



ARKit Capabilities

Michal Šindelář, Jan Čejka

Introduction

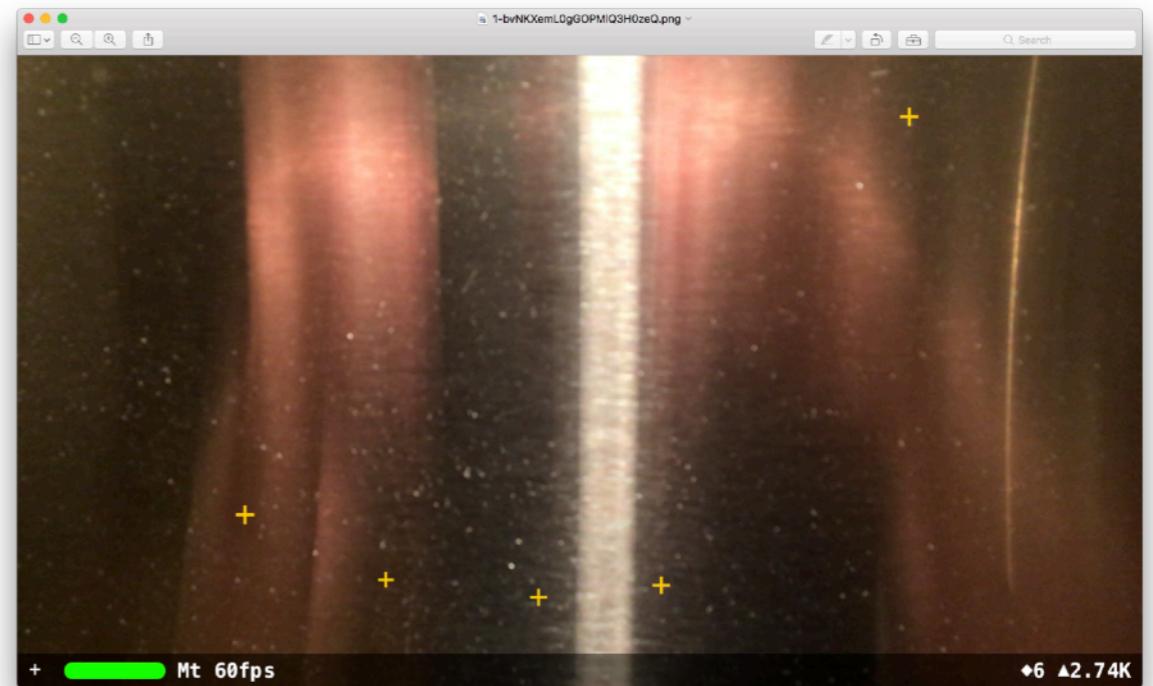
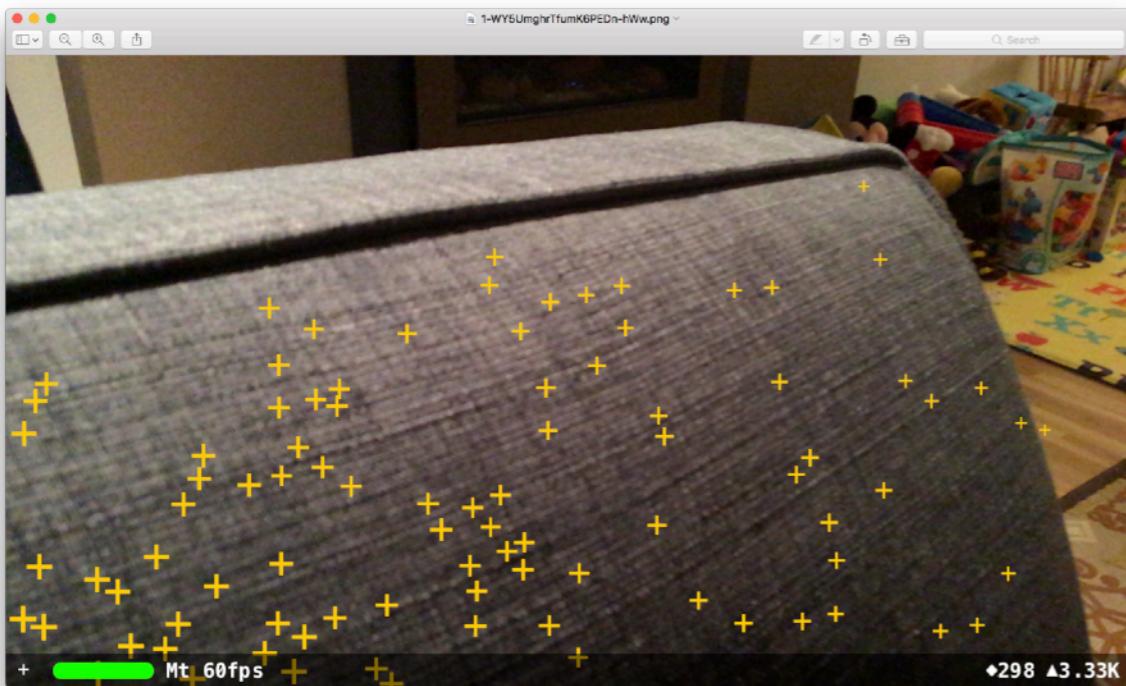
- Apple announced **ARKit** for **iOS 11**
 - Fast, stable motion tracking
 - Plane estimation
 - Ambient lighting estimation
 - Vision framework using Core ML
 - Unity, Unreal engine support



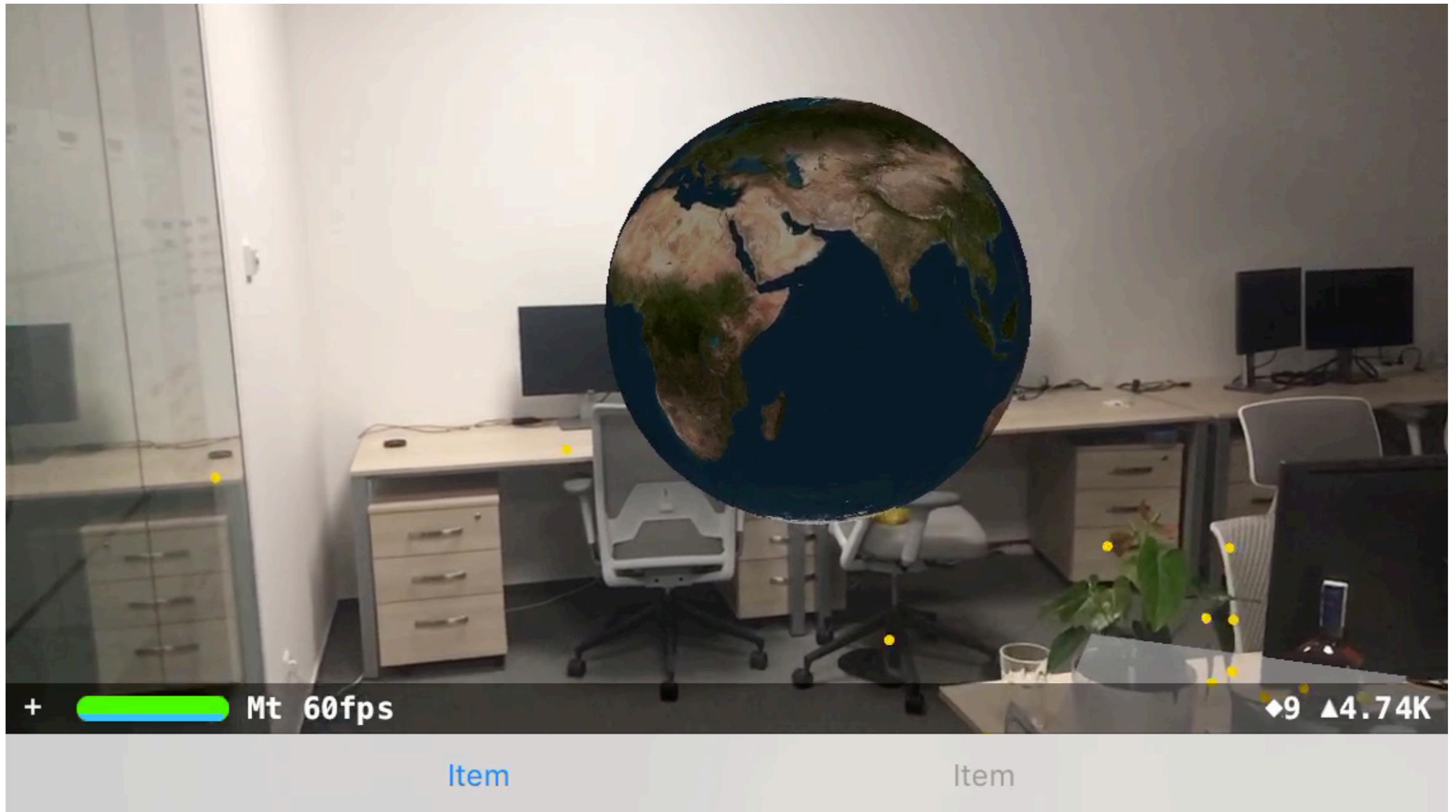
Try promised features...

Motion Tracking

- stable & realtime motion tracking
- ambient light intensity estimation (not direction)
- powered by CoreMotion

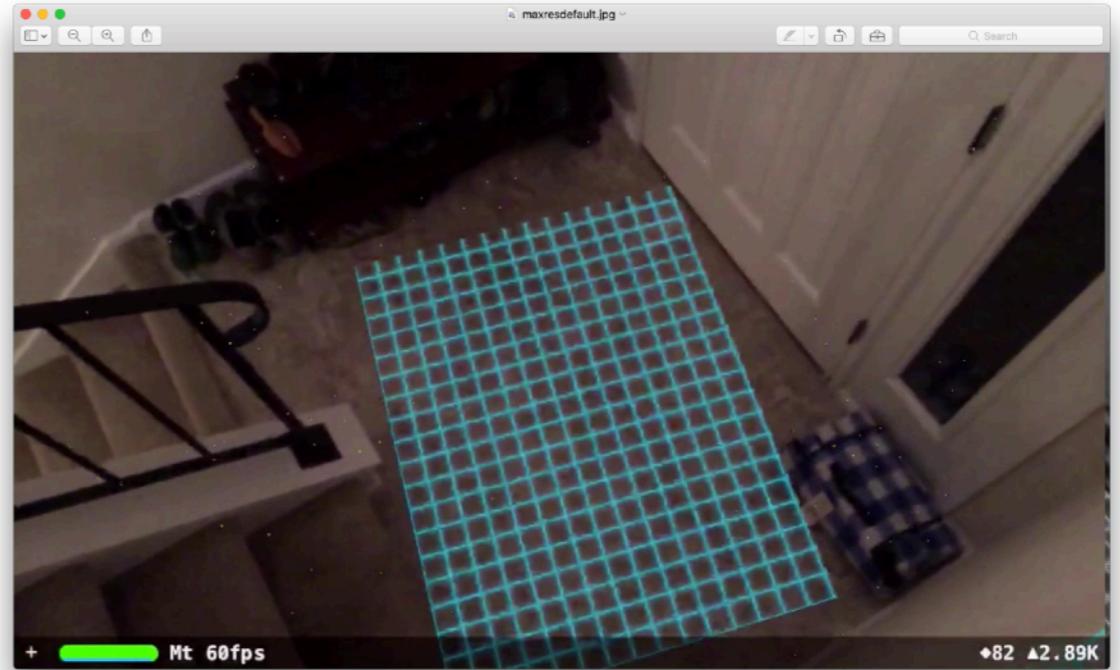
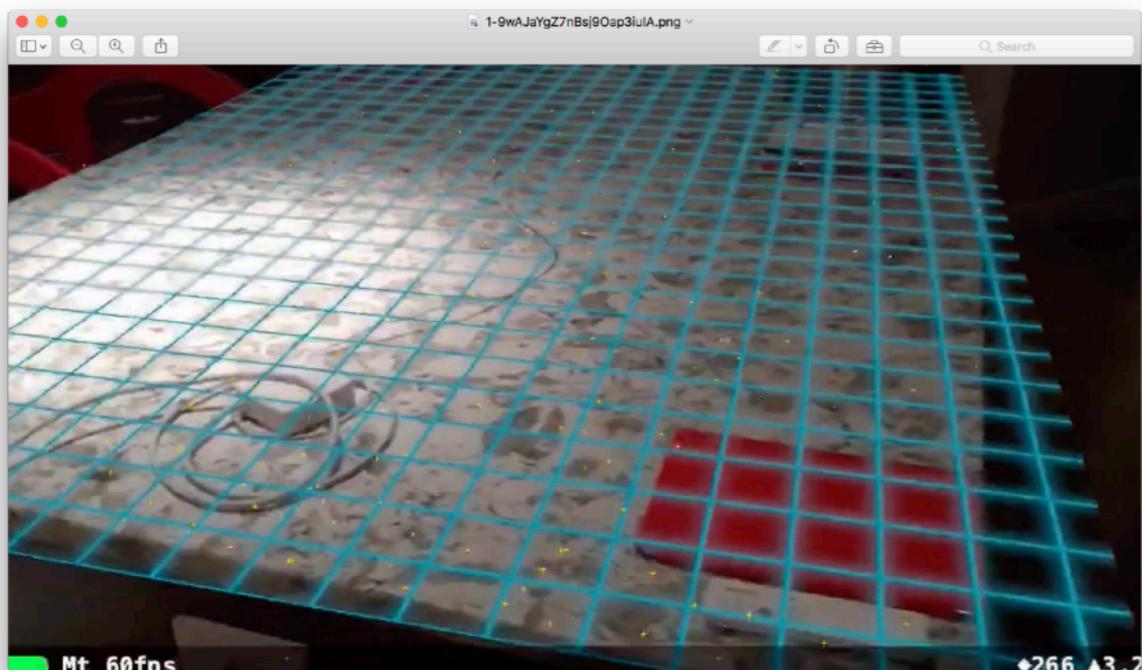


Motion Tracking

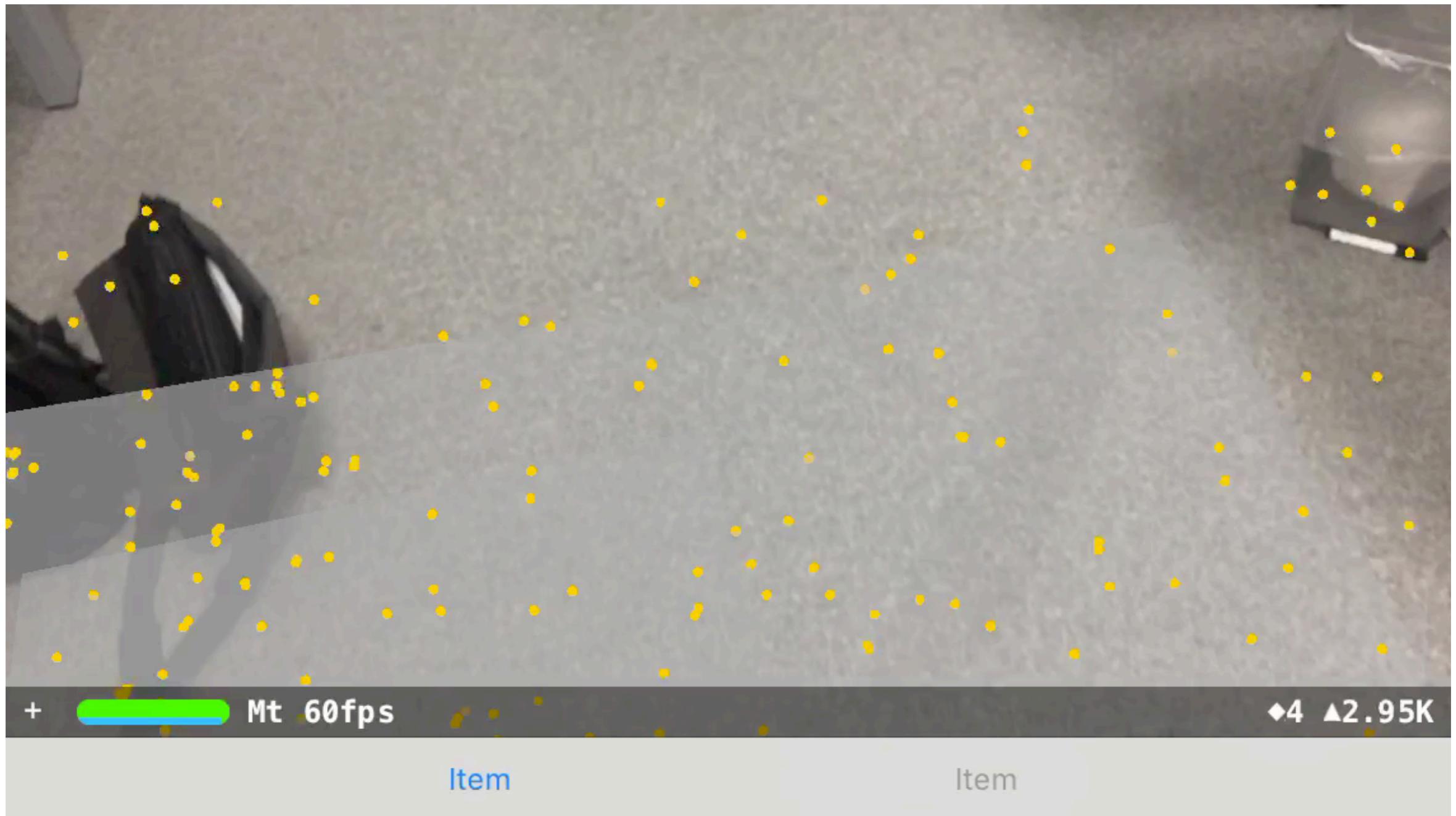


Plane estimation

- only horizontal plane detection (vertical probably coming soon)
- multiple plane detection different height
- after initialization stable in poor lighting



Plane Estimation



Vision Framework

- built on top of Core ML (Machine Learning)
 - high-performance image analysis
 - identify faces
 - detect features,
 - classify scenes
- Pretrained

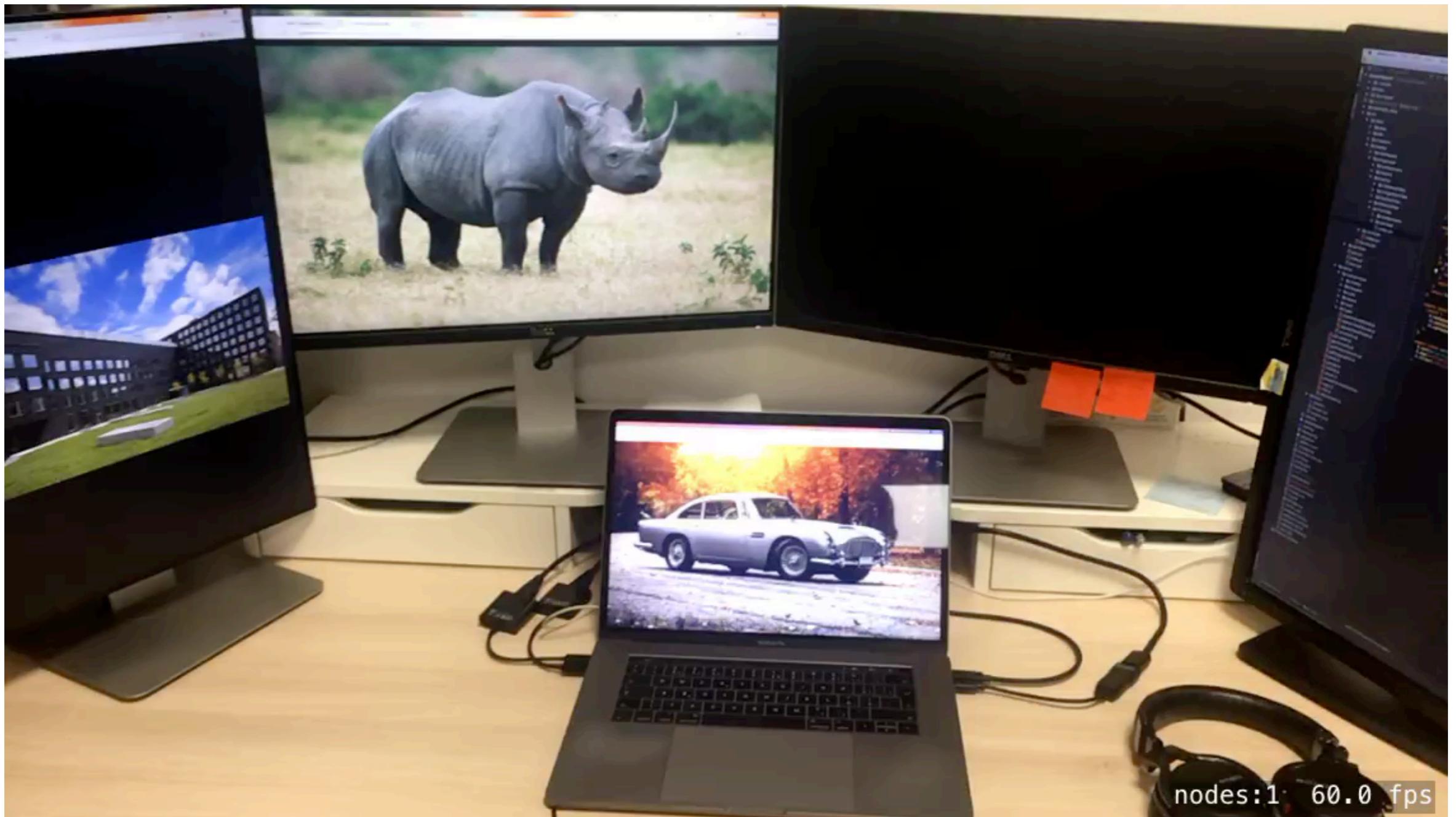


Vision Framework

- built on top of Core ML (Machine Learning)
 - high-performance image analysis
 - identify faces
 - detect features,
 - classify scenes
- trained machine learning models

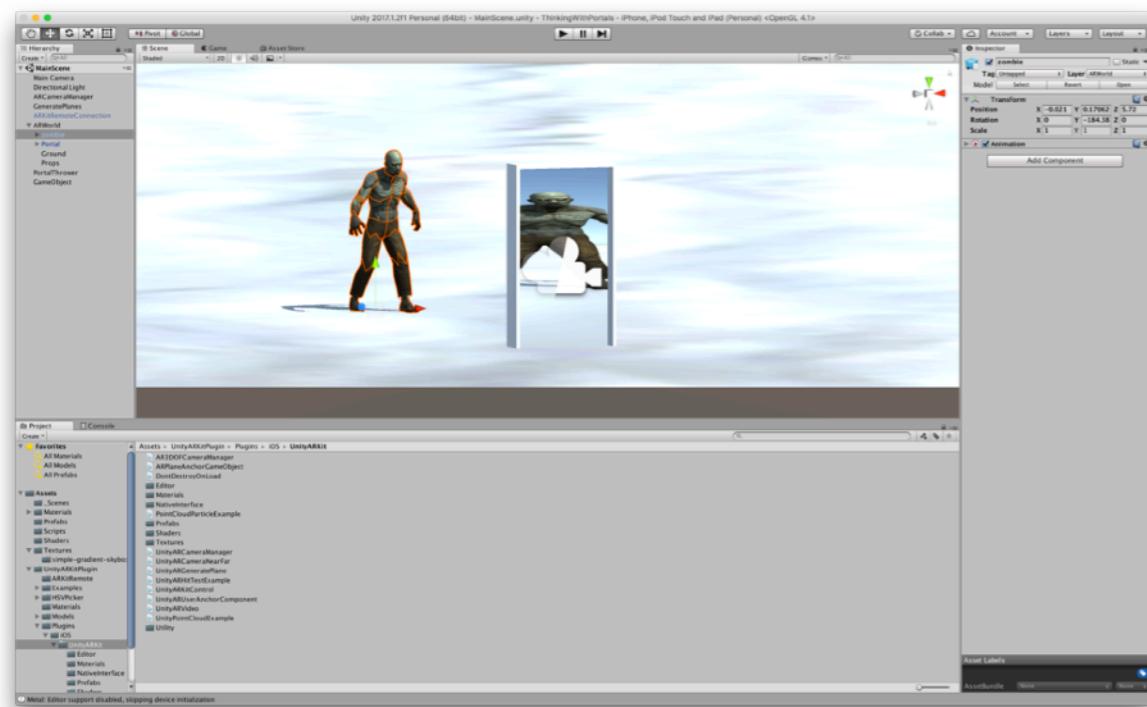


Vision Framework



Unity support

- Official ARKit middleware support
 - “Experimental native plugin that enables using all the functionality of the ARKit SDK”



```
Unity-iPhone | Build Unity-iPhone Succeeded | Yesterday at 22:18 | 43

CrashReporter.mm
=====
#import "PLCrashReporter.h"
#import "CrashReporter.h"

extern NSString* GetCrashReportsPath();
void CrashedCheckBelowForIntsMhy();

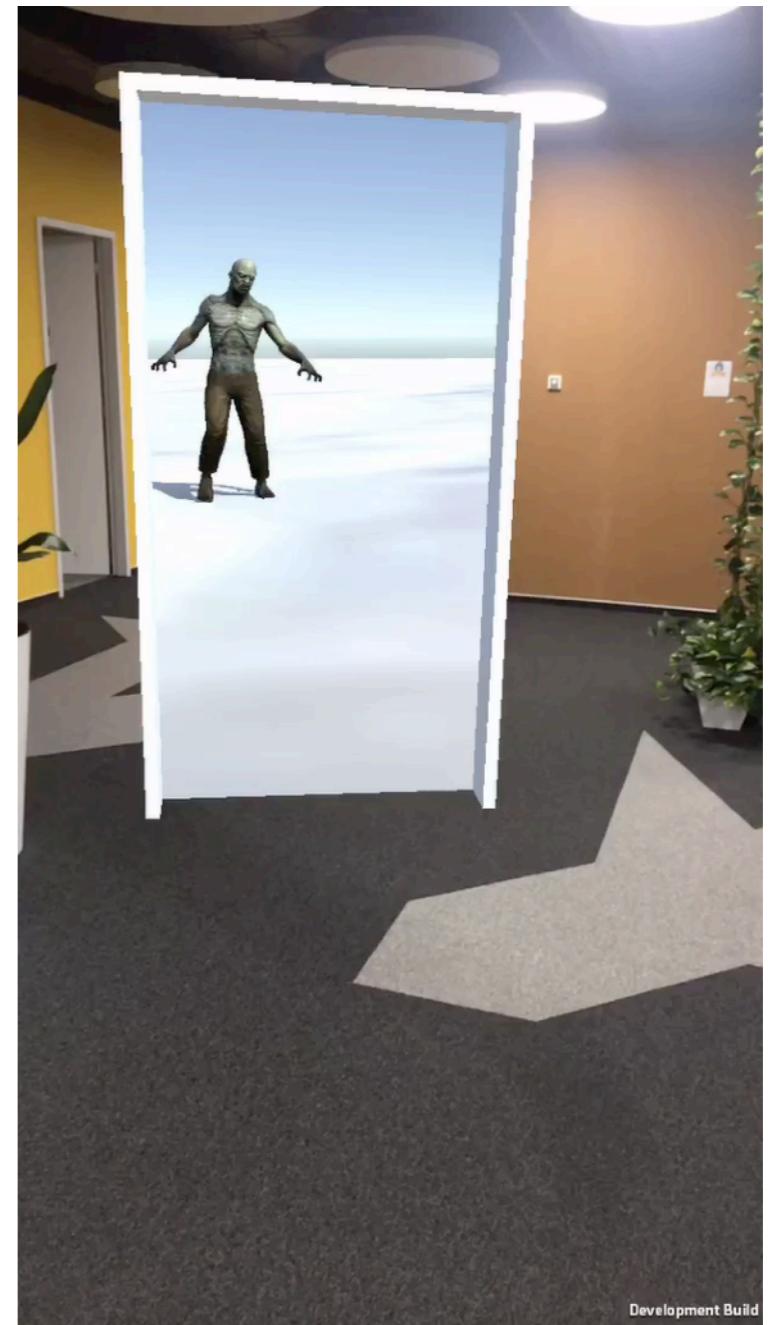
static NSUncaughtExceptionHandler* gsCrashReporterUHandler = NULL;

static void SavePendingCrashReport()
{
    if ([[UnityCrashReporter sharedReporter] hasPendingCrashReport])
        return;
    NSFileManager* fm = [NSFileManager defaultManager];
    NSError* error;
    if (![[fm createDirectoryAtPath: GetCrashReportsPath() withIntermediateDirectories: YES attributes: nil] error: &error])
        NSLog(@"CrashReporter: could not create crash report directory: %@", [error localizedDescription] UTF8String);
    else
        NSLogData *data = [[UnityPLCrashReporter sharedReporter] loadPendingCrashReportDataAndReturnError: &error];
    if (data == nil)
        NSLog(@"CrashReporter: failed to load crash report data: %@", [error localizedDescription] UTF8String);
    else
        NSString* file = [dataCrashReportsPath() stringByAppendingPathComponent: @"crash"];
        unsigned long seconds = [dataCrashReportsPath() timeIntervalSinceNow];
        file = [file stringByAppendingFormat: NSLocalizedString(@"%lu", seconds)];
        if ([file isEqualToString:@"0"]){
            if ([data writeToFile: file atomically: YES])
                NSLog(@"CrashReporter: saved pending crash report.\n");
            if ([[UnityPLCrashReporter sharedReporter] purgePendingCrashReportAndReturnError: &error])
                NSLog(@"CrashReporter: couldn't remove pending report: %@", [error localizedDescription] UTF8String);
            else
                NSLog(@"CrashReporter: couldn't save crash report.\n");
        }
}

static void InitCrashReporter()
{
    NSError* error;
    UnityInstallPostCrashCallback();
    if ([[UnityPLCrashReporter sharedReporter] enableCrashReporterAndReturnError: &error])
        NSLog(@"CrashReporter: failed to enable crash reporter: %@", [error localizedDescription] UTF8String);
}
```

Unity Portal

- Created simple portal app
- Unity scene uses
 - ARKit Plane detection





More coming soon...

Michal Šindelář, Jan Čejka