

# AR

дополняем реальность



**Данис Тазетдинов**  
Chief Software Engineer, EPAM



# Данис Тазетдинов

Chief Software Engineer, EPAM

# AR

дополняем реальность

- Евангелист платформ Apple
- 19 лет в разработке ПО
- 8 лет создаю мобильные приложения
- Ведущий подкаста Mobile People Talks



@edeniska



@edeniska



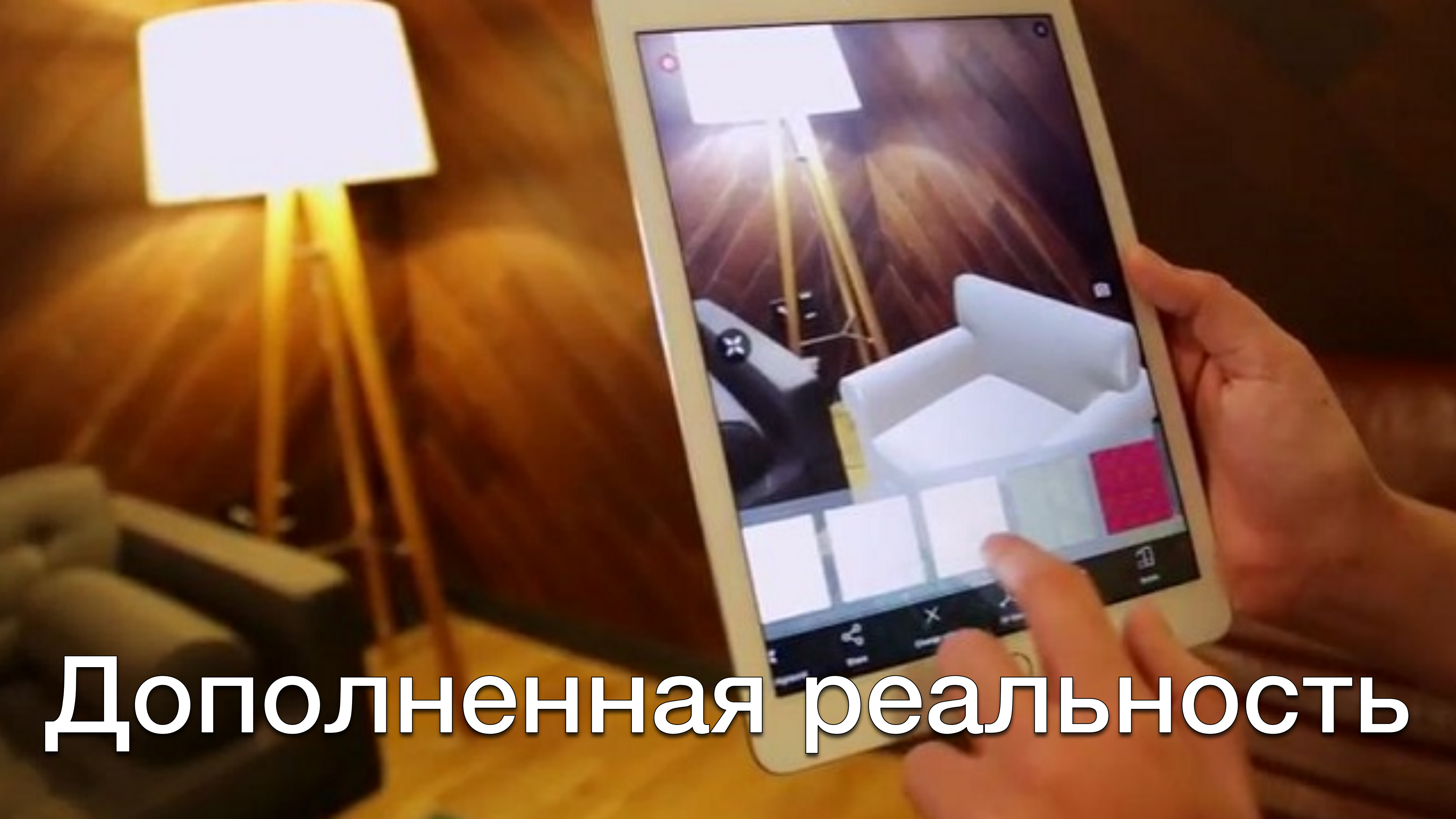
<https://facebook.com/dtazetdinov>

**MOBILE  
PEOPLE  
TALKS**



**Не забудьте подписаться :)**





**Дополненная реальность**

# Применение AR

- Ловля покемонов
- Расстановка мебели
- Примерка косметики
- Интерактивные книги
- Отправка глупых сообщений
- Навигация



# AR в ритейле



- Навигация по магазину
- Увеличение вовлеченности
- Геймификация шоппинга

# Сюрприз!

# GPS не работает

VSLAM и данные акселерометра – ARKit это умеет



**Еще сюрприз**

**AR и реальный мир – разные**

Оцифрованная карта привязывается к миру AR

# Навигация внутри помещений

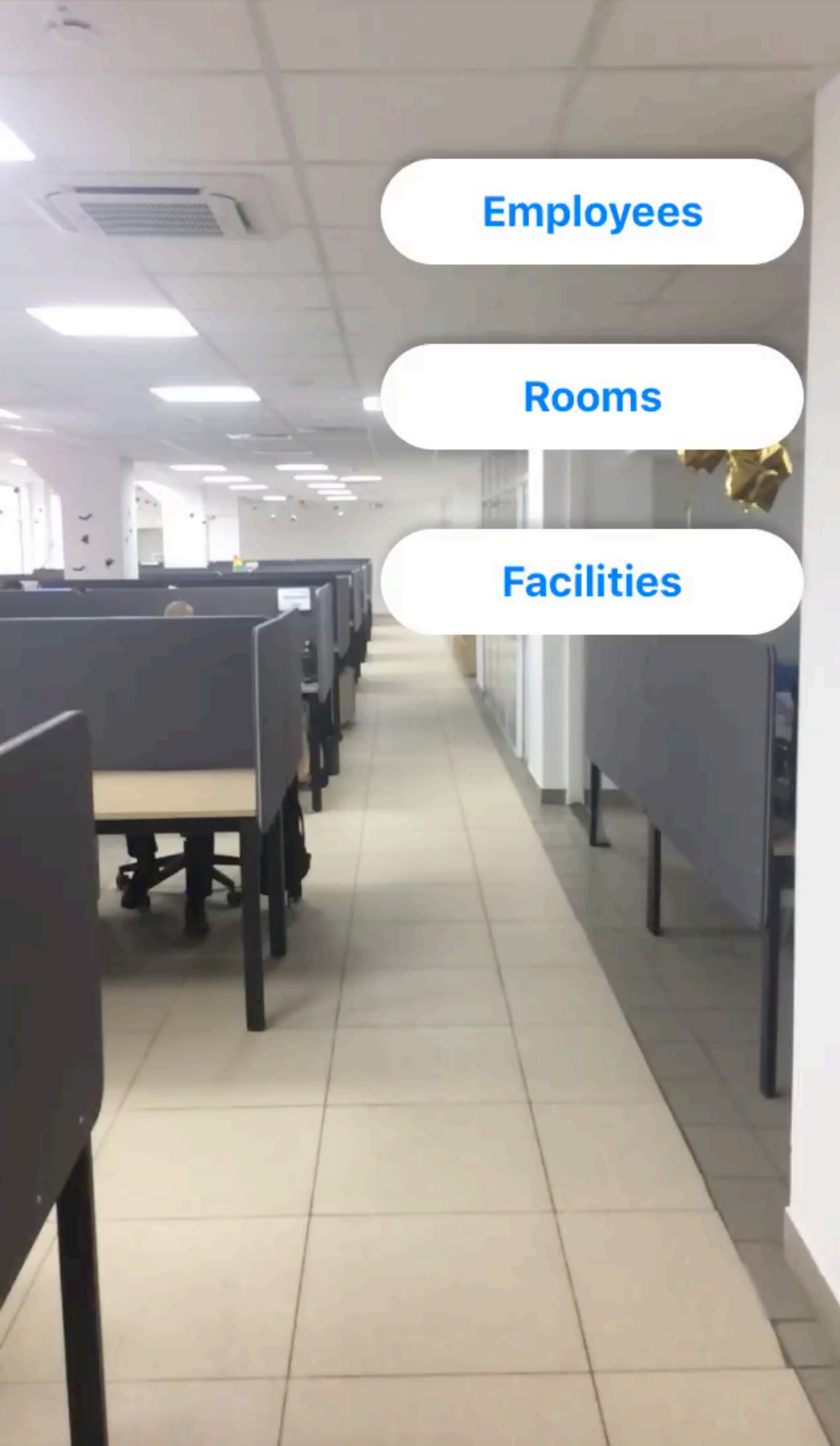
# Дорабатываем кейс

- Офис вместо магазина
- Сотрудники вместо товаров
- Схожее размещение
- Та же навигация
- Тот же стек технологий



# Приложение для сотрудника

- Адаптация новичков
- Виртуальный тур по офису
- Помощь в командировках



# OfficeApp

- Оцифрованная карта офиса
- Поиск сотрудников
- Построение маршрута
- Пути эвакуации



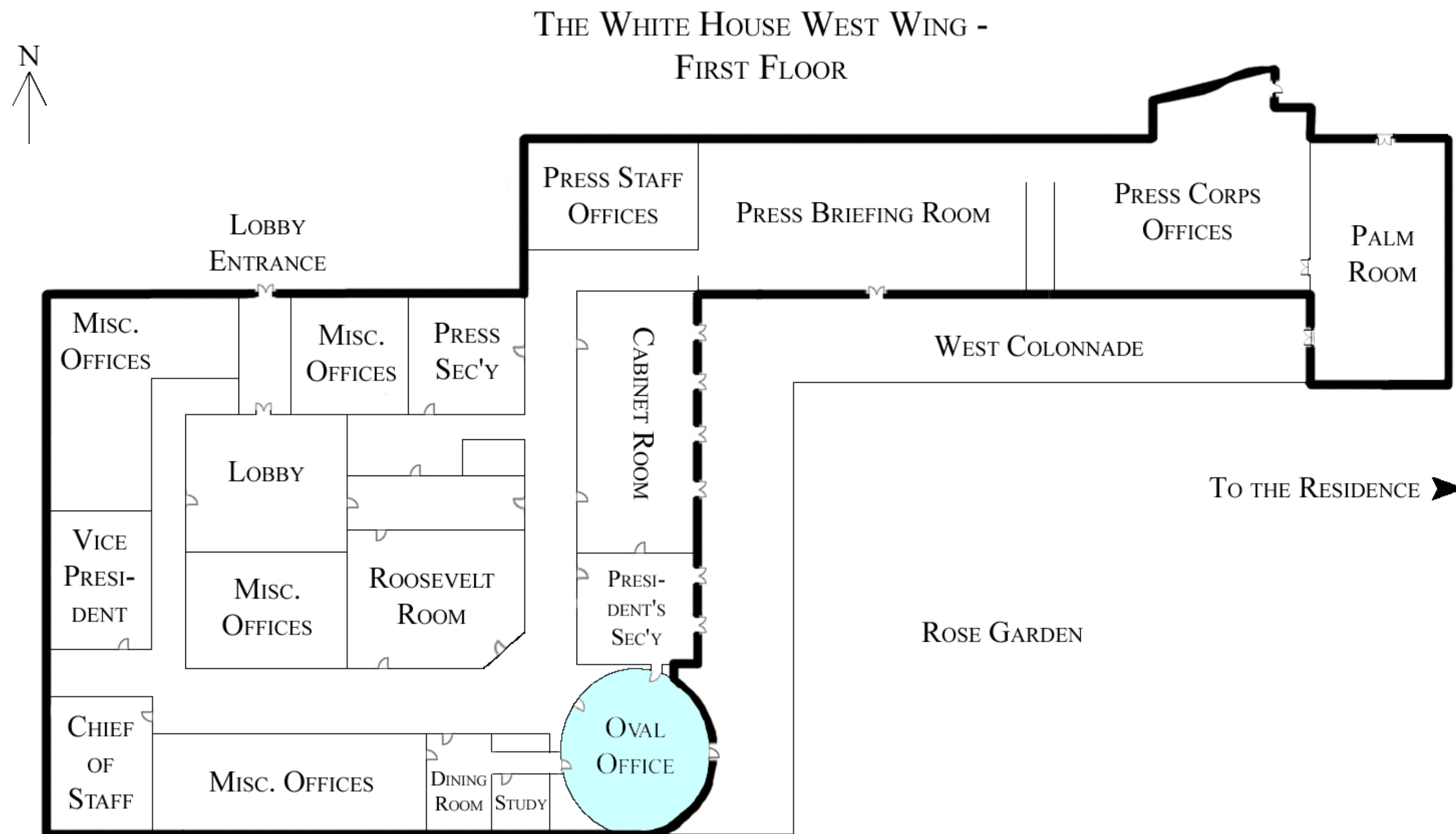
Что там внутри?

# Инструментарий

- ARKit + SceneKit + Swift
- Без сложных API вроде OpenGL, Metal
- Без сторонних фреймворков (Unity, Vuforia и т.п.)



# Строим карту мира



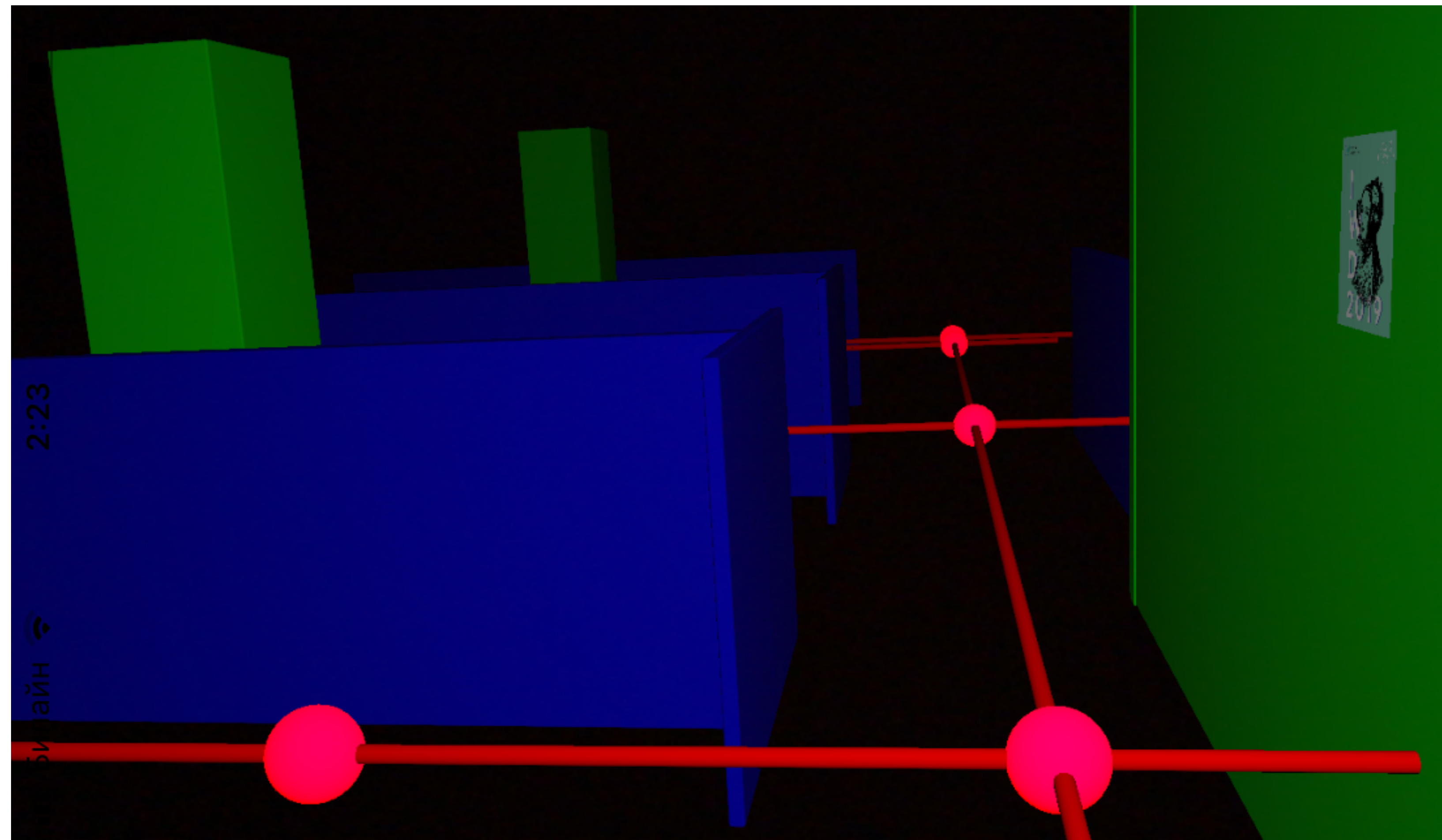
- Оцифровываем план офиса
- Расставляем маркеры
- Расставляем точки навигации
- Расставляем объекты



# Оцифровка карты

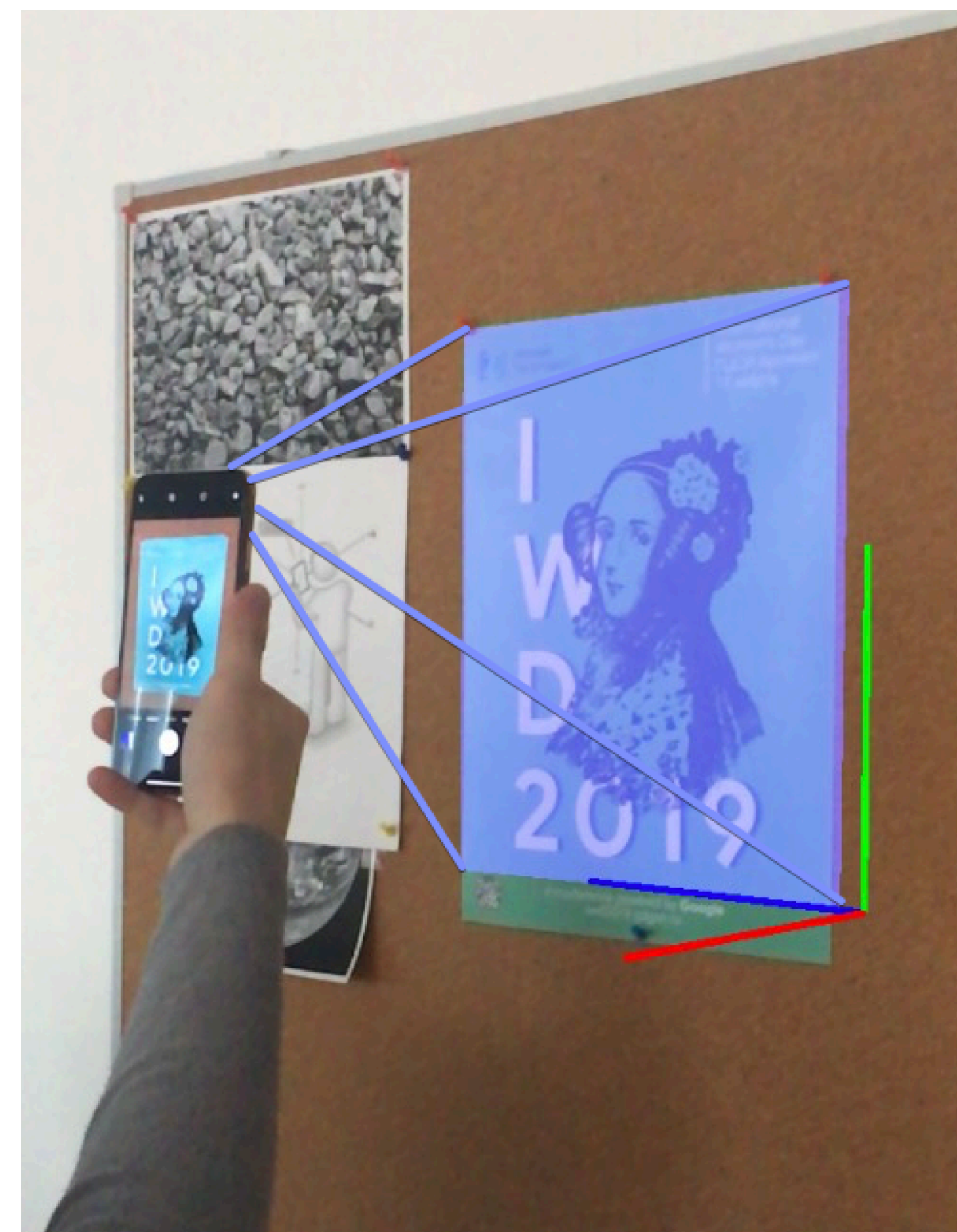
Создание цифровой карты офиса для навигации

- Оцифровываем план офиса
- Расставляем визуальные маркеры
- Расставляем точки навигации
- Расставляем искомые объекты



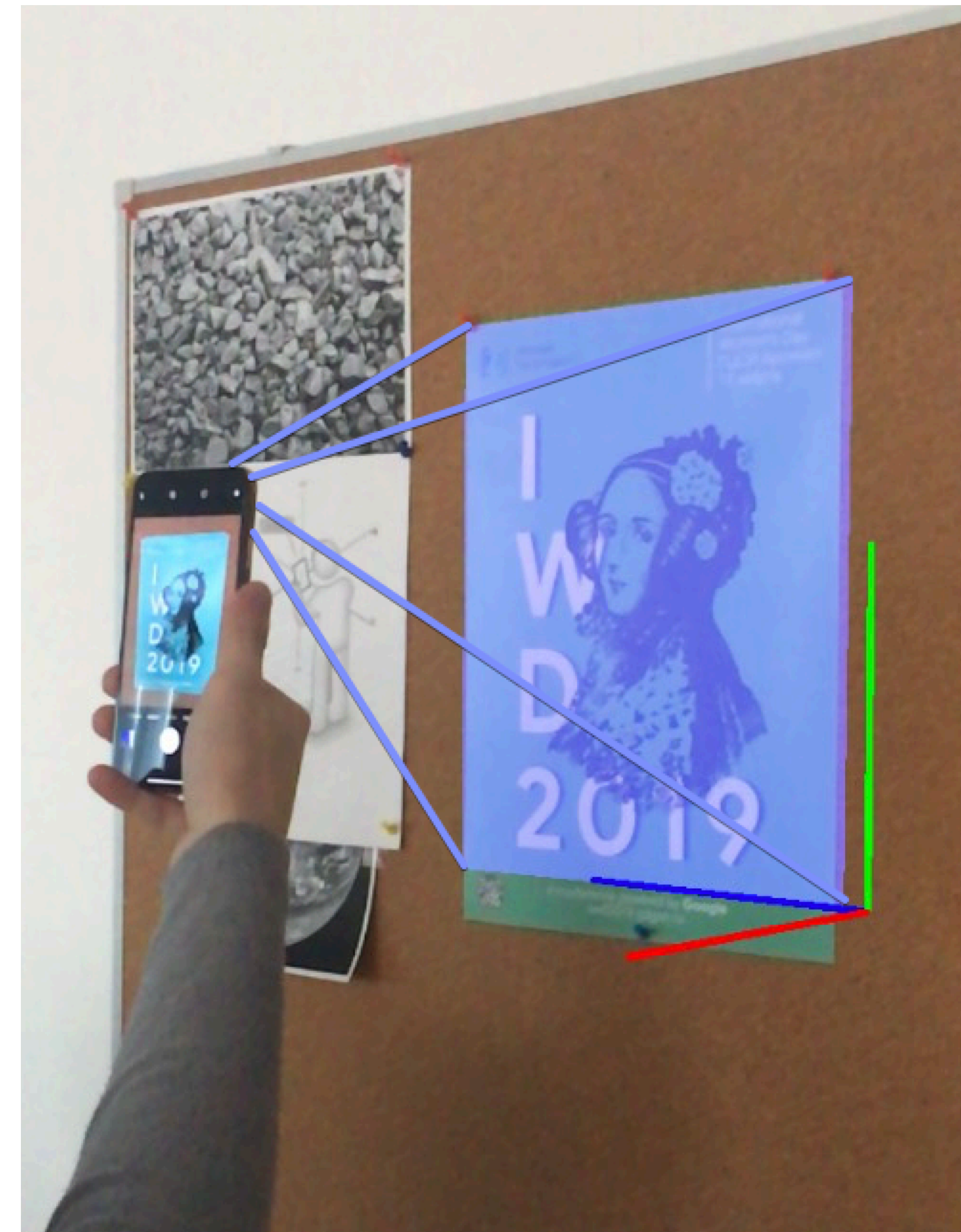
# Исходная позиция

- Движок AR распознает визуальный маркер
- Определяется положение устройства
- Размер изображения используется для определения масштаба в реальном мире
- Координаты изображения используются для определения положения пользователя в виртуальной карте



# Распознавание изображения

```
func startSession() {  
    let configuration = ARWorldTrackingConfiguration()  
    var referenceImages = Set<ARReferenceImage>()  
  
    storeMap.checkpoints.forEach { checkpointInfo in  
        let referenceImage = ARReferenceImage(  
            checkpointInfo.image.cgImage!,  
            orientation: .up,  
            physicalWidth: checkpointInfo.physicalWidth  
        )  
        referenceImage.name = checkpointInfo.imageName  
        referenceImages.insert(referenceImage)  
    }  
  
    configuration.detectionImages = referenceImages  
    sceneView.session.run(configuration, options: [])  
}
```



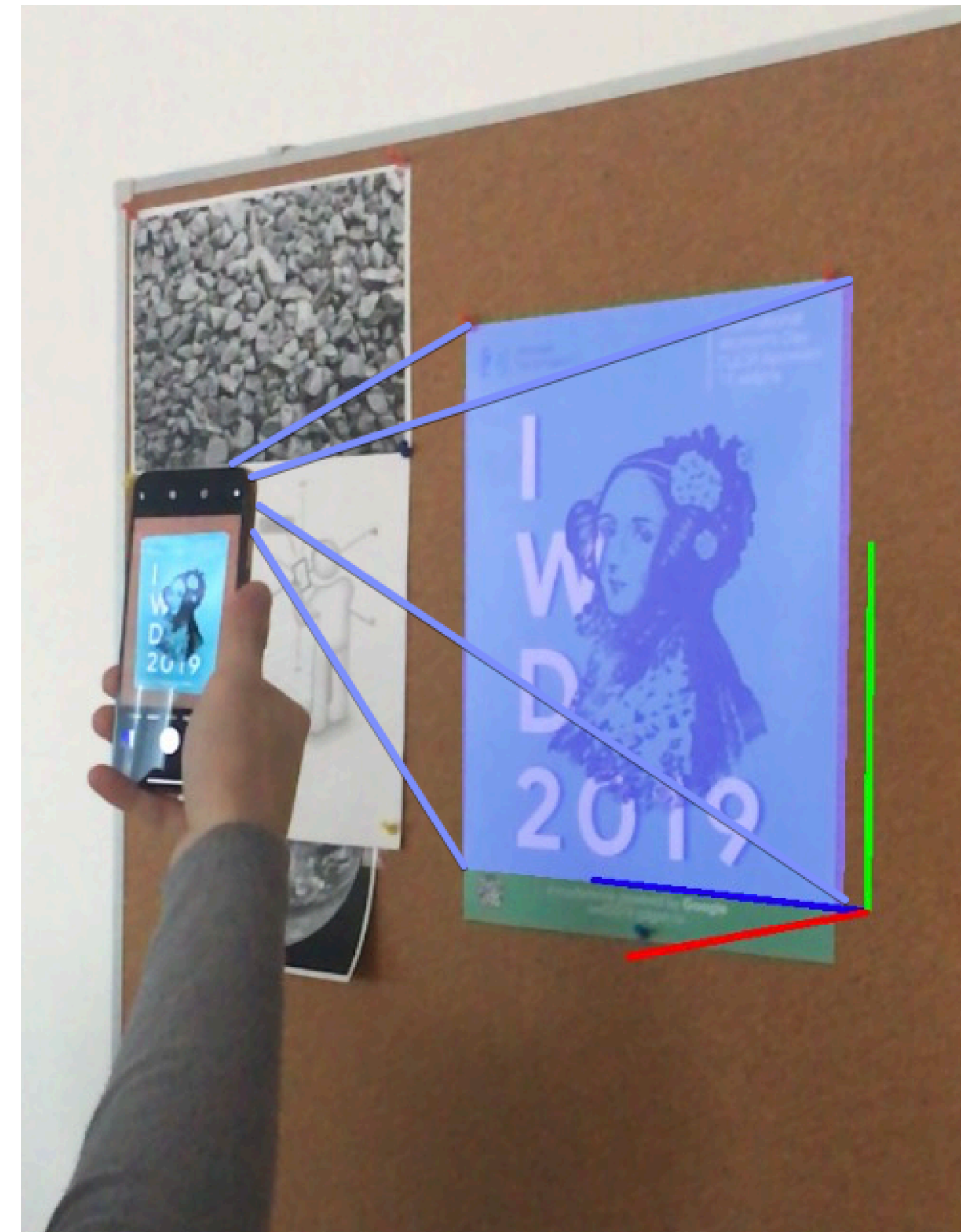
# Определение положения

Размеры картинки позволяют получить положение пользователя в виртуальном мире

```
func renderer(_ renderer: SCNSceneRenderer,
              didUpdate node: SCNNode,
              for anchor: ARAnchor) {
    guard let imageAnchor = anchor as? ARImageAnchor else {
        return
    }

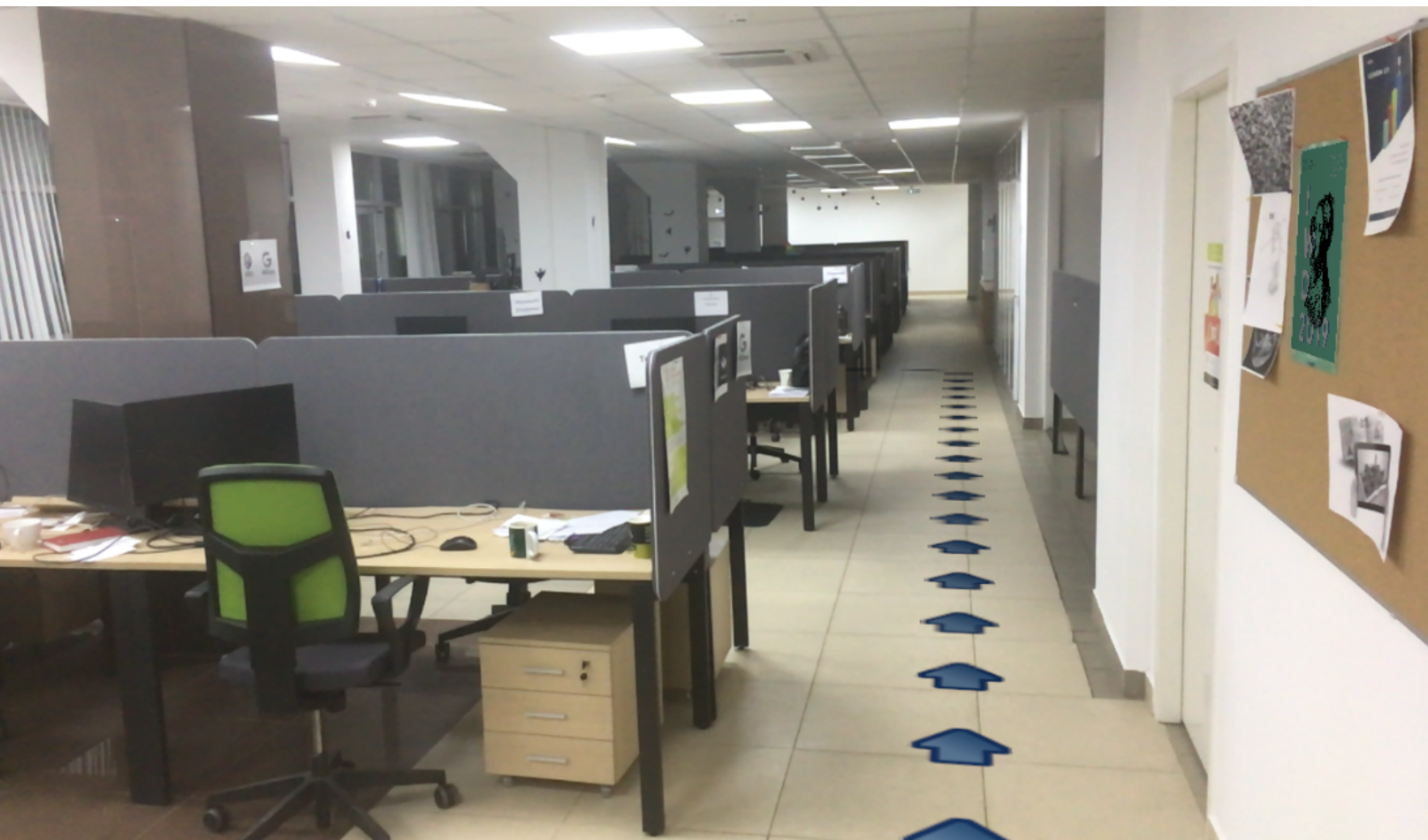
    // ARAnchor: A real-world position and orientation that can be used
    // for placing objects in an AR scene.
    // Image size & position relative to the AR World origin:
    let imageWidth = imageAnchor.referenceImage.physicalSize.width
    let imageHeight = imageAnchor.referenceImage.physicalSize.height
    let transform = SCNMatrix4(imageAnchor.transform)
    let imageName = imageAnchor.referenceImage.name

    virtualRealTransform = calculateTransform(objectTransform: transform)
    return createVerticalPlane(width: imageWidth, height: imageHeight,
                              color: UIColor.blue.withAlphaComponent(0.6))
}
```



# Наложение карты

Применение карты к виртуальному миру

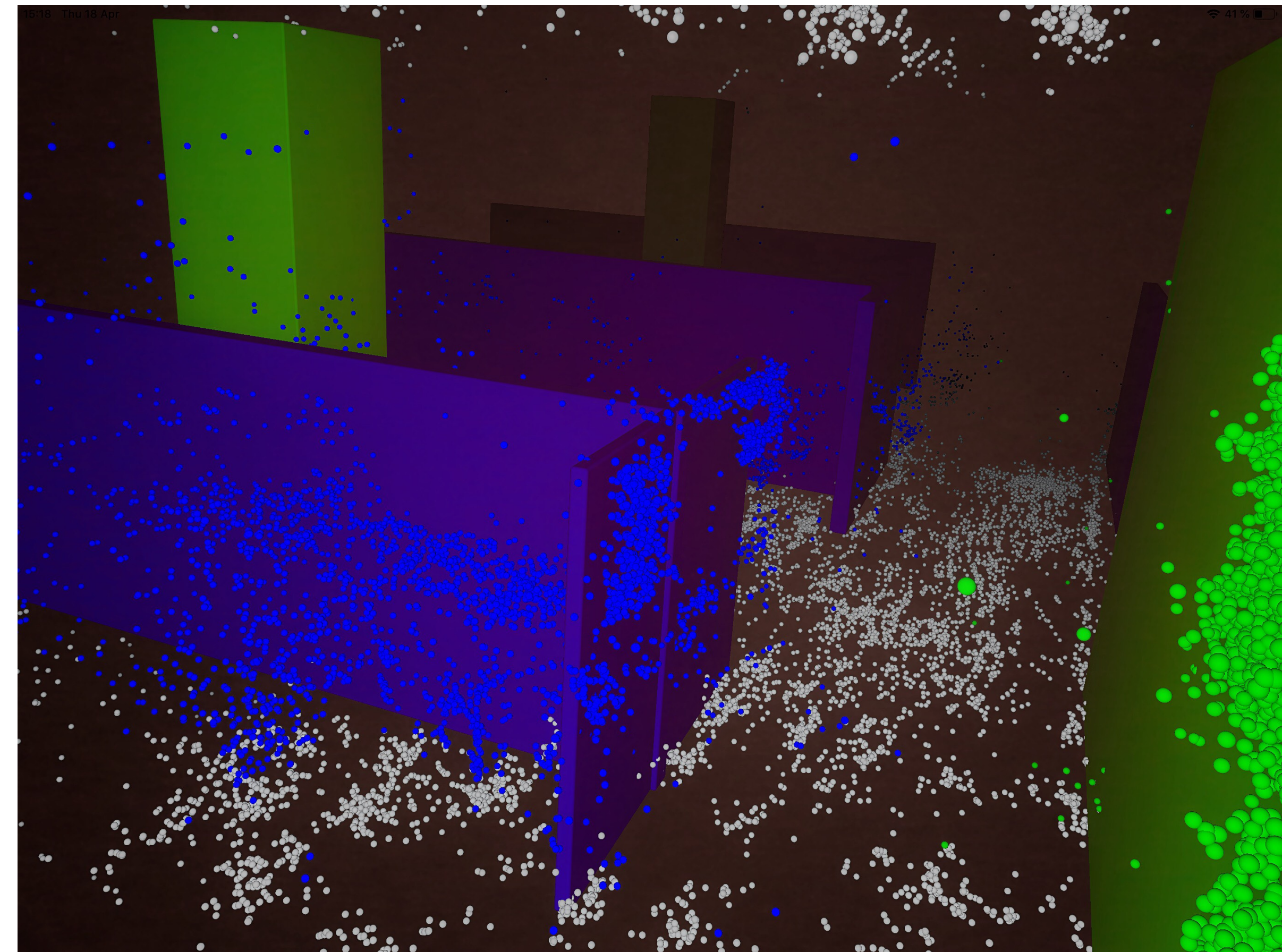


- Сканируем пространство
- Сканируем маркеры
- Ориентируем карту
- Строим маршрут

# Определение карты мира AR

## Сохранение и загрузка карты мира AR

- Сохраняем заранее отсканированную карту мира AR (опорные точки и точка отсчета координат)
- Движок AR может по нескольким опорным точкам определить положение устройства в мире AR



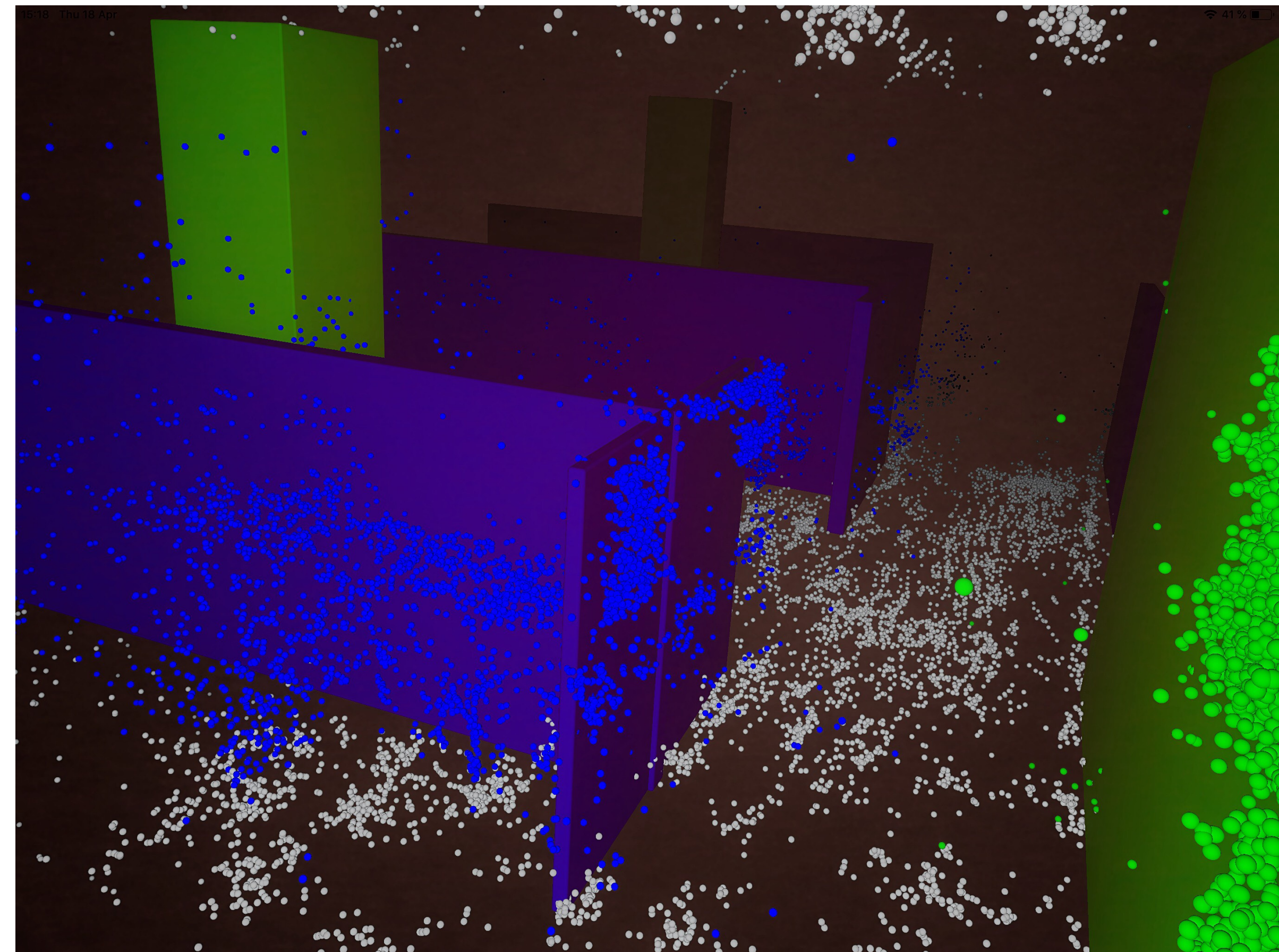
# Определение карты мира AR

## Сохранение и загрузка карты мира AR

```
sceneView.session.getCurrentWorldMap { worldMap, error in
    guard let worldMap = worldMap, error == nil else {
        return
    }

    FileManager.save(worldMap: worldMap)
}

if let worldMap = FileManager.readWorldMap() {
    let configuration = ARWorldTrackingConfiguration()
    configuration.initialWorldMap = worldMap
    sceneView.session.run(
        configuration,
        options: [.resetTracking, .removeExistingAnchors]
    )
}
```



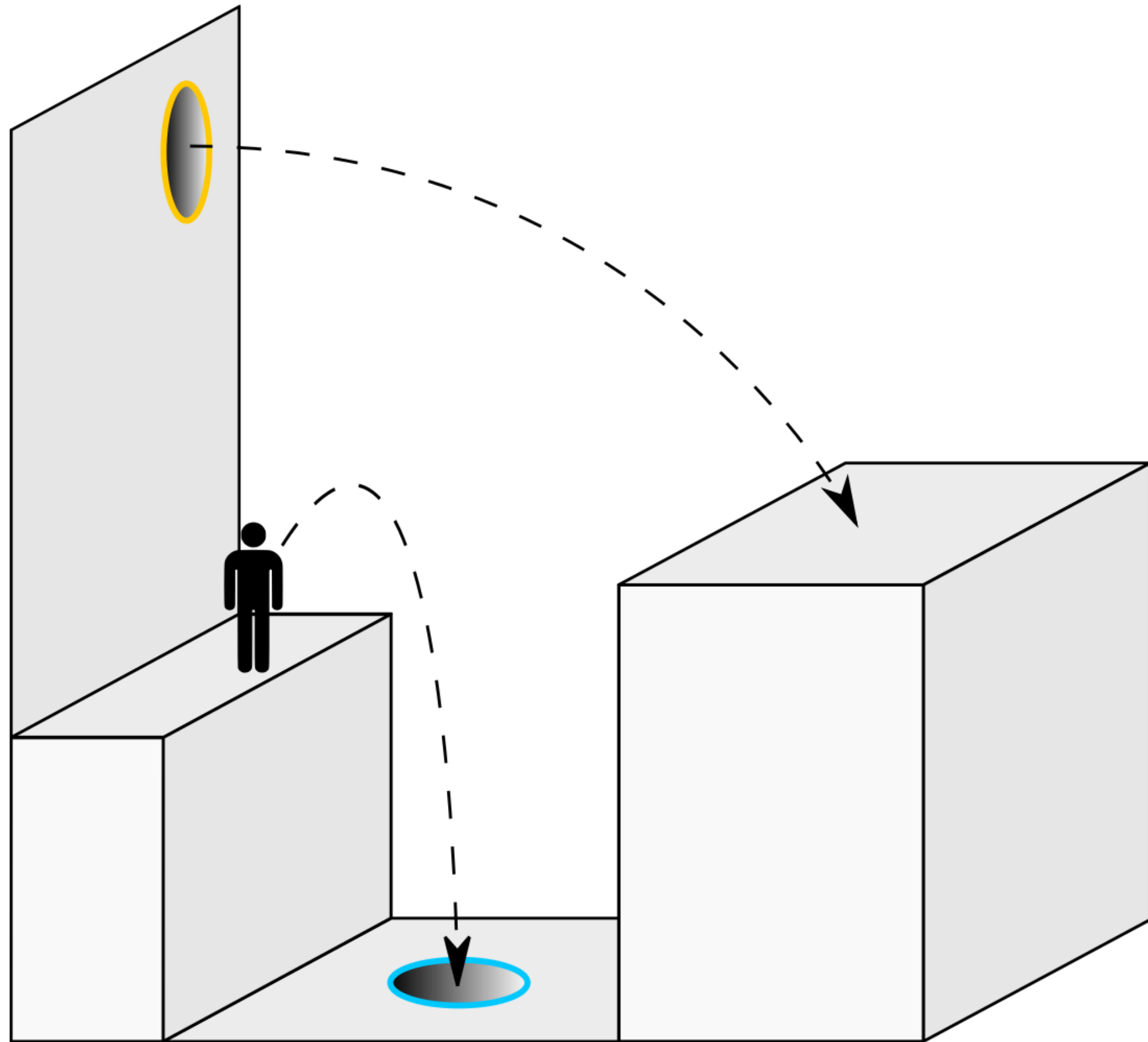
# Наложение карты офиса

```
func calculateTransform(_ anchorTransform: SCNMatrix4) -> SCNMatrix4 {  
    guard let currentCheckpointInfo = currentCheckpointInfo else {  
        return SCNMatrix4Identity  
    }  
  
    let node = SCNNode()  
    node.transform = SCNMatrix4Identity  
    node.eulerAngles.x = -.pi / 2  
    let rotationTransform = node.transform  
  
    let resultInAnchorCoordinate = SCNMatrix4Mult(  
        rotationTransform, anchorTransform  
    )  
  
    let result = SCNMatrix4Mult(  
        SCNMatrix4Invert(currentCheckpointInfo.transform),  
        resultInAnchorCoordinate  
    )  
  
    return result  
}
```





# Построение маршрута

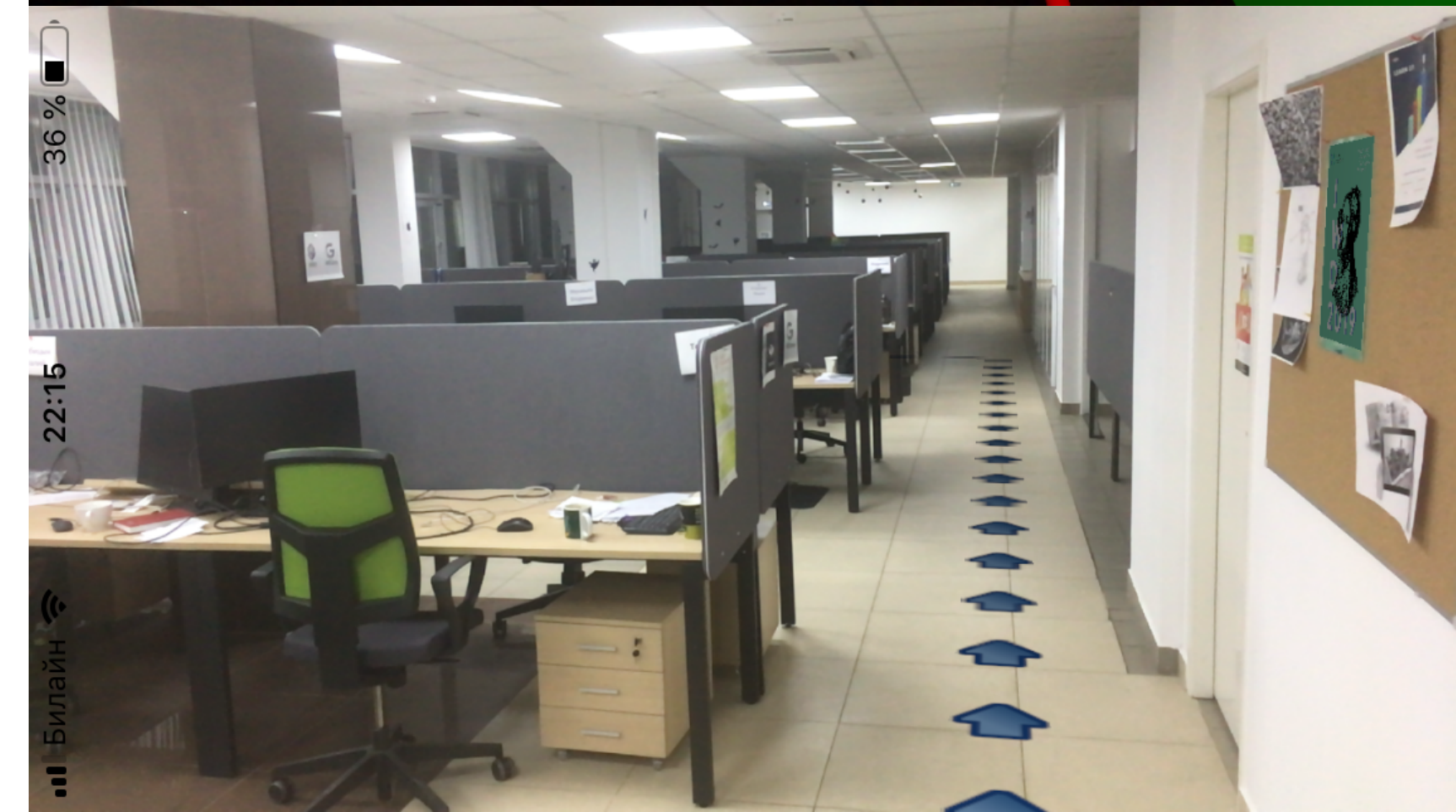
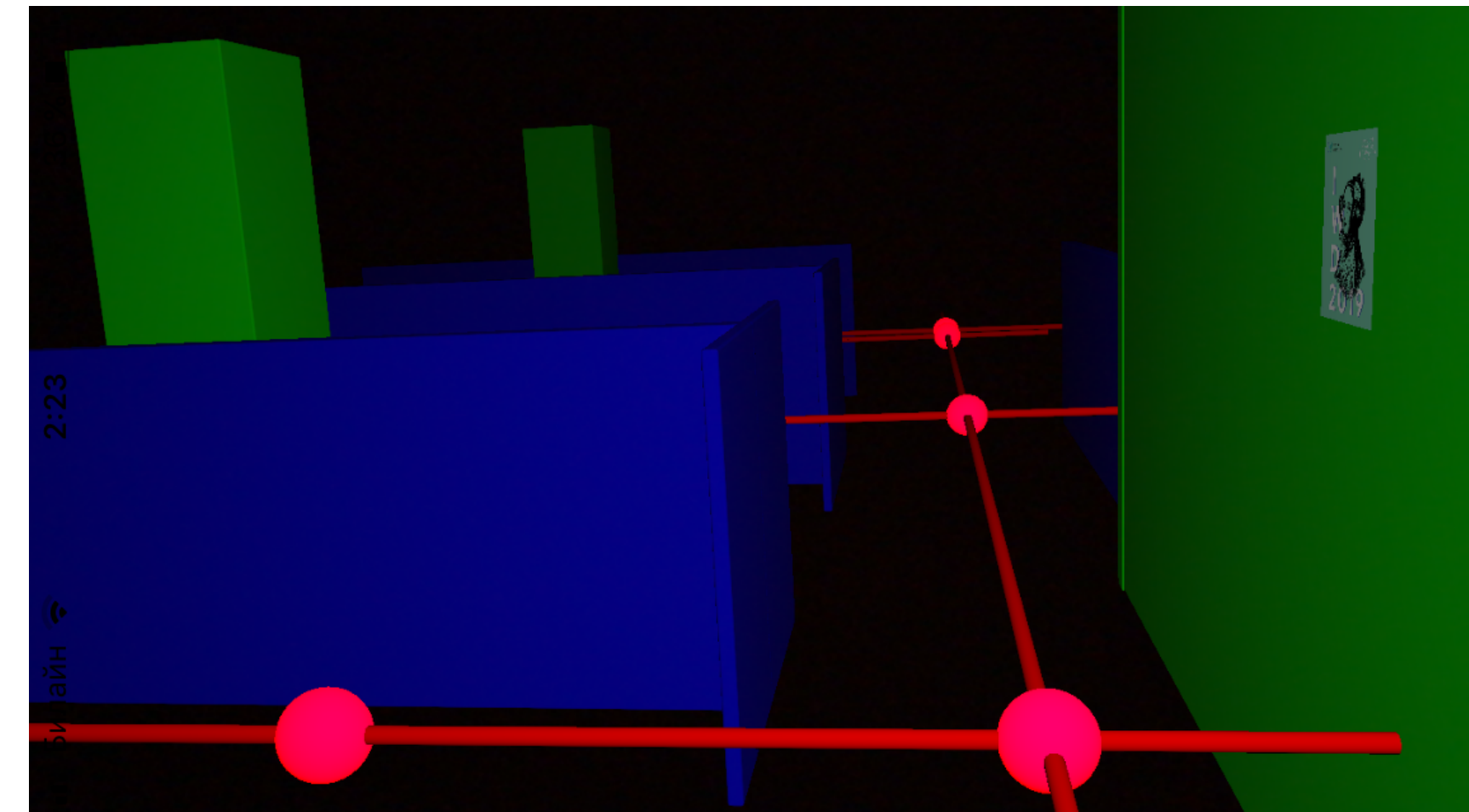


- С текущей позиции
- Только через проходимые зоны
- Алгоритм Дейкстры

# Построение маршрута

```
func routeNodes(targetPoint: WayPoint,  
                worldPosition: SCNVector3,  
                transform: SCNMatrix4 = SCNMatrix4Identity) -> [SCNNode] {  
    guard let firstPoint = nearestWayPoint(worldPosition: worldPosition,  
                                           transform: transform) else {  
        return [SCNNode]()  
    }  
  
    guard let path = WayPointPath.shortestPath(source: firstPoint,  
                                              destination: targetPoint) else {  
        return [SCNNode]()  
    }  
    return routeNodes(path: path)  
}
```

```
func routeNodes(path: WayPointPath) -> [SCNNode] {  
    var routeNodes = [SCNNode]()  
    for index in (0...path.array.count).reversed() {  
        if index > path.array.count - 2 {  
            continue  
        }  
        let node = buildRouteNode(firstPoint: path.array[index + 1],  
                                secondPoint: path.array[index])  
        routeNodes.append(node)  
    }  
    return routeNodes  
}
```





СТЕНЫ

# За стеной

Реальные стены должны скрывать виртуальные объекты



# Реализация «ОККЛЮЗИИ»



Используем `renderingOrder`,  
чтобы сделать невидимыми объекты:

```
node.renderingOrder = -100
```

```
guard let geometry = node.geometry else {  
    return  
}
```

```
geometry.firstMaterial?.diffuse.contents = UIColor.black
```

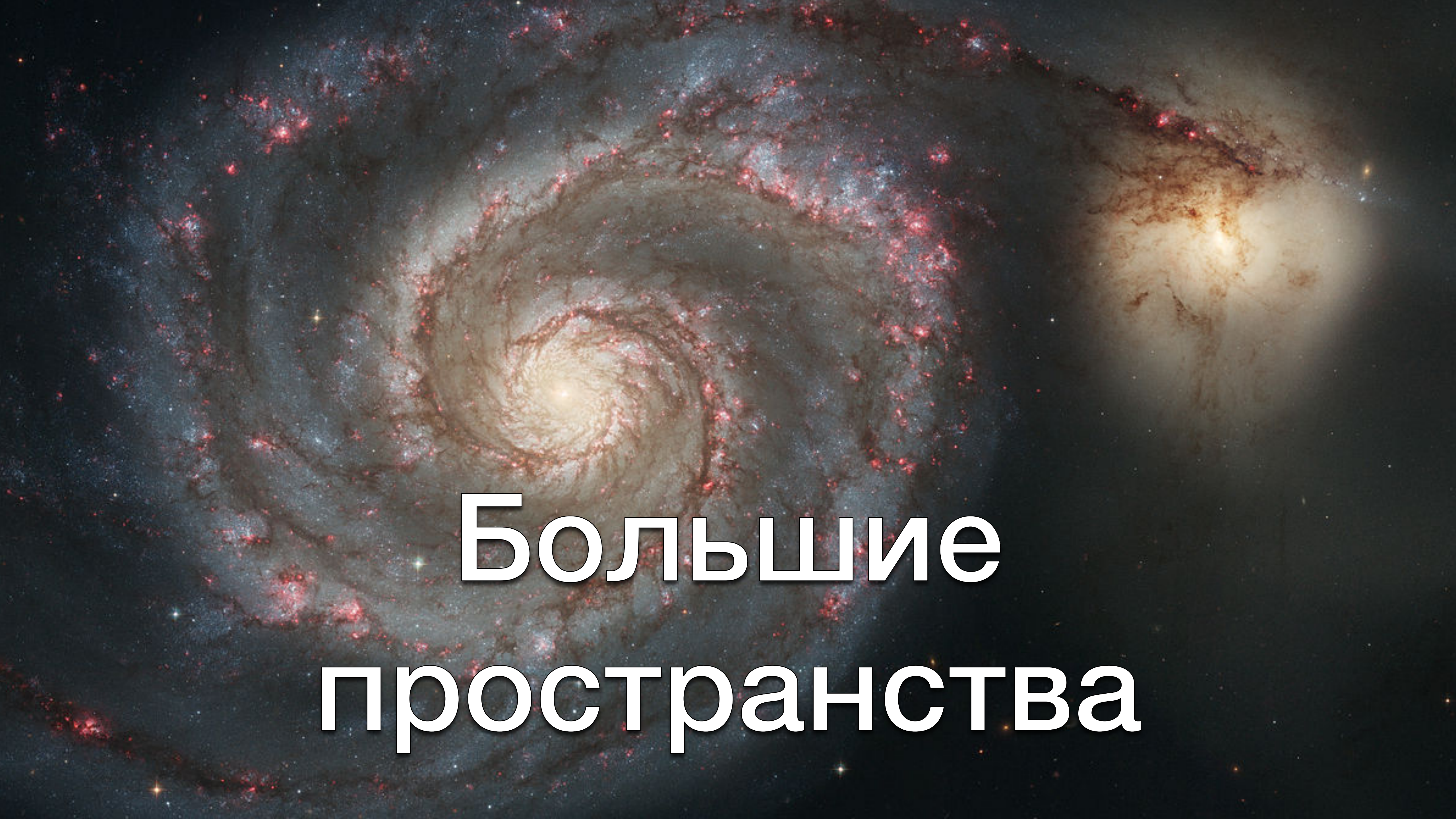
```
geometry.firstMaterial?.transparency = 0.000001
```



# Коррекция положения

- VSLAM не идеален
- Ошибки накапливаются со временем и расстоянием
- Необходима периодическая корректировка
- Большие пространства добавляют проблем





**Большие  
пространства**

# Проблема масштаба

- Ошибки позиционирования накапливаются
- Приложение не может их определить
- Карта мира AR имеет небольшой размер
- Ошибки заметнее при переходе от PoC к настоящим решениям
- Многоэтажность



# Большие пространства

- Добавляем больше маркеров
- Размещаем маркеры ненавязчиво
- Делим офис на регионы
- Используем iBeacons для выбора региона



# Ограничения AR

- Пользовательский опыт
- Точность определения положения
- Накапливающиеся ошибки
- Энергопотребление



# Ограничения можно обойти

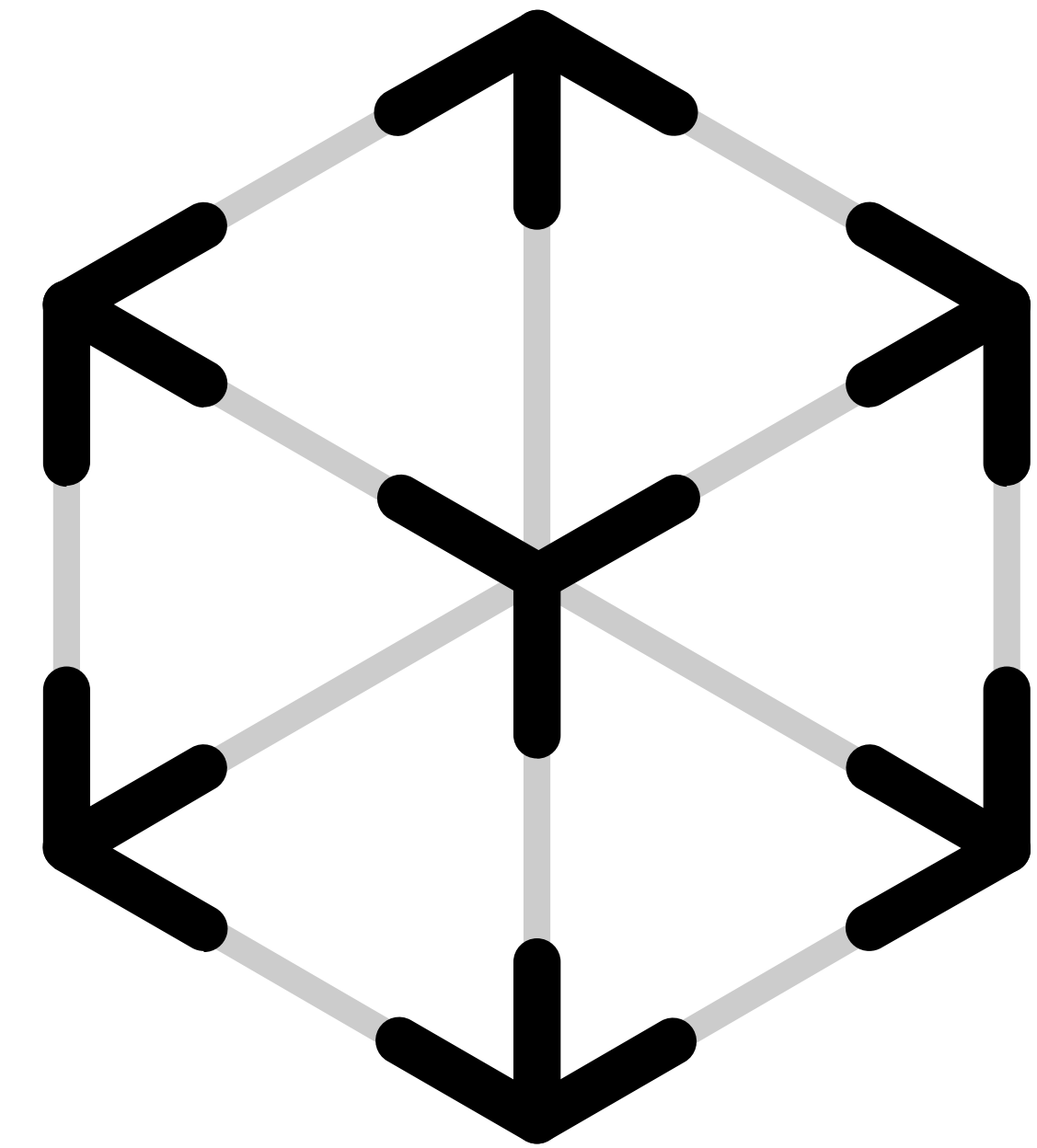
- Что-то можно обойти в коде
- Где-то будет нужно новое железо



# AR сейчас

- Приложения AR уже круты
- Приложения AR легко создавать
- Не нужно быть гейм-девелопером
- Начать можно с сайта Apple

<https://developer.apple.com/augmented-reality/>



# Будущее AR

- Новые устройства
- Новые сенсоры
- Новые применения



# Спасибо за внимание!

## Ваши вопросы?



@edeniska



@edeniska



<https://facebook.com/dtazetdinov>