

Встраиваем Unity в мобильное приложение



Боль и Костыли



О чем доклад?

- Какие вообще есть способы показать 3D в мобилке (плюсы и минусы)
- Что за зверь такой Unity?
- Почему рекомендуемый разработчиками Unity способ встраивания - ну такое
- Костыли, мат и горящая Ж в попытках заставить это работать
- Рефлексия на тему - на кой я туда вообще полез...



Чего не будет в докладе

- Как делать игры
- Как настраивать шейдеры в Unity
- Архитектура и оптимизация Unity приложений
- Объяснение работы 3D графики и рендеринга
- Как прийти в себя после осознания количества говнокода
- Как не свалиться в депрессию из-за синдрома самозванца



Немного о себе

- В мобильной разработке с 2011 года
- Начинал как iOS разработчик
- В 2014 совместно с Олегом Чебулаевым основал Mad Brains
- С 2016 года являюсь Техническим директором компании



MAD BRAINS

- Создаем мобильные приложения и цифровые сервисы
- Основаны в 2014 году
- Занимаем 15-е место в рейтинге лучших мобильных разработчиков (Tagline)
- Более 30 отраслевых наград (лучшее приложение)
- Более 200 успешно реализованных проектов
- Более 70 человек в команде
- Используем самые современные подходы и технологии



Задача

“Есть нативные мобильные приложения **iOS** и **Android**. Нужно в них на части экрана показать красивую **3D модельку**. Так чтоб крутилась и так **вжух** прозрачные штуки из нее росли, потом тыкаем тут так **хобана** и цвет меняется. А еще чтоб все динамически строилось и **анимашка** красивая была”

Типичный заказчик



Я спросил у ясеня chatGPT

Варианты работы с 3D в мобилке

- Нативные (SceneKit for iOS and “ARCore” for Android)
- OpenGL и прочий низкий уровень
- Игровые движки “высокого” уровня
 - Unity
 - Unreal Engine
 - Cocos2d-x
 - etc
- WebGL

SceneKit

```
override func viewDidLoad() {  
    ...  
    let light = SCNLight()  
    light.type = SCNLightTypeSpot  
    light.spotInnerAngle = 30.0  
    light.spotOuterAngle = 80.0  
    light.castsShadow = true  
    let lightNode = SCNNode()  
    lightNode.light = light  
    lightNode.position = SCNVector3(x: 1.5, y: 1.5, z: 1.5)  
    ...  
    let constraint = SCNLookAtConstraint(target: cubeNode)  
    constraint.gimbalLockEnabled = true  
    cameraNode.constraints = [constraint]  
    lightNode.constraints = [constraint]  
    ...  
}
```

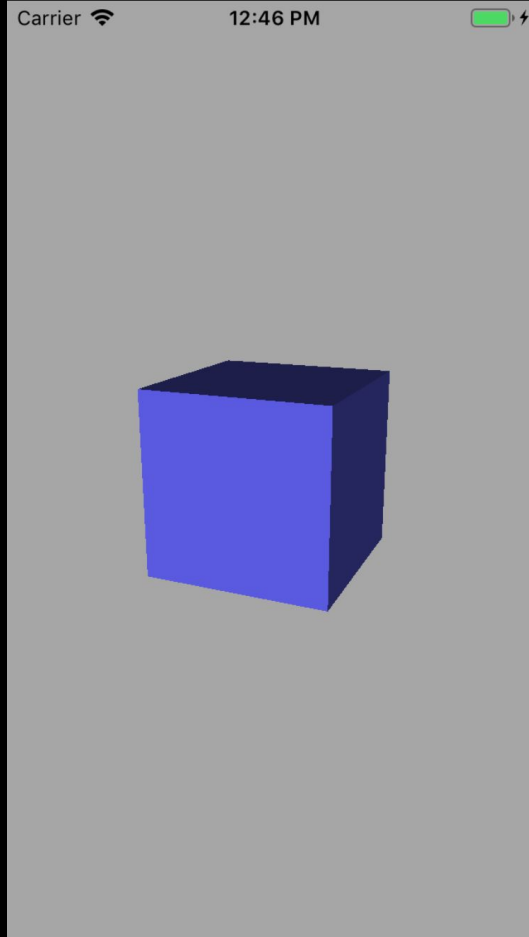
ARCore (SceneViewer)

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    mSceneView = findViewById(R.id.sceneView);  
    createScene();  
}  
  
private void createScene() {  
    mScene = new Scene(mSceneView);  
    ModelRenderer.builder()  
        .setSource(this, Uri.parse("file:///android_asset/model.sfb"))  
        .build()  
        .thenAccept(renderable->onRenderableLoaded(renderable))  
        .exceptionally(throwable -> {  
            Log.i("Sceneform", "failed to load model");  
            return null;  
        });  
}  
  
private void onRenderableLoaded(Renderable renderable) {  
    Node cakeNode = new Node();  
    cakeNode.setRenderable(renderable);  
    cakeNode.setParent(mScene);  
    cakeNode.setLocalPosition(new Vector3(0f, 0f, -1f));  
    mScene.addChild(cakeNode);  
}
```



OpenGL (Metal, Vulkan)

```
public func render() {  
  
    update()  
  
    glClear(GLbitfield(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT))  
    glClearColor(0.65, 0.65, 0.65, 1.0)  
  
    // Use shader  
    shader.use()  
  
    withUnsafePointer(to: &modelViewProjectionMatrix, {  
        $0.withMemoryRebound(to: Float.self, capacity: 16, {  
            glUniformMatrix4fv(shader.uniforms[UNIFORM_MODELVIEWPROJECTION_MATRIX], 1, 0, $0)  
        })  
    })  
  
    withUnsafePointer(to: &normalMatrix, {  
        $0.withMemoryRebound(to: Float.self, capacity: 9, {  
            glUniformMatrix3fv(shader.uniforms[UNIFORM_NORMAL_MATRIX], 1, 0, $0)  
        })  
    })  
  
    // Draw objects  
    for g in geometries {  
        g.draw()  
    }  
}
```



M WebGL

```
const textureLoader = new THREE.TextureLoader();

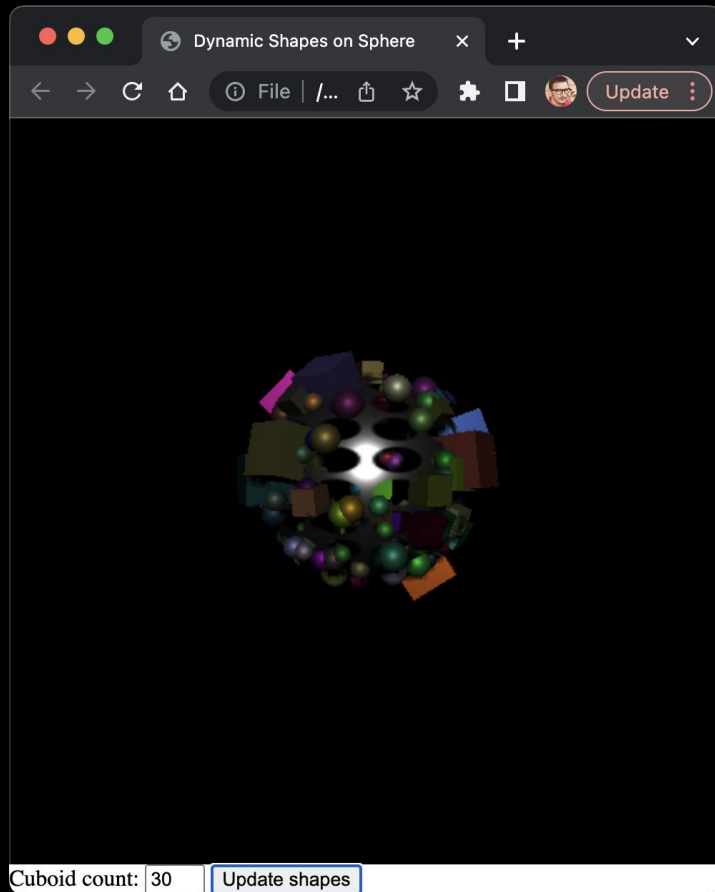
textureLoader.load(
  'path/to/your/texture.png',
  function (texture) {
    // Create the material with the loaded texture
    const material = new THREE.MeshBasicMaterial({ map: texture, transparent: true });

    // Create the sphere with the new material
    const geometry = new THREE.SphereGeometry(1, 32, 32);
    const sphere = new THREE.Mesh(geometry, material);
    scene.add(sphere);

    // Call the updateShapes function
    updateShapes();

    // Move the camera
    camera.position.z = 5;

    // Call the animate function
    animate();
  },
  undefined,
  function (error) {
    console.error('An error occurred while loading the texture:', error);
  }
);
```

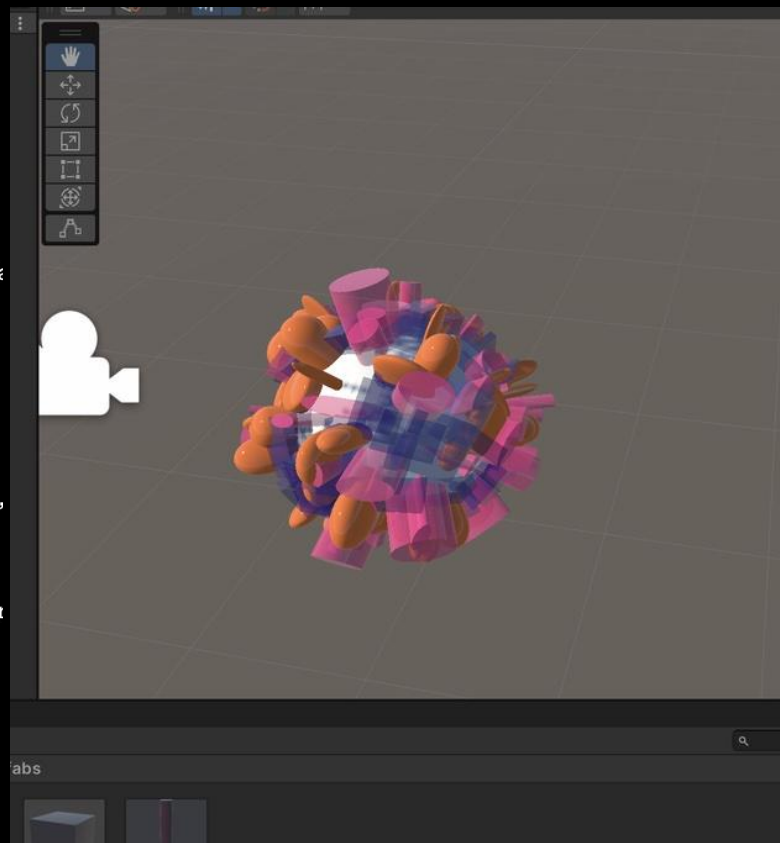




Unity

```
private void GenerateShape(GameObject shapePrefab, int count, Vector2 sizeRange)
{
    for (int i = 0; i < count; i++)
    {
        GameObject shape = Instantiate(shapePrefab);
        shape.transform.SetParent(transform);
        shape.transform.localScale = Vector3.one * Random.Range(sizeRange.x, sizeRange.y);

        Vector3 randomDirection = Random.onUnitSphere;
        shape.transform.position = transform.position + randomDirection * transform.localScale.x;
        shape.transform.up = randomDirection;
    }
}
```



M * filament

Filament is a real-time physically-based renderer written in C++. It is mobile-first, but also multi-platform

```
// init must be called before we can build any materials.
filamat::MaterialBuilder::init();

// Compile a custom material to use on the triangle.
filamat::Package pkg = filamat::MaterialBuilder()
    // The material name, only used for debugging purposes.
    .name("Triangle material")
    // Use the unlit shading mode
    .shading(filamat::MaterialBuilder::Shading::UNLIT)
    // Expose the COLOR attribute visible to our shader code.
    .require(VertexAttribute::COLOR)
    // Custom GLSL fragment shader
    .material("void material (inout MaterialInputs material) {"
        "    prepareMaterial(material);"
        "    material.baseColor = getColor();"
        "}")
    // Compile for Metal on mobile platforms.
    .targetApi(filamat::MaterialBuilder::TargetApi::METAL)
    .platform(filamat::MaterialBuilder::Platform::MOBILE)
    .build();
assert(pkg.isValid());

// shutdown should be called after all materials are built.
filamat::MaterialBuilder::shutdown();
```

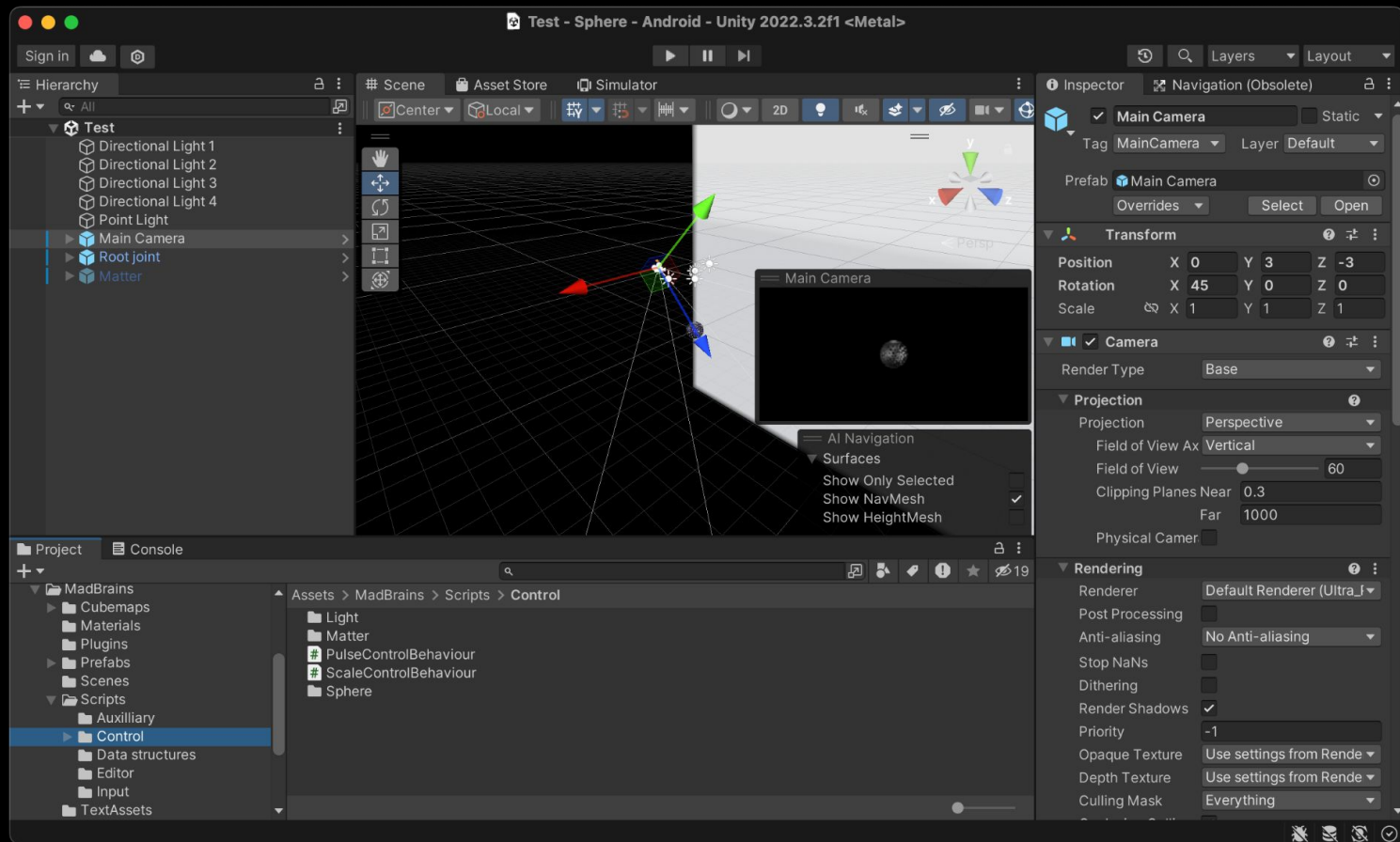


<https://google.github.io/filament/webgl/helmet.html>



Unity! Немного истории

- 2002: Дэвид Хелгасон и Николас Фрэнсис разрабатывали движок для себя. Цель - движок для мобильных устройств
- 2005: Представили на WDC но только для Mac OS X
- 2009: Версия 2.5 с поддержкой iPhone и iPod Touch
- 2010: Unity 3 поддержка Android
- 2015: Unity 5 много улучшательств и WebGL
- ...





Плюсы и минусы Unity

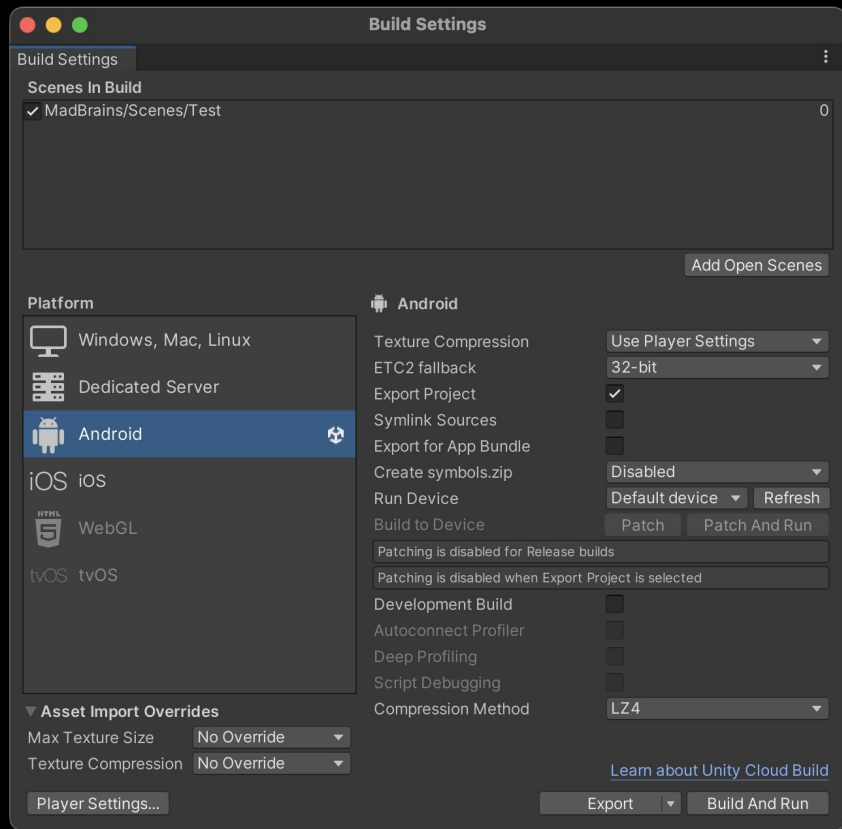
- + Кроссплатформенность
- + Доступность и легкость входа
- + Развитые инструменты
- + Поддержка сообщества
- + Производительность
- + “Зрелость”
- Проприетарность (не Open Source)
- Подписная модель
- Ограничения
- Отсутствие полного контроля



Компиляция проекта

Из коробки доступно
много всего

Дальше будем только про
мобилки (iOS и Android)





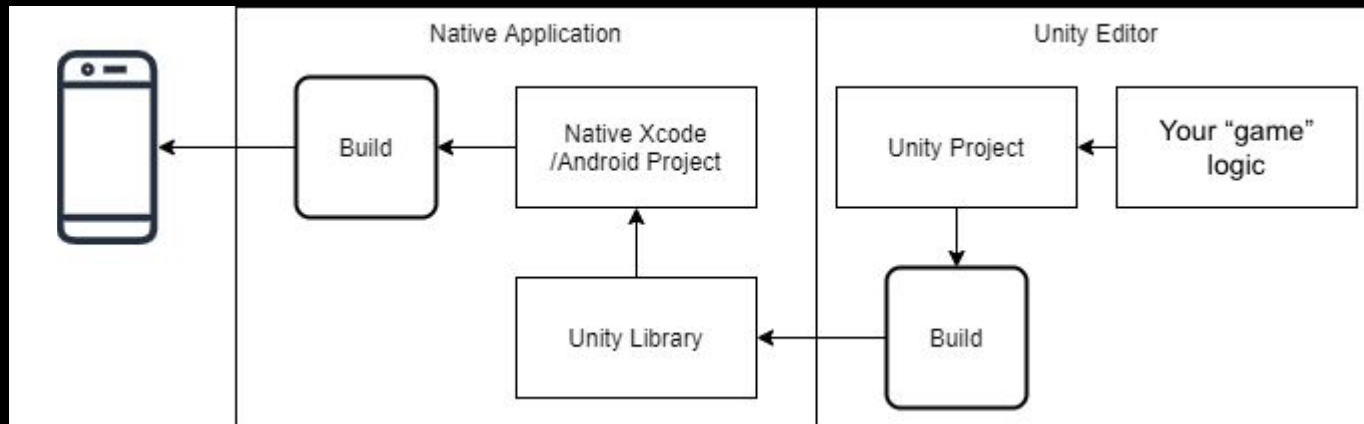
Unity as a Library

Появилась в Unity 2019.3

How it works

Unity offers controls that let you manage when and how to load/activate/unload the runtime library within the native application. Beyond that, the mobile app build process mostly stays the same. Unity lets you create iOS Xcode and Android Gradle projects.

<https://unity.com/features/unity-as-a-library>





Ограничения

Limitations

While we tested many scenarios for Unity as a library hosted by a native app, Unity does not control anymore the lifecycle of the runtime, so we cannot guarantee it'll work in all possible use cases. For example:

- Unity as a Library supports rendering only full screen, rendering on a part of the screen isn't supported.
- Loading more than one instance of the Unity runtime isn't supported.
- You may need to adapt 3rd party Plug-ins (native or managed) to work properly
- Overhead of having Unity in unloaded state is: 90Mb for Android and 110Mb for iOS



Ограничения

Limitations

While we tested many scenarios for Unity as a library hosted by a native app, Unity does not control anymore the lifecycle of the runtime, so we cannot guarantee it'll work in all possible use cases. For example:

- Unity as a Library supports rendering only full screen, rendering on a part of the screen isn't supported.
- Loading more than one instance of the Unity runtime isn't supported.
- You may need to adapt 3rd party Plug-ins (native or managed) to work properly
- Overhead of having Unity in unloaded state is: 90Mb for Android and 110Mb for iOS



Ограничения

Limitations

While we tested many scenarios for Unity as a native app, Unity does not control anymore the lifecycle of the runtime, so we can't guarantee it'll work in all possible use cases. For example:

- Unity as a Library supports rendering only full screen, rendering a partial screen isn't supported.
- Loading more than one instance of the Unity runtime isn't supported.
- You may need to adapt 3rd party Plug-ins (native or managed) to work properly
- Overhead of having Unity in unloaded state is: 90Mb for Android and 110Mb for iOS

Ограничения

Limitations

While we tested many scenarios, Unity does not control anymore the lifecycle of the app in all possible use cases. For example:

- Unity as a Library supporting multi-screen isn't supported
- Loading more than one instance of the app isn't supported.
- You may need to adapt 3rd party libraries (or managed) to work properly
- Overhead of having Unity as a native app, Unity does not guarantee it'll work in all possible use cases. For example:

и экрана показать

б кривтипась и т:

Android and 110Mb for iOS

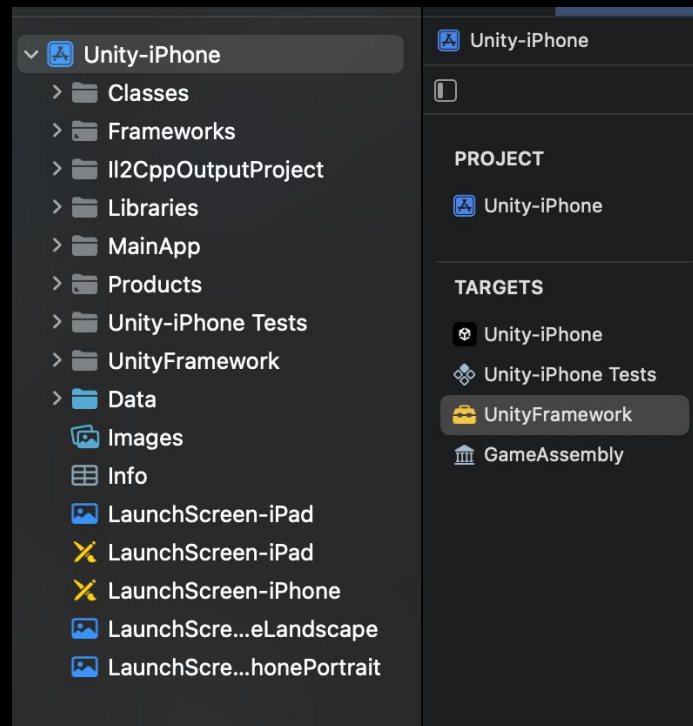
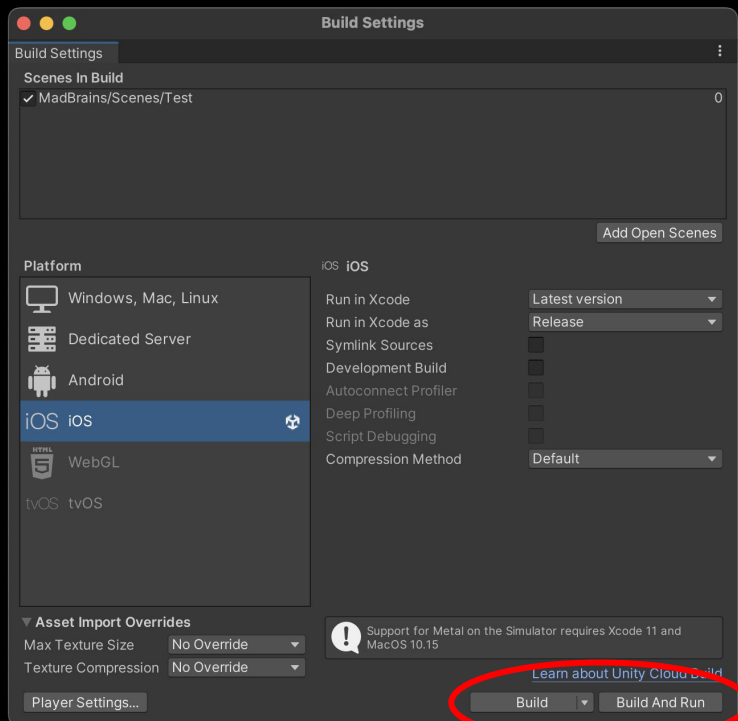


“as a Library”





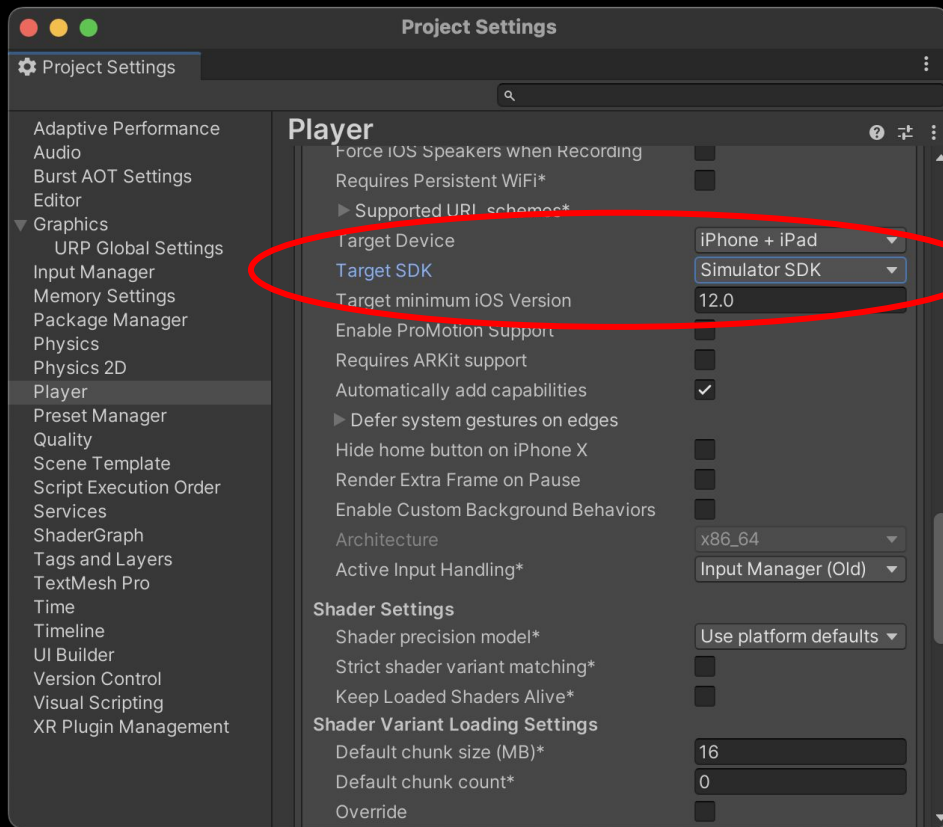
ios

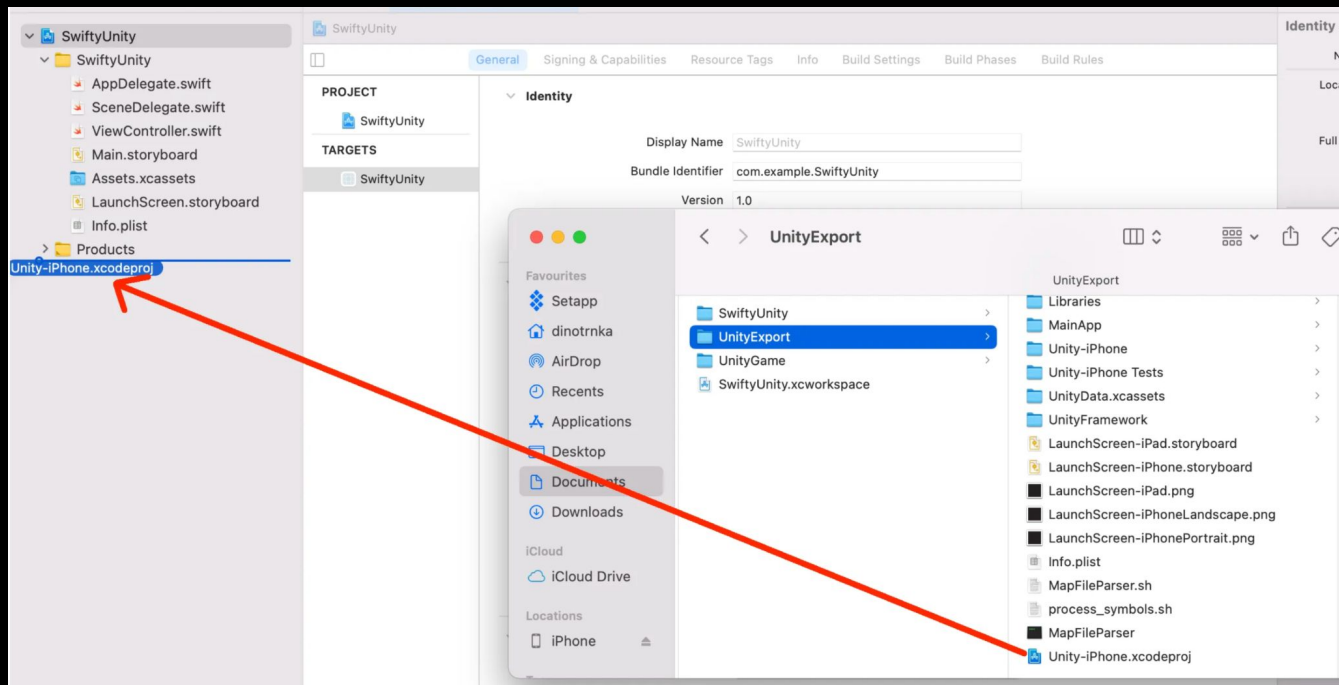




ios

Экспорты для Симулятора и
Устройства в Unity делаются
отдельно!





И еще пачка настроек зависимостей и путей

Ah shit, here we go again.



iOS - а либа то не настоящая!

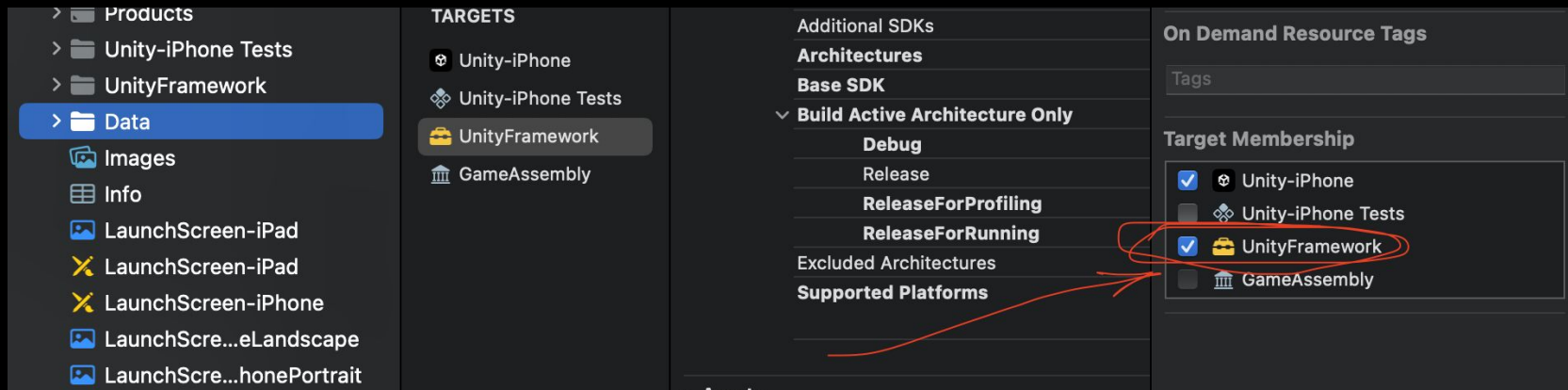
Библиотека не сконфигурирована!

- Папка Data прикреплена к приложению, а не к библиотеке
- На выходе *.framework, а не *.xcframework





iOS - донастройка



M iOS — а как либу-то загрузить?

```
private func loadUnityFramework() -> UnityFramework? {  
    let bundlePath: String = Bundle.main.bundlePath + frameworkPath  
  
    let bundle = Bundle(path: bundlePath)  
    if bundle?.isLoaded == false {  
        bundle?.load()  
    }  
  
    let ufw = bundle?.principalClass?.getInstance()  
    if ufw?.appController() == nil {  
        let machineHeader = UnsafeMutablePointer<MachHeader>.allocate(capacity: 1)  
        machineHeader.pointee = _mh_execute_header  
  
        ufw?.setExecuteHeader(machineHeader)  
    }  
    return ufw  
}
```



M iOS - finally!

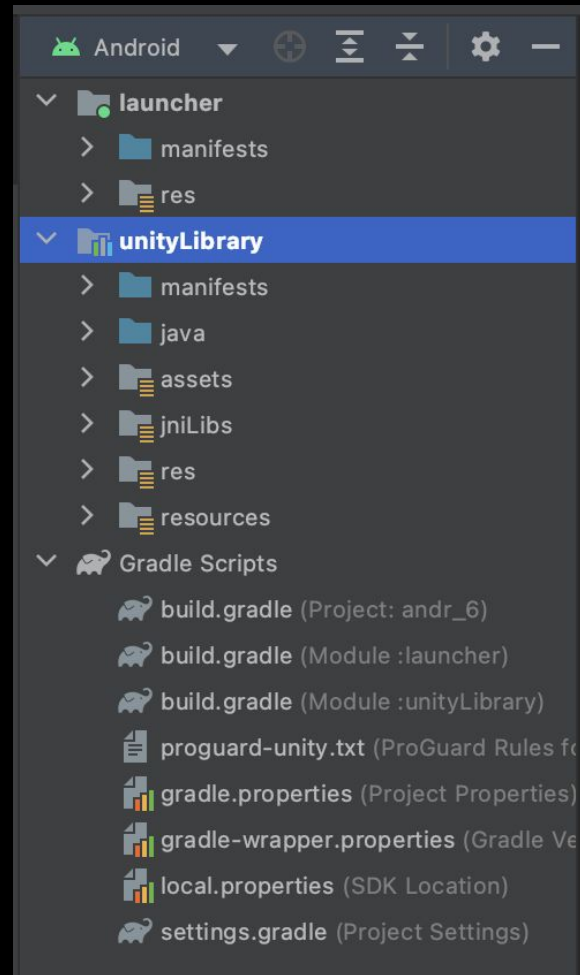
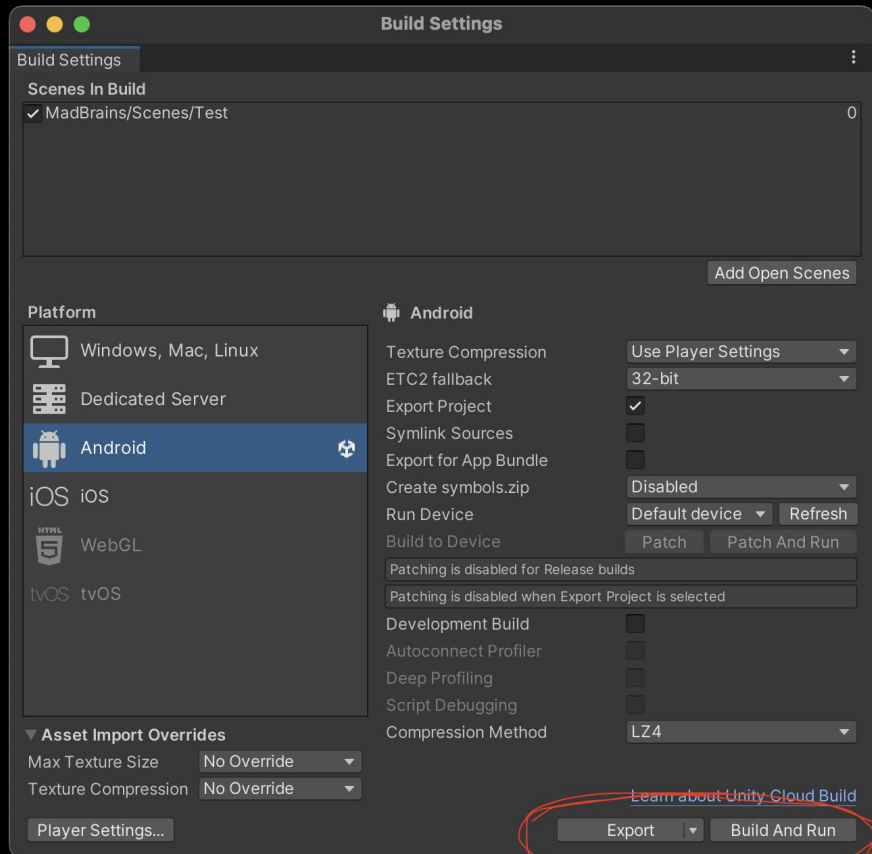
```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {  
    Unity.shared.setHostMainWindow(window)  
    //...
```

```
@IBAction func btnOpen(_ sender: UIButton) {  
    Unity.shared.show()  
}
```

Open Unity

MADE
WITH  Unity

M Android





Android



```
local.properties × settings.gradle (NativeAndroidApp) × build.gradle (:app) × build.gradle (NativeAndroidApp) ×  
gradle project sync failed. Basic functionality (e.g. editing, debugging) will not ... Try Again Open 'Build' View Sh  
include ':app'  
include ':unityLibrary'  
project(':unityLibrary').projectDir = new File('..\UnityProject\\androidBuild\\unityLibrary')
```

И еще пачка настроек зависимостей и путей



Android - нюанс номер 1

Версии зависимостей (пути и совместимость версий)

- Java Development Kit (JDK)
- Software Development Kit (SDK)
- Native Development Kit (NDK)
- Gradle
- Gradle Wrapper
- Android Gradle Plugin
- `com.android.library` Plugin
- `org.jetbrains.kotlin.android` Plugin



Minimum supported Gradle version is 7.2 -

is 7.2.

Android

java.lang.NoClassDefFoundError: Could not initialize class
org.codehaus.groovy.vmplugin.v7.Java7

1.8.10.jar!/META-INF/kotlin-stdlib.kotlin_module: Module was compiled with an incompatible
version of Kotlin. The binary version of its metadata is 1.8.0, expected version is 1.6.0.

Could not find

Development Kit (SDK)

Failed to install the following Android SDK packages as some licences have not been
accepted.

platforms;android-33 Android SDK Platform 33

build-tools;30.0.3 Android SDK Build-Tools 30.0.3

To build this project, accept the SDK license agreements and install the missing
using the Android Studio SDK Manager.

org.jetbrains.kotlin.android Plugin

- org.jetbrains.kotlin.android Plugin

Отдельная история





Android зависимости

JDK 17
Gradle 7.3.3
AGP 7.2.2

JDK 1.8
Gradle 6.1.1
AGP 4.0.1

JDK 11.0.1
Gradle 7.0.2
AGP 7.0.1





Android - нюанс номер 2

```
I/IL2CPP (31563): JNI_OnLoad
E/_flutter_widge(31563): Invalid ID 0x00000000.
D/AndroidRuntime(31563): Shutting down VM
E/AndroidRuntime(31563): FATAL EXCEPTION: main
E/AndroidRuntime(31563): Process: com.example.unity_project, PID: 31563
E/AndroidRuntime(31563): android.content.res.Resources$NotFoundException: String resource ID #0x0
E/AndroidRuntime(31563):     at android.content.res.Resources.getText(Resources.java:367)
E/AndroidRuntime(31563):     at android.content.res.Resources.getString(Resources.java:460)
E/AndroidRuntime(31563):     at com.unity3d.player.UnityPlayer.GetGlViewContentDescription(Unknown Source:20)
E/AndroidRuntime(31563):     at com.unity3d.player.UnityPlayer.<init>(Unknown Source:271)
E/AndroidRuntime(31563):     at android.os.Handler.handleCallback(Handler.java:883)
E/AndroidRuntime(31563):     at android.os.Handler.dispatchMessage(Handler.java:100)
E/AndroidRuntime(31563):     at android.os.Looper.loop(Looper.java:214)
E/AndroidRuntime(31563):     at android.app.ActivityThread.main(ActivityThread.java:7697)
E/AndroidRuntime(31563):     at java.lang.reflect.Method.invoke(Native Method)
E/AndroidRuntime(31563):     at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:516)
E/AndroidRuntime(31563):     at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:950)
D/ViewRootImpl[MainActivity](31563): windowFocusChanged hasFocus=false inTouchMode=true
I/Process (31563): Sending signal. PID: 31563 SIG: 9
Lost connection to device.
Exited (sigterm)
```



Android - нюанс номер 2

Библиотека не сконфигурирована!

- Добавить параметр в AndroidManifest.xml
- Добавить файл strings.xml в ресурсы с “магической” строкой





Android - донастройка

The screenshot displays an IDE interface with two files open. The top file is `AndroidManifest.xml`, showing the following XML code:

```
<meta-data android:name="notch.config" android:value="portrait|landscape" />
<meta-data android:name="unity.auto-report-fully-drawn" android:value="true" />
<activity android:name="com.unity3d.player.UnityPlayerActivity" android:exported="true" android:
    <intent-filter>
        <category android:name="android.intent.category.LAUNCHER" />
```

The bottom file is `strings.xml`, showing the following XML code:

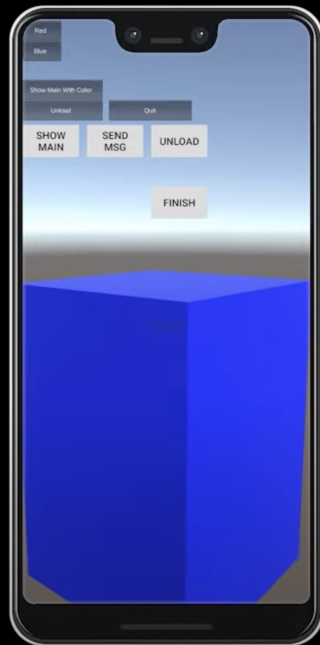
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="game_view_content_description">Game view</string>
</resources>
```

The left sidebar shows the project structure:

- unityLibrary
 - src
 - main
 - res
 - values
 - freeformwindow.xml
 - ids.xml
 - strings.xml (selected)
 - styles.xml

M Android - finally!

```
val intent = Intent(this, UnityActivity::class.java)  
startActivity(intent)
```





Чем плох такой подход?

Представьте что вы приходите в проект. Он нативный на iOS или Android, но чтобы его скомпилировать вам нужно:

- Скачать еще одну папку с другим проектом (или они уже рядом лежат)
- Установить Unity
- Настроить все окружение (особенно на Android)





А хочется - “as a Library”





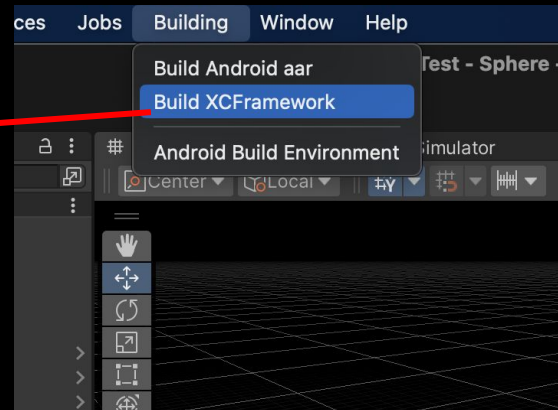
Unity Scripting API

```
public class ProjectBuilder: EditorWindow
{
    private static string xcframeworkPath = EditorPrefs.GetString("xcframeworkPath");

    [MenuItem("Building/Build XCFramework")]
    public static void BuildXCFramework()
    {
        if (EditorUserBuildSettings.activeBuildTarget != BuildTarget.iOS)
        {
            EditorUserBuildSettings.SwitchActiveBuildTarget(BuildTargetGroup.iOS, BuildTarget.iOS);
            UnityEngine.Debug.Log("Target platform is switched to iOS");
        }
        xcframeworkPath = EditorUtility.SaveFilePanel(
            "Select save folder",
            directory,
            defaultName,
            "xcframework");

        // Create temporary "builds" folder both for device and simulator
        string dataPath = Directory.GetParent(Application.dataPath).FullName;
        var tempBuildsPath = Path.Combine(dataPath, "Temp", "xcodeBuilds");
        string DeviceBuildPath = Path.Combine(tempBuildsPath, "DeviceBuild");
        string SimulatorBuildPath = Path.Combine(tempBuildsPath, "SimulatorBuild");

        BuildProject(SimulatorBuildPath, iOSSdkVersion.SimulatorSDK);
    }
}
```





Android - что делается в скриптах

- 1) Переключение и экспорт Android проекта
- 2) Настраивается окружение (Java Home, Gradle Path и тп)
- 3) Полностью заменяются или добавляются файлы
 - a) AndroidManifest.xml
 - b) build.gradle (рутовый)
 - c) build.gradle (библиотечного модуля)
 - d) gradle.properties
 - e) strings.xml
- 4) Компилируется unityLib.aar



iOS - что делается в скриптах

Процесс запускается дважды, сначала для Симулятора, потом для Устройства:

- 1) Переключение и экспорт iOS проекта
- 2) Проверяется что это macOS и есть Xcode
- 3) Полностью заменяются или добавляются файлы
 - main.mm
 - Preprocessor.h
 - UnityAppController.mm
 - UnityAppController+ViewHandling.mm
 - UnityFramework.h
 - UnityView.h
 - UnityView.mm
 - UnityView+iOS.mm
 - UnityView+Private.h
 - UnityViewControllerBase+iOS.mm
- 4) Компилируется *.framework

Финальный этап:

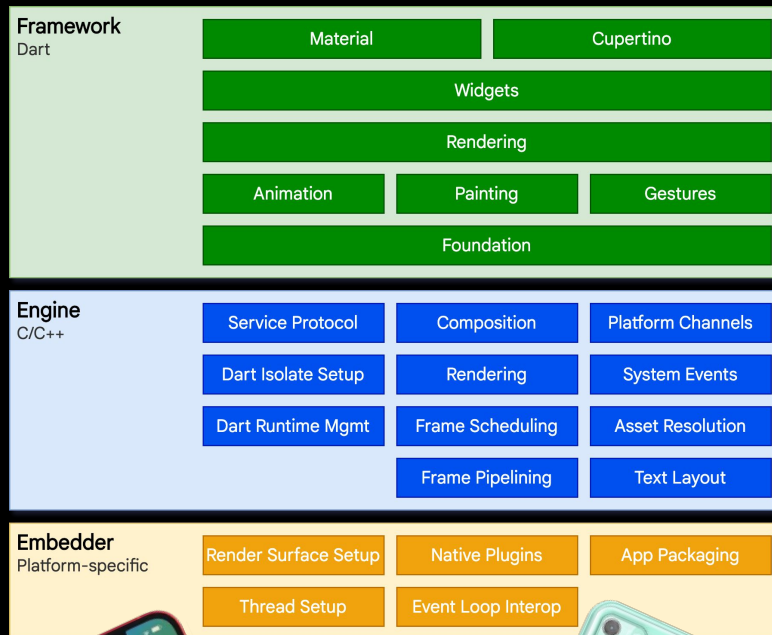
- два *.framework-а собираются в один unityLib.xcframework



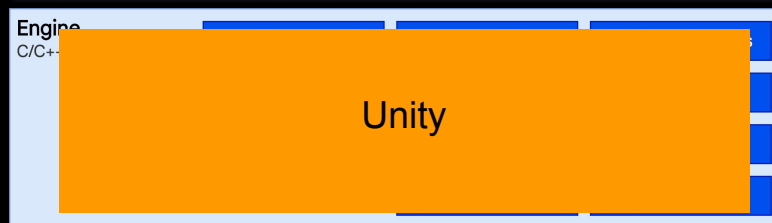
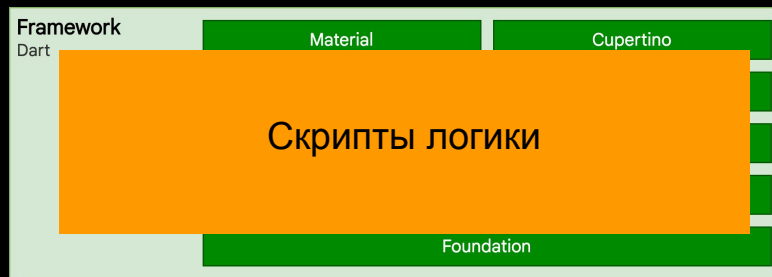
Но что по поводу требования?

...привлечения
них на части экрана показать
Так чтоб кру...

M Архитектура (аналогия)



Архитектура (аналогия)



M Android оболочка

```
15 public class UnityPlayerActivity extends Activity implements IUnityPlayerLifecycleEvents
16 {
17     18 usages
18     protected UnityPlayer mUnityPlayer; // don't change the name of this variable; referenced f
19
20     // Override this in your custom UnityPlayerActivity to tweak the command line arguments pas
21     // The command line arguments are passed as a string, separated by spaces
22     // UnityPlayerActivity calls this from 'onCreate'
23     // Supported: -force-gles20, -force-gles30, -force-gles31, -force-gles31aep, -force-gles32,
24     // See https://docs.unity3d.com/Manual/CommandLineArguments.html
25     // @param cmdLine the current command line arguments, may be null
26     // @return the modified command line string or null
27     1 usage
28     protected String updateUnityCommandLineArguments(String cmdLine) { return cmdLine; }
29
30
31     // Setup activity layout
32     1 usage
33     @Override protected void onCreate(Bundle savedInstanceState)
34     {
```

M Но это ж JVM!

unityLibrary.aar

unknown (Version Name: unknown, Version Code: 0)

Raw File Size: 40.7 MB

Compare with previous APK...

File	Raw File Size	Download Size	% of Total Download Si...
libs	111.8 KB	111.9 KB	0.3%
unity-classes.jar	111.8 KB	111.9 KB	0.3%
com	101.5 KB	101.5 KB	0.2%
unity3d	96.1 KB	96 KB	0.2%
player	96.1 KB	96 KB	0.2%
UnityPlayer.class	12 KB	12 KB	0%
s.class	5.8 KB	5.8 KB	0%
N.class	4 KB	4 KB	0%
ReflectionHelper.class	3.9 KB	3.9 KB	0%

```
6 package com.unity3d.player;
7
8 import ...
9
52
53 public class UnityPlayer extends FrameLayout implements IUnityPlayerLifecycleEvents {
54     public static Activity currentActivity;
55     private static final int ANR_TIMEOUT_SECONDS = 4;
56     private static final int RUN_STATE_CHANGED_MSG_CODE = 2269;
57     private static final String SPLASH_ENABLE_METADATA_NAME = "unity.splash-enable";
58     private static final String SPLASH_MODE_METADATA_NAME = "unity.splash-mode";
```

Reader Mode



Android - разбираемся

UnityPlayer - это FrameLayout

Находим, что в качестве view используется как раз он

Немного шаманства с передачей жизненного цикла

И... используем UnityPlayer напрямую!

M Android - разбираемся

```
object UnityManager {  
    @SuppressWarnings("StaticFieldLeak")  
    private lateinit var mUnityPlayer: UnityPlayer  
  
    var isUnityReady = false  
  
    /* Antol */  
    val unityView: View  
    | get() = mUnityPlayer  
  
    /* Antol */  
    @RequiresApi(Build.VERSION_CODES.R)  
    fun initUnity(activity: Activity) {  
        | mUnityPlayer = UnityPlayer(activity)  
        | isUnityReady = true  
    }  
  
    /* Antol */  
    fun onDestroy() {  
        | mUnityPlayer.quit()  
        | isUnityReady = false  
    }  
  
    /* Antol */  
    fun onPause() {  
        | mUnityPlayer.pause()  
    }  
  
    /* Antol */  
    fun onResume() {  
        | mUnityPlayer.resume()  
    }  
  
    /* Antol */
```

```
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        | super.onCreate(savedInstanceState)  
        | setContentView(R.layout.activity_main)  
        | UnityManager.initUnity( activity: this)  
        | if (savedInstanceState == null) {  
        | | supportFragmentManager.beginTransaction()  
        | | .replace(R.id.container, MainFragment.newInstance())  
        | | .commitNow()  
        | }  
    }  
  
    /* Antol */  
    override fun onDestroy() {  
        | UnityManager.onDestroy()  
        | super.onDestroy()  
    }  
  
    /* Antol */  
    override fun onPause() {  
        | super.onPause()  
        | UnityManager.onPause()  
    }  
  
    /* Antol */  
    override fun onResume() {  
        | super.onResume()  
        | UnityManager.onResume()  
    }  
  
    /* Antol */  
    override fun onConfigurationChanged(newConfig: Configuration) {  
        | super.onConfigurationChanged(newConfig)  
    }  
}
```

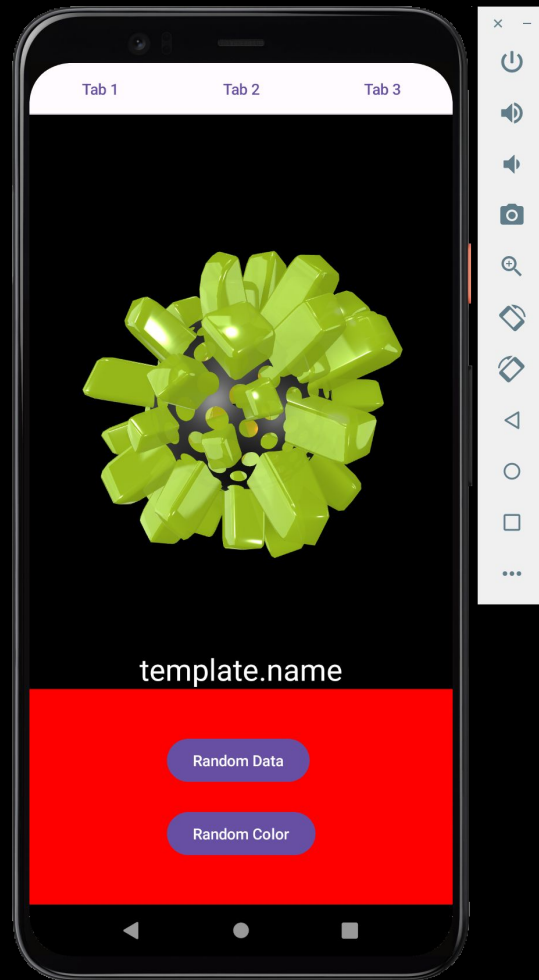


Android - теперь можно так!

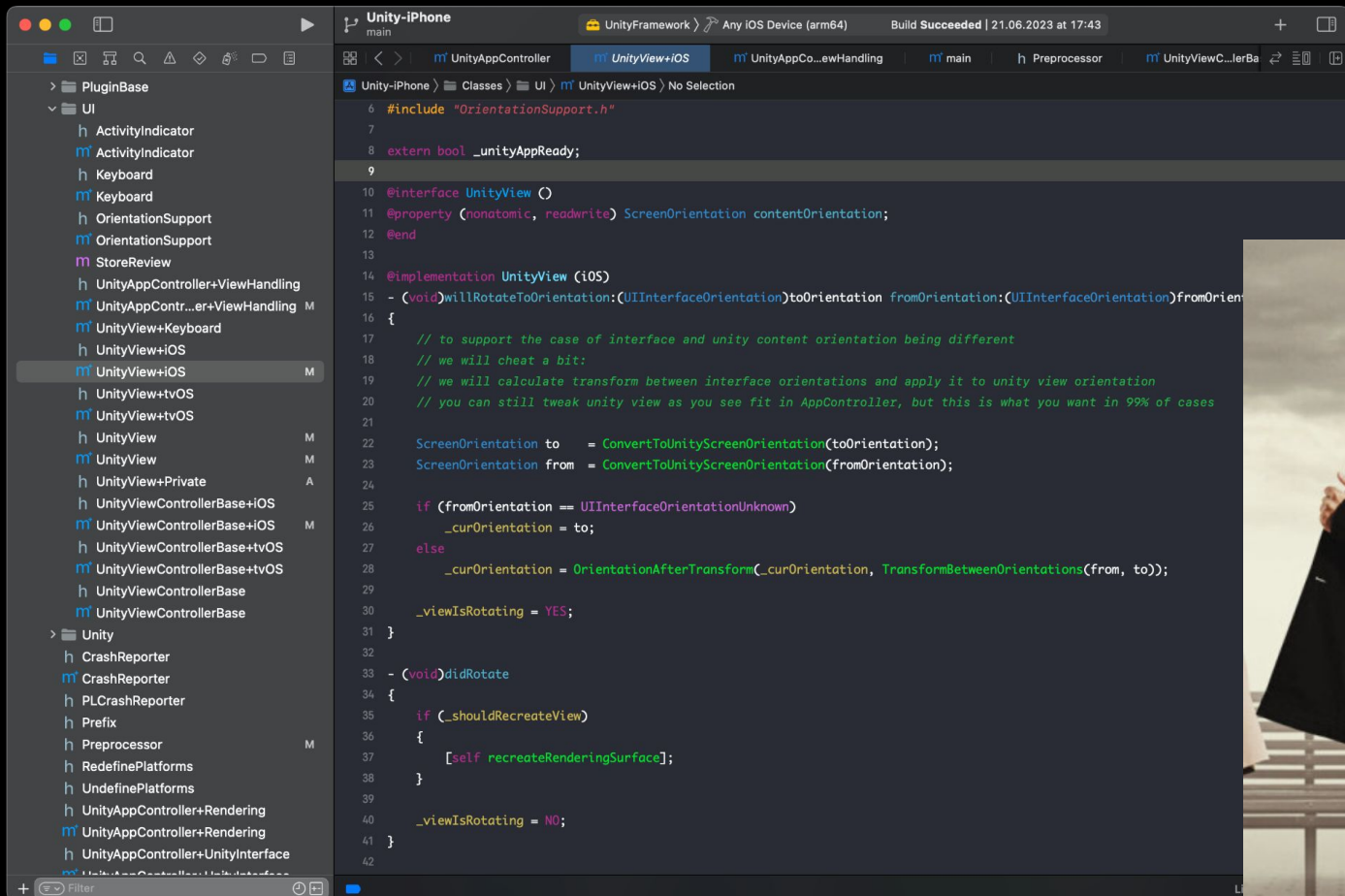
```
@Composable
fun UnityView() {
    Box(modifier = Modifier.fillMaxSize()) { this: BoxScope
        if (UnityManager.isUnityReady)
            AndroidView(
                modifier = Modifier.fillMaxSize(),
                factory = { it: Context
                    UnityManager.onPause()
                    UnityManager.onResume()
                    UnityManager.unityView() lambda
                }
            )
        else
            Text(
                text = "Unity is not ready",
                fontSize = 28.sp,
                color = Color.White
            )
    }
}
```



Android — PROFFIT!

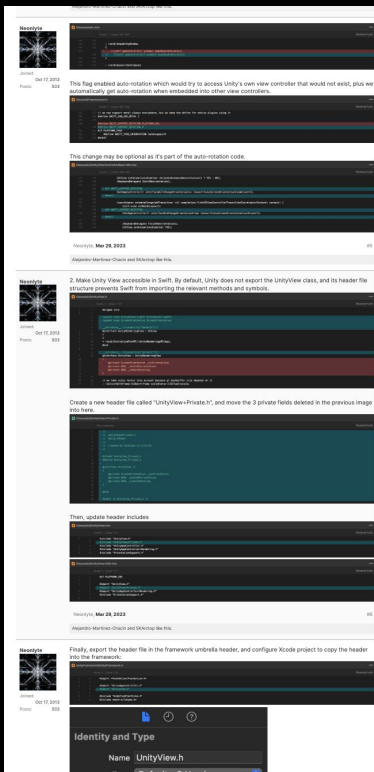
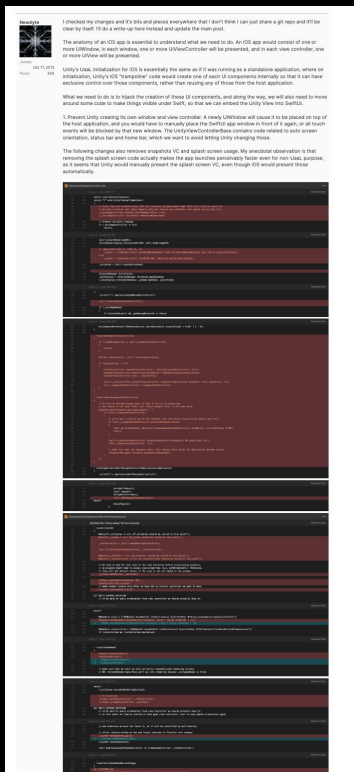


M iOS оболочка



❌ iOS — Низкий поклон

<https://forum.unity.com/threads/unity-as-a-library-rendered-on-partial-screen-with-swiftui.1398856/>



iOS — примеры исправлений

```
[[SwiftUI_fix]
// if (@available(iOS 13, tvOS 13, *))
//     _window = [[UIWindow alloc] initWithWindowScene: [self pickStartupWindowScene: application.connectedScenes]];
// else
//     _window = [[UIWindow alloc] initWithFrame: [UIScreen mainScreen].bounds];
```

```
- (void)createUI
{
    NSAssert(_unityView != nil, @"_unityView should be inited at this point");
    [[SwiftUI_fix]
//     NSAssert(_window != nil, @"_window should be inited at this point");
//
//     _rootController = [self createRootViewController];
```

```
[[SwiftUI_fix]
//     [_window addSubview: _rootView];
//
//     // We should have rootViewController set always, otherwise UIKit might throw exception
//     _window.rootViewController = _rootController;
//
//     [UIView setAnimationsEnabled: NO];
//
//     // make window visible only after we have set up initial controller we want to show
//     [_window makeKeyAndVisible];
```

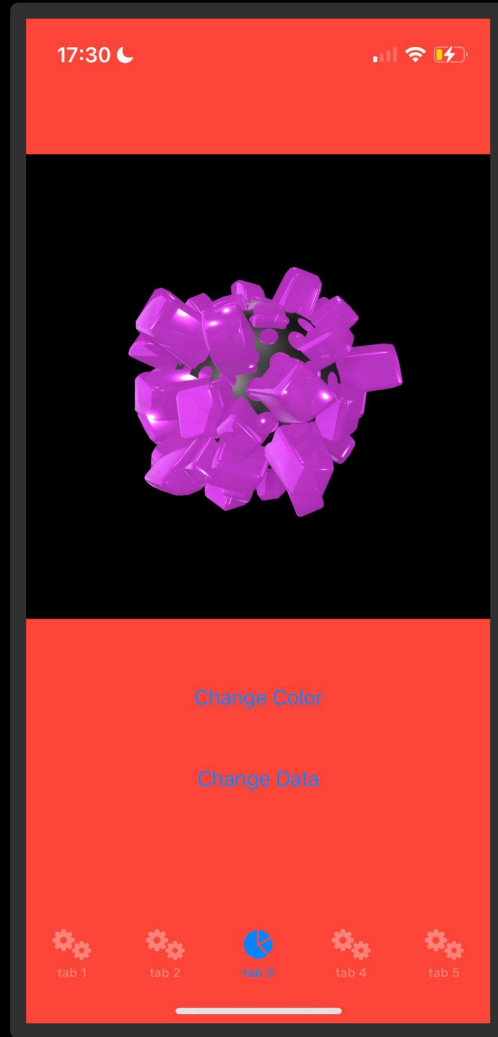


iOS - теперь можно так!

```
struct UnityWrapperView: UIViewRepresentable {  
    func makeUIView(context: Context) -> UIView {  
        initializeUnityIfNeeded()  
        let view = UnityInstance.appController().unityView!  
        return view  
    }  
  
    func updateUIView(_ uiView: UIView, context: Context) {  
    }  
}
```



iOS — PROFFIT!





Бонус

М

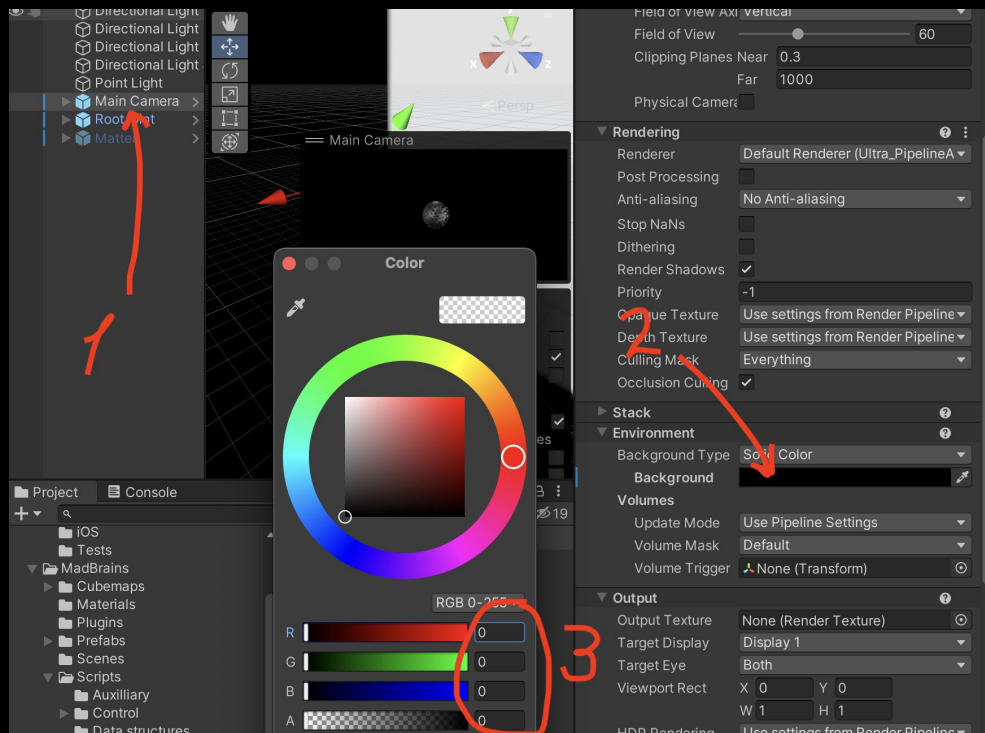
**Заказчик: А еще фон надо
прозрачный**

Я:





Unity: Можно попробовать





iOS: Да пожалуйста

Вообще без изменений в коде



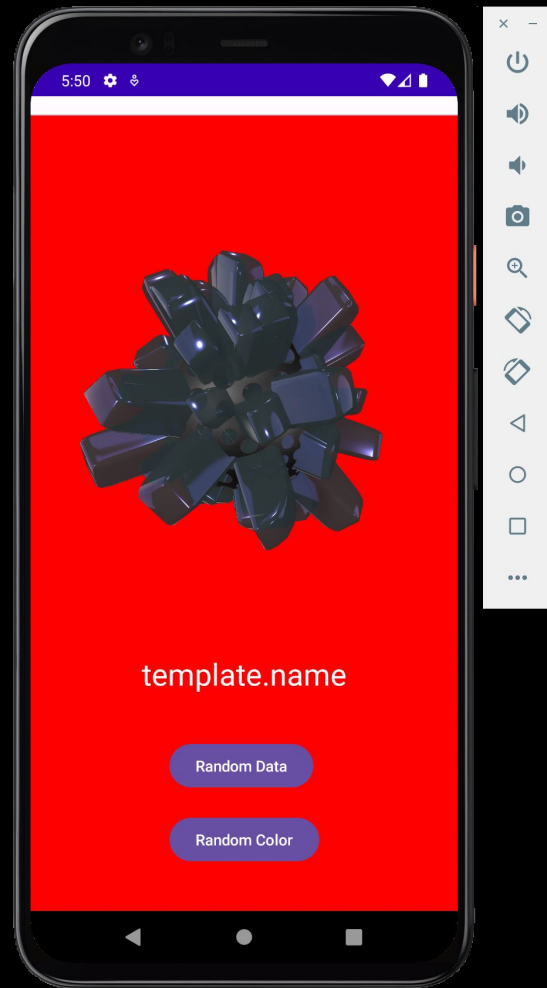


Android: тут такое дело...

```
@RequiresApi(Build.VERSION_CODES.R)
fun initUnity(activity: Activity) {
    mUnityPlayer = UnityPlayer(activity)

    val f: Field = mUnityPlayer::class.java.getDeclaredField("mGLView")
    f.isAccessible = true
    val glView = f[mUnityPlayer] as View
    glView.setBackgroundColor(Color.TRANSPARENT)
    val f2: Field = glView::class.java.getDeclaredField("a")
    f2.isAccessible = true
    val surfaceView = f2[glView] as SurfaceView
    surfaceView.setZOrderOnTop(true)
    surfaceView.holder.setFormat(PixelFormat.TRANSLUCENT)

    //window.insetsController is not working for unknown reason
    activity.window.clearFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN)
    isUnityReady = true
}
```





Всё!* Спасибо за внимание



*огромная куча технических деталей осталась за гранью доклада



MAD BRAINS

- madbrains.ru
- YouTube: www.youtube.com/c/MadBrains
- Telegram: [madbrainsru](https://t.me/madbrainsru)
- VK: vk.com/madbrainsru



- antol@madbrains.ru
- [@a_madbrains](https://t.me/a_madbrains)

