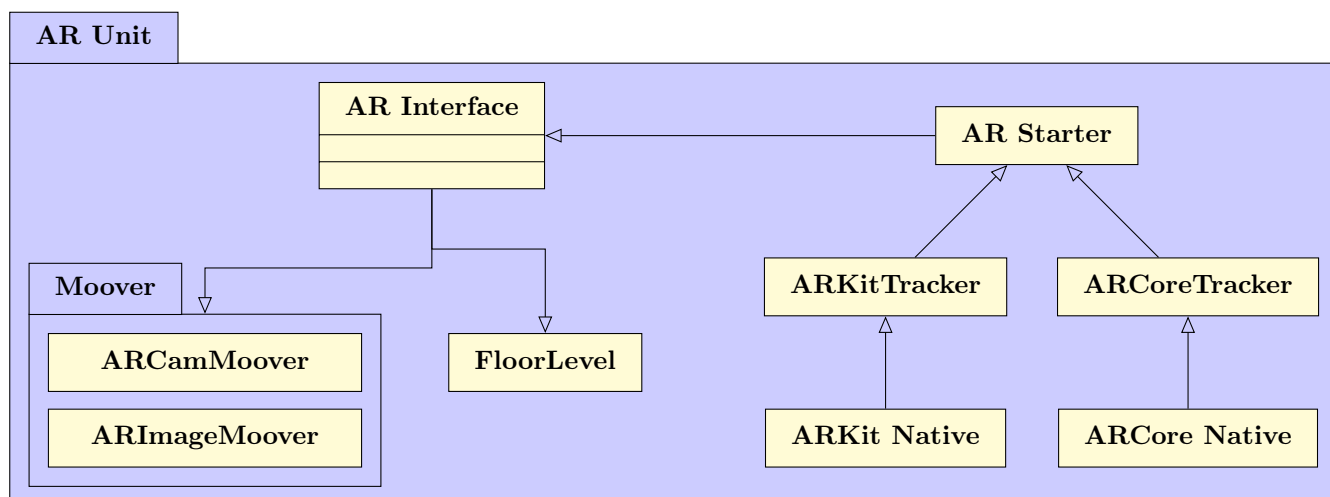
Модуль реализован на системе Events и предоставляет в использование **ARInterface**

Для работы модуля на сцене необходим префаб **ARStarter** с дочерними **ARKitTracker** и **ARCoreTracker**.

При запуске сцены **ARStarter** активирует "Tracker" соответствующий текущей платформе (ios — ARKitTracker, android — ARCoreTracker).

Tracker выполняет функцию проброски api между нативным Unity Plugin'ом и ARInterface.

На объекте Tracker выполняются настройки для запуска AR сессии конкретной платформы.



### 3.1.1 Некоторые примитивы

ARTransform	ARImage	ARPlane
position : Vector3 rotation : Quaternion	name : String position : Vector3 rotation : Quaternion	identifier : String position : Vector3 rotation : Quaternion extent : Vector3

ARStatus	ARTrackingState
Stopped Initializing Running Unsupported Failed	ARTrackingStateUnsupported ARTrackingStateNotAvailable ARTrackingStateLimited ARTrackingStateNormal

### 3.1.2 ARInterface

Разделён на две логические части — делегаты состояний и функции их вызывающие.

Делегаты:

1. OnARTransformUpdate(ARTransform) — обновление координаты устройства в пространстве
2. OnARCameraProjectionMatrixUpdate(Matrix4x4) — обновление параметров камеры (fov etc)
3. OnImageAdd(ARImage) — первое появление AR метки в сцене
4. OnImageUpdate(ARImage) — обновление положения существующей AR метки
5. OnImageRemoved(ARImage) — удаление AR метки со сцены ( $\neq$  выход за пределы экрана, обычно вызывается в ARKit при остановке сцены)
6. OnPlaneAdd(ARPlane) — первое появление ARPlane в сцене
7. OnPlaneUpdate(ARPlane) — обновление положения существующей ARPlane
8. OnPlaneRemoved(ARPlane) — обновление положения существующей ARPlane
9. OnStatusChange(ARStatus) — изменение статуса AR сцены
10. OnTrackingStateChange(ARTrackingState) — изменение статуса позиционирования
11. OnTrackingStateReasonChange(ARTrackingStateReason) — информация о текущем статусе позиционирования (например: недостаточно освещения)
12. OnStartSession() — запуск сессии
13. OnRestartSession() — перезапуск сессии на лету
14. OnStopSession() — остановка сессии
15. OnSessionFailed() — критическая ошибка в сессии приводящая к её остановке (например: запрещён доступ к камере)
16. OnChangePlaneMode(bool) — изменение состояния трекинга плоскостей

### 3.1.3 Эмуляция AR сессии

AR Unit предоставляет объекты для полной эмуляции всех событий AR сессии.

Все необходимые файлы находятся в Assets/Units/ARUnit/Fake все дальнейшие пути указаны относительно этой директории.

Для полной эмуляции перетащить в сцену Prefabs/FAKE\_AR. Его дочерние объекты определяют поведение симуляции.

1. FakeARMain — отвечает за эмуляцию статусов сессии
2. Camera position AR generator — отвечает за эмуляцию положения камеры в пространстве
3. FakeImage — отвечает за эмуляцию трекинга картинки

**3.1.3.1 AREventRecorder** Объект позволяющий записать и сохранить в файл все события происходящие во время AR сессии, а после этот файл воспроизводить. Для использования добавить на сцену Prefabs/SessionRecorder.

### 3.1.4 Алгоритм создания AR сцены в Unity

Все необходимые файлы находятся в Assets/Units/ARUnit все дальнейшие пути указаны относительно этой директории.

1. Создание сцены:
  - (a) Перетащить на сцену префаб /Prefabs/ARUnit
  - (b) При необходимости отключить объект ARFloorCalculate отвечающий за расчёт уровня пола
  - (c) На основную камеру добавить скрипт ARCamMoover и указать эту камеру в настройках ARKitTracker и ARCoreTracker
2. Настройка:
  - (a) IOS
    - i. Создать в проекте UnityARKitPlugin/ARReferensImagesSet и перетащить его на ARKitTracker в соответствующие поле
    - ii. Создать в проекте UnityARKitPlugin/ARReferensImage для каждой желаемой метки, и указать ей текстуру и физический размер (ширину). Заполнить ими созданный ReferensImagesSet.
    - iii. На объект трекинга добавить скрипт ARImageMover и в его имя указать имя метки
  - (b) Android
    - i. Создать в проекте GoogleARCore/SessionConfig и перетащить его на ARCoreTracker в соответствующие поле
    - ii. В проекте выделить необходимые метки и создать GoogleARCore/AugmentedDataBase. Перетащить получившийся объект на созданный SessionConfig.
    - iii. На объект трекинга добавить скрипт ARImageMover и в его имя указать имя метки
3. Запуск  
Вызвать функцию ARInterface.StartARSession() из UI или другого скрипта. После инициализации ARInterface.ARStatus перейдёт в состояние Running и сессия будет успешно запущена.
4. Остановка  
Для остановки сессии вызвать функцию ARInterface.StopARSession()

### 3.1.5 Пример AR сцены

Пример сцены расположен в Assets/Units/ARUnit/Example/FullARUnitExample в нём реализованы все возможности ARUnit

## 3.2 GPS Unit

Задача GPS Unit предоставить уровень абстракции над location service.

Модуль реализован на системе Events и предоставляет в использование GPSInterface.

Для работы модуля на сцене необходим префаб GPSTracker.

### 3.2.1 GPSInterface

Разделён на две логические части — делегаты состояний и функции их вызывающие.

Делегаты:

1. OnStartGPS(desiredAccuracyInMeters, updateDistanceInMeters) — запуске GPS трекинга с заданными параметрами погрешности
2. OnStopGPS() — остановка GPS трекинга
3. OnGPSStatusUpdate(GPSServiceStatus) — событие обновления GPS статуса
4. OnGPSUpdate(GPSInfo) — событие обновления координаты
5. OnStartCompass() — запуске компаса
6. OnStopCompass() — остановка компаса
7. OnGPSCompassUpdate(GPSCompassInfo) — событие обновления азимута

### 3.2.2 Алгоритм создания GPS сцены в Unity

Все необходимые файлы находятся в Assets/Units/GPSUnit все дальнейшие пути указаны относительно этой директории.

1. Создание сцены  
Перетащить на сцену префаб /Prefabs/GPSTracker
2. Запуск  
Вызвать функцию GPSInterface.StartGPS() для отслеживания позиционирования и GPSInterface.OnStartCompass() для отслеживания азимута
3. Отслеживание  
Подписаться на события GPSInterface.OnGPSUpdate и GPSInterface.OnGPSCompassUpdate
4. Выключение  
Вызвать функцию GPSInterface.StopGPS() и GPSInterface.StopCompass()

### 3.2.3 Пример GPS сцены

Пример сцены расположен в Assets/Units/GPSUnit/Example/FullGPSUnitExample в нём реализованы все возможности GPSUnit

## 3.3 POS Unit

### 3.3.1 Задача позиционирования

Задачей позиционирования является вычисление матрицы перевода координат из **локальной** в **глобальную** систему координат.

Где **локальна** — система координат связанная с точкой запуска трекинга.

**Глобальная** — система координат связанная с положением на карте.

## 4 Надстройки

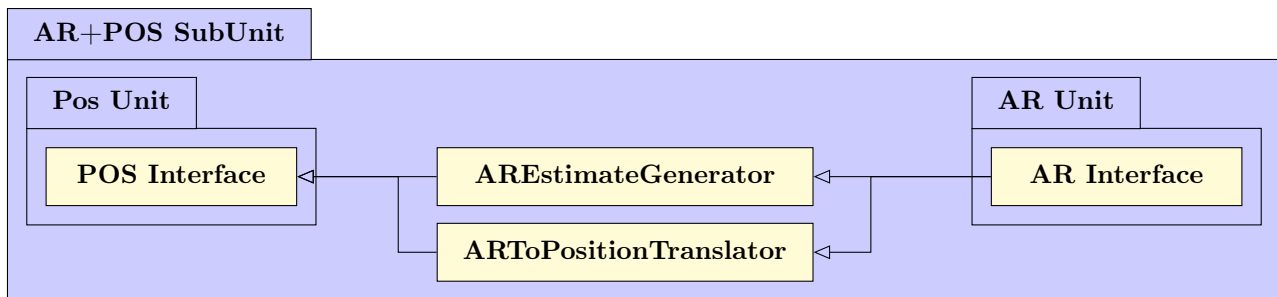
### 4.1 Pos + AR subUnit

#### 4.1.1 Описание

Задача надстройки — пробросить события между **ARUnit** и **PosUnit**.

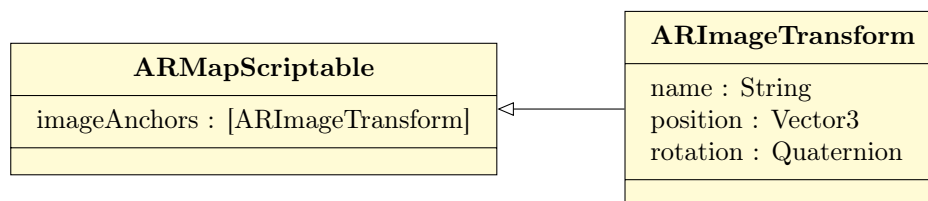
Состоит из двух объектов:

1. **AREstimateGenerator** — отправляет событие **OnEstimateAdd** при обнаружение метки
2. **ARToPositionTranslator** — отправляет событие обновления координаты устройства в локальной системе координат



#### 4.1.2 ARMapTool

Для корректной работы **AREstimateGenerator** необходимо знать координаты меток в глобальной системе координат, для этого используется **ARMapScriptable** настраиваемый с помощью **ARMapTool**.



#### 4.1.3 Использование

Для работы проброски ивентов перетащить префабы **AREstimateGenerator** и **ARToPositionTranslator** на сцену.

##### 4.1.3.1 Создание ARMapScriptable

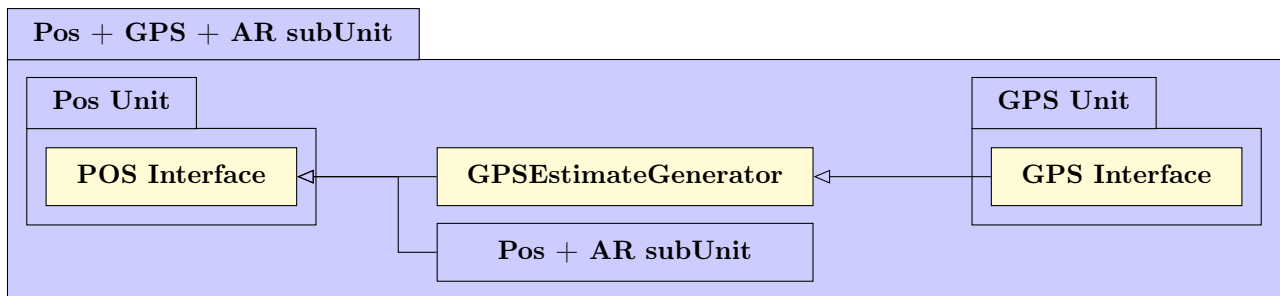
1. Создать в проекте **ARMapScriptable**.
2. Создать на сцене **GameObject** и добавить на него компонент **ARMapTool**
3. Указать в **TargetScriptable** объект созданный в первом пункте
4. Перетащить нужное количество префабов **A4 Target** на сцену дочерним к **ARMapTool**
5. Назначить им названия и координаты на сцене
6. На объекте **ARMapTool** нажать кнопку **Set Anchors**



## 4.2 Pos + GPS + AR subUnit

### 4.2.1 Описание

Задача надстройки — пробросить события между **GPSUnit** и **PosUnit** для этого используется **AREstimateGenerator**



### 4.2.2 GPSMapTool

Для корректной работы **GPSEstimateGenerator** необходимо знать связку между глобальной и геодезической системами координат, для этого используется **GPSMapScriptable** настраиваемый с помощью **GPSMapTool**.

GPSMapScriptable
latitude : float
longitude : float
altitude : float
localPos : Vector3
width : float
height : float
filter : Texture2D

1. latitude — широта
2. longitude — долгота
3. altitude — высота
4. localPos — координата в локальной системе
5. width — ширина карты в метрах
6. height — высота карты в метрах
7. filter — фильтр погрешности GPS, картинка в красный канал которой, записан коэффициент умножения текущей погрешности

### 4.2.3 Использование

Для работы проброски ивентов перетащить префаб **GPSEstimateGenerator** на сцену.

#### 4.2.3.1 Создание GPSMapScriptable

1. Создать в проекте **GPSMapScriptable**.
2. Перетащить на сцену префаб **GPSMapTool**
3. Изменить спрайт карты на свой

4. Указать в TargetScriptable объект созданный в первом пункте
5. Указать в инспекторе координаты PivotMain и PivotScale
6. Объекты PivotMain и PivotScale установить на сцене в нужные координаты по спрайту карты
7. Нажать кнопку **Set size** для масштабирования карты и приведения её глобального размера к геодезическому
8. Нажать кнопку **Set map** для записи в GPSMapScriptable