

VRコンテンツの つくりかた 2



れいぞうこのドア

VR コンテンツのつくりかた 2

youten 著

2017-10-23 版 発行

目次

第 1 章	はじめに	3
1.1	賞味期限とリポジトリ公開について	3
第 2 章	Daydream を知る	5
2.1	Daydream とは	5
2.2	Cardboard と Daydream の歴史的経緯	13
2.3	Daydream standalone	14
2.4	AR と ARCore とポジショントラッキング	15
第 3 章	Daydream を使う	18
3.1	Daydream のセットアップ	18
3.2	デバッグ設定と VR Entry Flow	21
3.3	Daydream View 互換機 BOBOVR Z5 を試す	27
第 4 章	Unity で Daydream アプリ開発の入り口を覗く	29
4.1	Hello, Daydream	29
4.2	Instant Preview	30
4.3	Daydream Elements	33
4.4	VR Video と Daydream Media App Template アプリ	35
4.5	6DoF モーショントラッキングシステム NOLO を試す	38
あとがき		44

第 1 章

はじめに

半年前にはもうそろそろ XR 元年がバズっていると考えていたのですが、よく言われる通り VR のハイプ・サイクルが少しフェーズが進んだところまでいってしまって落ち着いていたり、意外と AR が単独で盛り上がっており、iPhone のほんわかとした影響力を感じる 2017 年をお過ごしのみなさま、いかがおすごしでしょうか。筆者の youten と申します。

本書は、以下のような方をターゲットにしています。

- もうそろそろ日本でもデビューしてメジャー街道まっしぐらが約束されているはずの Daydream のことが知りたい
- Daydream View を買わずになんとか Daydream を試す方法が知りたい
- VR や Android や Unity やプログラミングについてちょっとは知っていて、Daydream アプリを開発してみたい

1.1 賞味期限とリポジトリ公開について

なお、本書は記載内容の賞味期限が切れてしまうことへのアップデート対応、あまり紙メディアが向いていない環境等のバージョンアップに追従するため、初版以降の PDF と関連ファイル一式を全て以下のリポジトリで公開予定です。あらかじめご了承ください。

- <https://github.com/youten/howto-create-vr-contents2>
 - Re:VIEW の素材一式、出力 PDF、関連プロジェクトのソースコードを全て含む想定です。

本書のうち、私 youten が著作権を有する範囲のライセンスについては、文章は CC-BY 4.0 ライセンス^{*1}、ソースコードについては Apache License v2^{*2}を適用します。

本書籍は Re:VIEW で作成されており、その設定ファイル等について、MIT ライセンス^{*3}に基づき「C89 初めての Re:VIEW v2」リポジトリ^{*4}で公開されているものを利用させていただいております。

表紙他、高崎柚乃モデル^{*5}を CC-BY 4.0 ライセンス^{*6}に基づき利用させていただいております。

^{*1} <https://creativecommons.org/licenses/by/4.0/legalcode.ja>

^{*2} <http://www.apache.org/licenses/LICENSE-2.0>

^{*3} <https://opensource.org/licenses/MIT>

^{*4} <https://github.com/TechBooster/C89-FirstStepReVIEW-v2>

^{*5} © Gugenka ® from CS-REPORTERS.INC/YUNO <https://gugenka.jp/original/yuno-3d.php>

^{*6} <https://creativecommons.org/licenses/by/4.0/legalcode.ja>

第 2 章

Daydream を知る



▲図 2.1 Daydream ロゴ

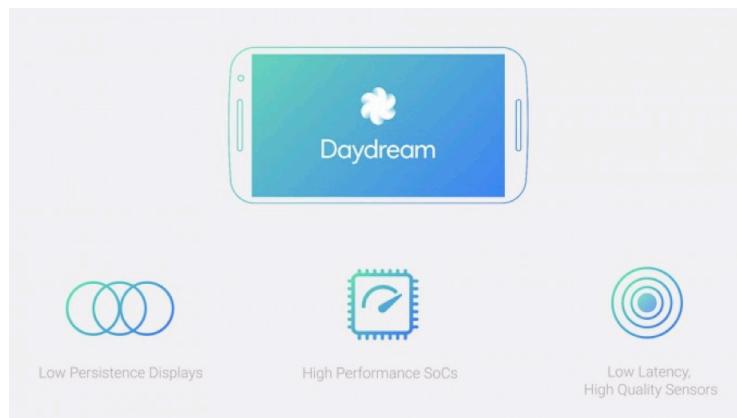
2.1 Daydream とは

Daydream とは、Google の Android スマートフォン向け VR プラットフォームです。ハードウェアからアプリマーケットまでを含む統合プラットフォームであり、主に以下に示す要素から構成されています。

- Daydream-ready と呼ばれる高スペックの Android デバイス
- ジャイロ・加速度センサを搭載し、Bluetooth で接続するコントローラとセットになったヘッドユニットの Daydream View
- Android 7.0 (Nougat) 以降での、低い入力遅延・高速な描画処理のための Android OS 内部での特別な対応
- Google Play とシームレスに融合し、HMD を装着したままでアプリのインストールから起動を行えるランチャー機能をもった専用のホームアプリ

それぞれの要素について、詳細を示します。

Daydream-ready phones



▲図 2.2 Daydream-ready phones の 3 要件

その母艦となるハードウェア要件があり、それを満たしたものを Daydream-ready phones^{*1}と呼びます。SoC、センサ、ディスプレイの3つの要件があります。

- 3D 画像処理と FPS を維持するための高いパフォーマンスを有する SoC
- 姿勢予測のための高い分解能と低遅延のセンサ
- 高い応答速度、低遅延のディスプレイ

SoCについては、現状 Daydream-ready phones に認定されている Android デバイスから判断すると Snapdragon 820 以上であれば満たすと考えられます。ただし、「MediaTek が Daydream のための SoC を開発する」というニュース^{*2}も発表されており、Snapdragon に限定はされていない模様です。

ディスプレイについては、現時点では種々の応答性能の都合上有機 EL のみが採用されています。「JDI の開発した VR 専用の超高精細・高速応答 液晶ディスプレイ^{*3}」のよ

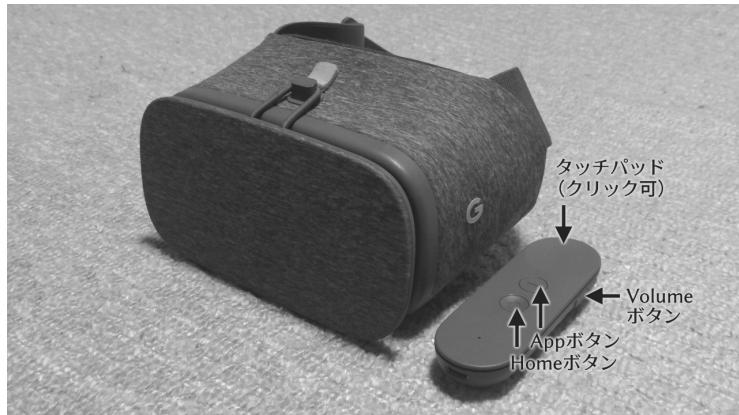
*1 Daydream-ready phones <https://vr.google.com/daydream/phones/>

*2 MediaTek's Next-Gen SoCs To Support Google's Daydream | Androidheadlines.com <https://www.androidheadlines.com/2016/05 MEDIATEKs-next-gen-socs-support-googles-daydream.html>

*3 ニュース&イベント：ニュースリリース | 株式会社ジャパンディスプレイ <http://www.j-display.com/news/2016/20161121.html>

うに、同等の性能があれば液晶ディスプレイでも要件を満たすと考えられます。将来は LCD を搭載した Daydream-ready phones も登場するでしょう。

Daydream View とコントローラ



▲図 2.3 Daydream View (旧型) 本体と専用コントローラ

専用コントローラ

Daydream の VR アプリを体験するには、レンズ付きヘッドユニットがセットになった Daydream View (図 2.3) 本体と、Bluetooth 接続の専用コントローラが必要です。

このレンズつきヘッドユニット部ですが、特別な回路は搭載されていないため、レンズ部分が同等であれば、実は別の Cardboard ゴーグルを利用しても動作させることができます。旧型の Daydream View には NFC タグが埋め込まれています。以前は NFC タグの読み込みが初期セットアップの途中で必要だったのですが、現在は不要になっています。新型の Daydream View には NFC タグは搭載されていないかもしれません。

このコントローラは IMU センサを含み、クリックのできるタッチパッド、App ボタン、Home ボタンとボリュームボタンを持ちます。また、Bluetooth で母艦である Daydream-ready の Android デバイスと接続して動作します。

このコントローラですが、BLE デバイスとしては単純なつくりになっています。Bluetooth LE (Low Energy) の用語で言うと、「ある Service のある Characteristic で 3 種のセンサとボタン・タッチパッドの情報を bit 列にして Notify しているだけ」です。

Daydream コントローラのハック

「iPhone でも Daydream コントローラは使えないか?」という話に始まった、Daydream コントローラをハックした事例があるので紹介します。

センサ類のイベント情報を BLE 経由で取得し、解析した話が「How I hacked Google Daydream controller - Hacker Noon^{*4}」という blog エントリに掲載されています。最終的には A-Frame を用いた iPhone 上の HTML コンテンツ上で Daydream コントローラの情報がとれるところまで簡潔にまとまっており、全貌を読み取るには少々の BLE の知識が必要ですが、ぜひ読んでみてください。

このハック内容によると、Daydream コントローラはセンサ情報を Bluetooth LE の GATT Notify で Central (スマートフォン母艦) 側に送信し続けています。その内容は、ジャイロ・地磁気・加速度センサそれぞれの X 軸・Y 軸・Z 軸の情報、Home・App・VolumeUp・VolumeDown・タッチパッドのクリック状態の各ボタンの on/off 情報、タッチパッドの XY 座標の情報などがぎゅっと 160bit の bit 列に詰められています。複雑な式による計算結果であったり、暗号化されたデータではないとのことです。

のことより、Daydream のコントローラ側はシンプルに IMU センサの取得値やボタン類の状態を送信するだけのつくりになっており、賢いアームモデルのエミュレーションはスマートフォン内部で行なっていることがわかります。

^{*4} How I hacked Google Daydream controller - Hacker Noon <https://hackernoon.com/how-i-hacked-google-daydream-controller-c4619ef318e4>

新型の Daydream View について



▲図 2.4 新型の Daydream View

つい先日の 2017 年 10 月 4 日、Google の新製品発表会が行われ、話題の Google Home ファミリーや Pixel 2/Pixel 2 XL とあわせて、新型の Daydream View^{*5}が発表されました（図 2.4）。特徴と、気になるポイントを以下に述べます。

- 重い Android デバイスを装着時にバランスが悪くなりやすかったため、取り外し可能な上部のバンドを追加。あわせて頬にあたる部分のクッション形状を変更
- 熱暴走対策として、抑え蓋がヒートシンクの役割をするように
- 價格は\$99
 - \$20 アップ
- 日本でも発売予定あり
- 視野角が 10 度ほど向上
 - 約 90 度→約 100 度
 - シンプルな凸レンズから、特殊な形状のフレネルレンズに変更
- IPD 調節機構はなし

値段アップは残念です。安価な中国製 Cardboard HMD が近年徐々に使い勝手がよくなってきている流れを汲んで、できれば値下げしてほしかったです。物価が気軽に上昇し

^{*5} Google Daydream View Tech Specs - Google Store https://store.google.com/us/product/google_daydream_view_specs?hl=en-US

がちなシリコンバレー感覚では、パワーアップしたのにおおむね価格据え置きで「勉強しました」という感覚なのかもしれません。

新型 Daydream Viewについて、GoogleがYouTubeにプロモーション動画を公開しています^{*6}。この動画中に出てくるChromecastを使った、外部ディスプレイへのキャストは、10月上旬ごろからすでに既存のDaydreamで使えるようになっています。Daydreamアプリのホーム画面から「キャスト」を選択すると常時ストリーム状態になります。パフォーマンスには影響がありそうですが、大きな遅延はなくきちんと動作しており、デモなどいくつかのシーンでは役に立ちそうです。

Googleのblogには年内発売と述べられています^{*7}ので、あわせてDaydream 2.0 EuphratesアップデートやHMDを外す必要がないフルVRモードのChromeアップデートもあわせて行われることを期待しています。

Android 7.0 Nougat と VR モード

Android 7.0 NougatでDaydreamのために特別な対応が入り、AndroidプラットフォームとしてはVRモードというものが定義されました。Androidデバイスの互換性について定義したCDD^{*8}というドキュメントがあり、その中にVirtual Realityという項目があります。CDDはAndroidデバイスそのものをつくる仕事でもしていなければあまり縁がないドキュメントなのですが、VRモードの定義の他、「DaydreamというVirtual Reality要件に対応したAndroidデバイスはどんな条件を満たさないといけないのか」という話はアプリ開発側の観点でも面白い情報が詰まっています。詳細を少しみていきましょう。

CDDの「7.9. Virtual Reality」の項目^{*9}に「7.9.1. Virtual Reality Mode」という記載があり、そこでは「VRモードをサポートすること。そのモードでは、通知などを双眼のHMDのためにステレオでレンダリングし、VRアプリケーションが動作している間はモノラル（单眼）のシステムUIを無効化すること。」とあります。ディスプレイの黒挿入モードやフレームバッファへのダイレクトレンダリング、センサ情報の低遅延伝達あたりはこの「VRモード」と直接は関係ない模様です。

^{*6} Meet Google Daydream View | Dream with your eyes open - YouTube <https://www.youtube.com/watch?v=PNBL2DpB1YE>

^{*7} Pixel 2 and Daydream View: new experiences in AR and VR <https://www.blog.google/products/google-vr/pixel-2-and-daydream-view-new-experiences-ar-and-vr/>

^{*8} Android Compatibility Definition Document <https://source.android.com/compatibility/cdd>

^{*9} Android CDD 7.9. Virtual Reality - https://source.android.com/compatibility/android-cdd#7_9_virtual_reality

7.9.2. Virtual Reality High Performance

If device implementations identify the support of high performance VR for longer user periods through the `android.hardware.vr.high_performance` feature flag, they:

- [C-1-1] MUST have at least 2 physical cores.
- [C-1-2] MUST declare `android.software.vr.mode` feature.
- [C-1-3] MUST support sustained performance mode.
- [C-1-4] MUST support OpenGL ES 3.2.
- [C-1-5] MUST support Vulkan Hardware Level 0 and SHOULD support Vulkan Hardware Level 1.
- [C-1-6] MUST implement `EGL_KHR Mutable_render_buffer`, `EGL_ANDROID_front_buffer_auto_refresh`, `EGL_ANDROID_get_native_client_buffer`, `EGL_KHR_fence_sync`, `EGL_KHR_wait_sync`, `EGL_IMG_context_priority`, `EGL_EXT_protected_content`, and expose the extensions in the list of available EGL extensions.
- [C-1-7] The GPU and display MUST be able to synchronize access to the shared front buffer such that alternating-eye rendering of VR content at 60fps with two render contexts will be displayed with no visible tearing artifacts.

▲図 2.5 CDD 7.9.2 Virtual Reality High Performance（冒頭）

つづけて「7.9.2 Virtual Reality High Performance」という項目に"high performance VR"という表現で、Daydream の要件が記載されています（図 2.5）。一部を抜粋して紹介します。

- これらの要件は `android.hardware.vr.high_performance` feature flag を有効にするのに必要な項目である
- [C-1-2] `android.software.vr.mode` feature flag の宣言が必須である

Feature Requirements^{*10}に記載されている通り、この二項目が「Daydream が有効であるか」を判定するためのフラグになっています。逆に言うと、Android 7.0 Nougat 以降を搭載しており、rooted な Android デバイスに必要な apk などを移植してこのフラグを有効にすると、非対応デバイスであっても Daydream が動作することが確認されています^{*11}。実態としては Snapdragon820 以降であれば快適に動くとのことで、この CDD で記載された要件を満たしていればそりゃ動かないはずはないよね、という話の模様です^{*12}。

*¹⁰ Daydream Feature requirements <https://developers.google.com/vr/distribute/daydream/functionality-requirements>

*¹¹ Reddit Got Daydream to work on my 5X, might work on other Android 7.0 phones as well https://www.reddit.com/r/Android/comments/5hno6u/got_daydream_to_work_on_my_5x_might_work_on_other/

*¹² Android よくできてんな。

- [C-1-11] 最低でも 3840x2160@30fps-40Mbps の H.264 のデコードのサポートが必須である(1920x1080@30fps-10Mbps が4つ、あるいは 1920x1080@60fps-20Mbps が2つ同時にデコード可能、に相当)
- [C-1-12] 最低でも 1920x1080@30fps-10Mbps の HEVC と VP9 のデコードのサポートが必須、できれば 3840x2160@30fps-20Mbps に対応すべき(1920x1080@30fps-5Mbps が4つ同時にデコード可能、に相当)

4K の H.264 は対応してくれないと困るよ、というのが読み取れます。4K HEVC は SHOULD で書かれているので少し弱めです。

- [C-1-14] スクリーン解像度はフル HD (1080p) が必須、QuadHD (1440p) 以上を強く推奨
- [C-1-15] ディスプレイサイズは 4.7 インチから 6.3 インチの間であること
- [C-1-16] ディスプレイは VR モード時に最低 60Hz で動作すること
- [C-1-17] ディスプレイは Gray-to-Gray、White-to-Black、Black-to-White がの応答性能が 3ms 以下であること
- [C-1-18] ディスプレイは残像が 5ms 以下の低残像モードの対応が必須

このあたりがディスプレイ要件です。現時点の Daydream 対応スマートフォンは全て有機 EL ですが、VR 向けの低残像・高密度・高い開口率の液晶が徐々に発表されるプロダクトにも適用されはじめています^{*13}。

- [C-1-19] Bluetooth 4.2 と Bluetooth LE (Low Energy) と Data Length Extension 対応必須

BLE は Daydream コントローラ用の要件です。

- [SR] android.hardware.sensor.hifi_sensors の対応を強く推奨

hifi sensor は Android 6.0 から定義が追加された「高い忠実性 (high fidelity) をもつセンサ」のことです。Android の要件としては加速度センサやジャイロセンサの性能は「50Hz が必須」レベルなのですが、hifi sensor では「400Hz またはそれ以上が必須」となっています。頭部の姿勢予測に使われているものと考えられます。

^{*13} Pioneering the Frontier of VR: Introducing Oculus Go, Plus Santa Cruz Updates <https://www.oculus.com/blog/pioneering-the-frontier-of-vr-introducing-oculus-go-plus-santa-cruz-updates/>

2.2 Cardboard と Daydream の歴史的経緯

2014年5月、Google I/O 2014のkeynoteで、ダンボールにアクリル製レンズをはめ込んだゴーグル型のVRデバイス、Cardboardが発表されました。Googleが設計図を公開したことと、その構成要素がシンプルであったため、互換品が数多く制作され、世界中に広まっていきました。

Oculus Rift DK1/DK2によって確立された、ジャイロセンサによる姿勢予測、スマートフォン向けの（安価な）高精細で応答速度に優れた有機ELディスプレイと、複雑な光学系機構の代わりにソフトウェア側でレンズにあわせて画面表示を歪めておく樽型歪み補正（Barrel Distortion）を組み合わせ、安価なデバイスで高度なVR体験を実現する手法を、そのままその要素を搭載したスマートフォンで実現したのがCardboardでした。

ところが、Cardboardはあくまでエントリー向け、PCのそれと比較して機能が劣るのはもちろん、場合によっては安からう悪からう低品質なVR体験デバイスとしての地位を確立してしまいました。

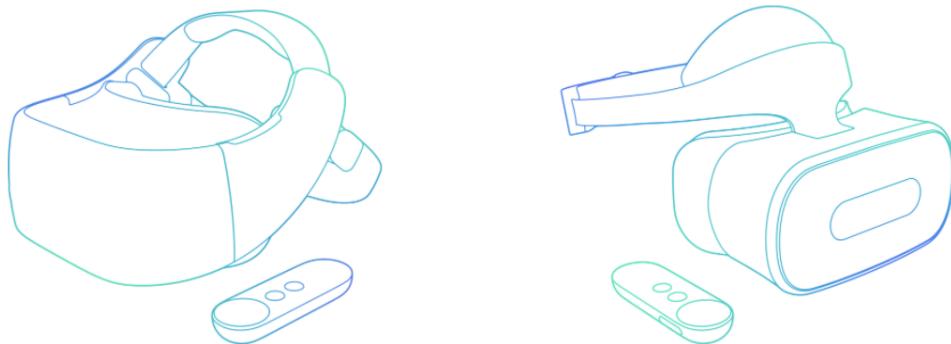
主に3つの要因があります。1つ目は、Androidデバイスにとって加速度・地磁気・ジャイロセンサの高い測定分解能がハードウェアとしての必須要件ではなかったこと。特にローエンドのAndroidデバイスに多いジャイロセンサ未搭載のデバイスではドリフト対応の副作用として、Google VR SDKは正常に動作しません。2つ目は、幅広いAndroidデバイスのスペックバリエーションに対して、VR空間を実現するのに十分なクオリティで60fpsを維持したCGのレンダリングが大変シビアであったこと。3つ目に、いわゆる「360度動画」と呼ばれる、ステレオ全天球などの形式である動画コンテンツの大きな解像度・fpsを問題なく扱うことができるハードウェアチップはごくハイエンド向けに限られることが挙げられます。

そんな状況の中、2016年5月、Google I/O 2016にて、Google VRの新たなプラットフォームとしてDaydreamが発表されました。Daydreamは前述の問題点が、特定の基準を満たしたハードウェア要件を満たすDaydream-ready phonesと、Androidプラットフォームとしてのソフトウェアでの対応が密に連携することで解決されます。また、それらに加えてコントローラの標準装備、HMDを外すことなくアプリを切り替えることができるホーム・ランチャー対応とGoogle Playへの融合、「なるべく快適にコンテンツを楽しむ効率をあげる」ための工夫がDaydreamにはたくさんつまっています。

モバイルVRの基準が、Daydream級にすみやかに移行してほしいと思います^{*14}。

^{*14}iOS 12でVRKitが出て、オルタナティブガールズとかときめきアイドルとかラブプラスEVERYとかがあっさりとCardboardを捨ててくれる未来を夢見ています。

2.3 Daydream standalone



▲図 2.6 Vive (HTC) (左) と Lenovo (右) の Daydream standalone

2017年5月、Google I/O 2017にて、スタンドアローン型のDaydreamヘッドセット、Daydream standalone^{*15}が発表されました。

ひとことで言うと「Androidが埋まっていて、電源を入れるとDaydreamのホームアプリが立ち上がるVRヘッドセット」です。特徴はTangoをベースとしたWorldsenseと呼ばれる技術により、ポジショントラッキング（平行移動）に対応していることです。

年内発売予定で、Vive(HTC)製とLenovo製の2つ（図2.6）が予定されています。

今のDaydream（Cardboardも同様）はヘッドトラッキングと呼ばれる、頭の向きのみを考慮したトラッキングになっています。体を大きく前に乗り出したり、敵の弾を避けようと横に上半身を躲すような動作に対応しておらず、そのような動作をした際には体の動きとヘッドセットで表示される視界に不一致が起きるため、酔いに繋がってしまいます。

そのため、現行のDaydreamでは「回らないソファにゆったりと座った状態で軽く首を動かすことを想定し、大きく体をひねって真後ろの敵を撃つようなコンテンツは避けること」といったことが推奨されています^{*16}。

つい先日、Oculus Connect 4で、スタンドアローン型のヘッドセット、Oculus Goが発表されましたので、せっかくなので比較してみましょう（表2.1）。どちらも発表時のみの情報であり、実際に発売されるまでには変更される可能性が高いことはご了承願い

*15 Daydream standalone <https://vr.google.com/daydream/standalonevr/>

*16 VR／AR／MRは「空間的コンピューティング」の波だ Googleが語るDaydream・Tangoの基礎【Unite】 | PANORA VR <http://panora.tokyo/27157/>

ます。

▼表 2.1 Daydream standalone と Oculus Go

項目	Daydream standalone	Oculus Go	備考
ベース技術	Daydream + Tango	Gear VR	
発売元	Vive (HTC) , Lenovo	Oculus	
アーリリストア	Google Play	Oculus Store	
価格	未定	\$199	
SoC	Snapdragon 835	不明	
ヘッドトラッキング	対応	対応	
ポジショントラッキング	対応 (WorldSense)	非対応	
発売時期	年内予定	2018 年	
コントローラ	3DoF + タッチパッド + ボタン	3DoF + タッチパッド + ボタン (トリガ)	

Oculus Go はコアとなる SoC について、その価格からミドルレンジのデバイスであると推測されます。Daydream standalone は Snapdragon 835 ベースでハイエンドのデバイスであると推測されます。

これらのスタンドアローン機はどのようなメリットがあるのでしょうか。現行機の「Galaxy S8 と Gear VR のセットで\$749」や「Pixel 2 と Daydream View のセットで \$749」という価格帯は決して気軽に購入できるものではありません。「Oculus Go の\$199」や「Daydream standalone の\$599」は販促目的での VR 体験キャンペーンなどに 20 台とか用意する際の費用や、物理的な取り回しにおいてメリットがある、という無難な期待をしておくのが妥当そうでしょうか。

ただし、そもそも日本での入手性が不明なため、残念ながらどちらも「VR マニアなら持ってる」という一品で終わってしまう可能性もあります。

2.4 AR と ARCore とポジショントラッキング

AR 戦国時代がいつのまにか始まりました。HoloLens の MR や、2 機種目が市販された Google Tango が盛り上がって…というテクノロジー主導ではなく、どちらかというとソーシャル主導で始まったように見えています。

写真 SNS である Instagram が台頭する中、楽しく"盛れる"snowなどのアプリによるリアルタイムの画像処理技術の進化競争が行われていた中、"たまたま"ポケモン GO がやってきました。ポケモン GO で、人は「カメラ映像に CG を重畳する AR」を思い出しました。この流れの分かりやすい例としては、Snapchat による、世界を AR で装

飾する World Lenses^{*17}があります。

ポケモン GO はなんとなくあの UI になってしまったわけではなく、「現実世界に拡張した情報を重ね合せる」という本質的な AR サービスである Google マップを開発していたバックグラウンドがあって、十分に人の心が動いて、その結果として体が動いてもらうための機能を検討をした上で今のような実装になったことが、CEDEC 2017 のセッションで説明されています^{*18}。

そんな中、まっすぐに AR 機能を提供する ARKit が iOS11 でやってきました。ARKit は買収した Metaio^{*19}の技術が使われていると言われていますが、こちらもタイミングとしては"たまたま"できあがったものと推測されます。なぜなら、歴史的に iOS は「既存のサービスのクローンを作りやすくする機能、あるいはクローンそのもの」をぶつけてくることが多いのですが、ARKit についてはそうではありません。前述の写真を"盛る"ための機能として iPhone X の TrueDepth カメラシステムによる 3D ウ〇コがやってきた話とは違うのです。

そんな ARKit ですが、世間におおいに担ぎ上げられたため、Google は大慌てで Tango を ARCore にピボットすることになりました。その ARCore は ARKit とほぼ同等で、水平面しか検出できず、Depth もとれず、Area Learning による外界検出結果の保存・読み出しには対応していません。Tango と比較して、機能としてはかなりの劣化版です。

本質的な AR は Google マップすでに日常使われています。「カメラ映像に CG を重畳する」の延長線上としてみんなが期待する電腦メガネやオーグマーは、HoloLens がその入り口に立ったところで、もう少し将来の話です。今の AR ブームは波がすぎると、現状のニッチであっても市場ができた VR のように定着しないと思っています。

そんな ARKit と ARCore ですが、VR の世界から眺めて「モバイルにポジショントラッキング技術が来た」と考えると、次の一手が楽しみになります。Windows MR や Oculus の Santa Cruz のように複数のカメラでもなければ、Tango や iPhone X のように Depth カメラを持つわけではない、ハードウェアがよりシンプルで難易度の高い Visual SLAM に Apple と Google というモバイルの二大巨頭がパワーをつぎ込んでくれるわけです。

iOS 12 で VRKit が来るのが先か、\$199 の Daydream standalone 2 が来るのが先か、

*17 Snapchat adds world lenses to further its push into augmented reality - The Verge <https://www.theverge.com/2017/4/18/15333130/snapchat-world-lenses-something-new-for-facebook-to-copy>

*18 『ポケモン GO』を開発したナイアンティックが目指す“世界を動かす AR”【CEDEC 2017】 ファミ通 App https://app.famitsu.com/20170901_1129022/

*19 Apple が定評のある拡張現実スタートアップ Metaio を買収していた | TechCrunch Japan <http://jp.techcrunch.com/2015/05/29/20150528apple-metaio/>

Tango の取り下げで凹んだ分はきっちり他の分野が凸ってくれるといいなと思っています。

第3章

Daydream を使う

この章では、Daydream の使い方について紹介します。セットアップから始まり、Google Play との連携具合や、開発する際に便利なデバッグ設定について記載します。

3.1 Daydream のセットアップ

初期セットアップ



▲図 3.1 Daydream のセットアップ

Daydream のアイコンから Daydream アプリを立ち上げます。まずは VR モードではない通常の 2D の UI で、初期セットアップウィザードが始まります。Daydream のアプリも通常のアプリと同様の Google Play で公開されており、そこからダウンロードする仕組みになっていますので、Google アカウントを選択します（図 3.1-左）。

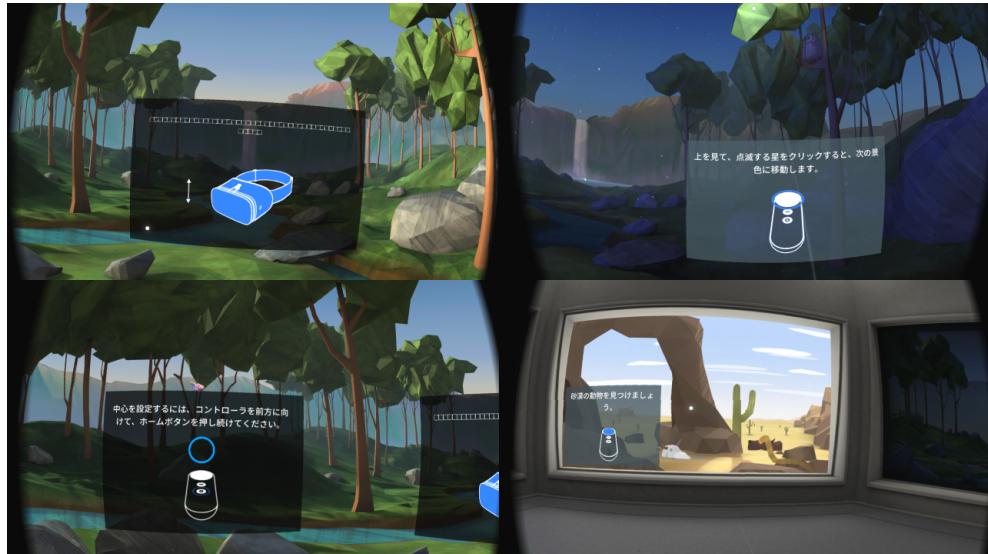
Daydream 対応ヘッドセットを持っている場合のみ続行できます（図 3.1-中）。ただし、抜け道があり、Daydream View を持っていないとも、Bluetooth 経由で別の Android デバイスを Daydream コントローラエミュレータとして使うことができます。これについては後述します。



▲図 3.2 VR モードに移行する際に表示されるエントリー画面

Daydream コントローラのペアリングを行った後（図 3.1-右）、エントリー画面（図 3.2）を経て、ここ以降は VR モードでのチュートリアルが始まります。

チュートリアルとホーム画面

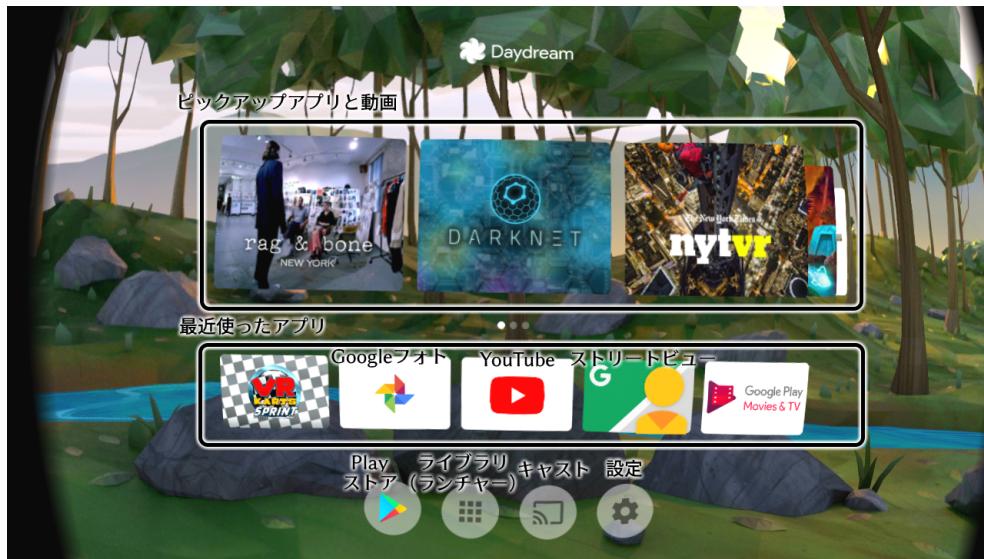


▲図3.3 チュートリアル画面

きちんとバンドがしまっていて、首をふってもヘッドセットがずれたりしないかの確認に始まり（図3.3-左上）、コントローラの使い方を順番に教えてくれます^{*1}。

とくにこのホームボタンの長押しによる中心のリセット（図3.3-左下）は重要操作なので覚えておきましょう。

^{*1} 2017年10月15日現在、手元の環境では一部の文字が豆腐になっており、見事に読みませんでした。日本語が正常に表示されている画面もあり、詳細は不明です。



▲図 3.4 Daydream のホーム画面

チュートリアルが終わると、Daydream のホーム画面が表示されます（図 3.4）。

ピックアップアプリや動画が上段に、中段には最近使ったアプリが並びます。下段は左から、アプリ「Play ストア」、インストール済みアプリ一覧の「ライブラリ」、Chromecast でアプリ画面を写す「キャスト」、「設定」です。

セットアップが終わると、どのアプリを試せばいいのか迷うのですが、まずは「Google ストリートビュー」をオススメします。次に、徐々に VR 空間での使い勝手がよくなってきた「YouTube」で 360 度・3D（ステレオで立体的に見える）動画を閲覧しましょう。もしあなたが THETA ユーザで、全天球の画像をそれなりに保有してあるのであれば、「Google フォト」も大変オススメです。

ゲームなどのダイナミックなコンテンツもいいのですが、現状はこれだ！ というキラータイトルが定まっていないように思えます。まずは写真・動画系のアプリで Daydream の操作に慣れたあとでゆっくり探して見てください。

3.2 デバッグ設定と VR Entry Flow

この節では、Daydream のデベロッパー向けのデバッグ設定を用いて、不要な操作を省略したり、Daydream View がなくとも Daydream アプリの利用や開発を可能にする方法を紹介します。ユーザにトラブルなく Daydream を利用してもらうための特別なフロー

である「VR Entry Flow」の他、様々な仕組みがあって、特別な設定や操作の上に可能となります。

VR Entry Flow

Daydream には、コントローラのペアリングや、必要な権限の設定を、ヘッドセットをかぶる前にチェックの上、必要な操作をユーザに促す VR Entry Flow^{*2}という作業フローが定義されています。

VR アプリの起動時など、非 VR の状態から VR モードに遷移する際に、以下のフローに従います。

1. 設定済みでなければ、コントローラをペアリングします
2. 必要があれば、コントローラのファームウェアをアップデートします
3. 必要があれば、Google VR Services を有効にして、必要な権限を設定します
4. Bluetooth や NFC の他、必要な設定が ON になっていなければ設定画面を呼び出します
5. 実施していなければ、Daydream ヘッドセットの設定を行います
6. ガイド（図 3.2）を表示し、スマートフォンをヘッドセットに挿入します
7. コントローラと接続し、ヘッドセットとコントローラの向きを初期化、キャリブレートします

これらのフローは Google VR Services によって自動的にハンドリングされ、通常アプリ開発者はこの流れを気にする必要はありません。ただし、ホームアプリのランチャー や、タスク一覧からあなたの VR アプリを立ち上げた際に、Android ライフサイクルに沿って起動したあなたの VR Activity は、VR Entry Flow によって割り込まれ、中断されます。Daydream アプリに限定された話ではなく Android アプリとして当然の話ではありますが、あなたのアプリがいつ中断されても問題がないように作っておく必要があります。

デベロッパー向けの設定

Daydream には、デベロッパー向けのデバッグ設定があります。このうち、「VR Entry Flow をスキップする設定」と「Bluetooth 経由でのコントローラのエミュレータを使う設定」の 2 つが大変重要ですので覚えておいてください。前者はコントローラを操作する必要がないアプリの利用時や機能の開発時に便利です。また、後者を組み合わせると、

^{*2} VR Entry Flow | Google VR | Google Developers <https://developers.google.com/vr/daydream/guides/vr-entry>

Daydream View を持っていないとも Daydream アプリの利用や開発を行うことが可能になります。

デベロッパー向けの設定については公式サイトの「The Controller Emulator^{*3}」のページに記載されていますが、手順のみがシンプルに掲載されており、少しあわづらいです。もう少し噛み砕いて手順を確認していきましょう。

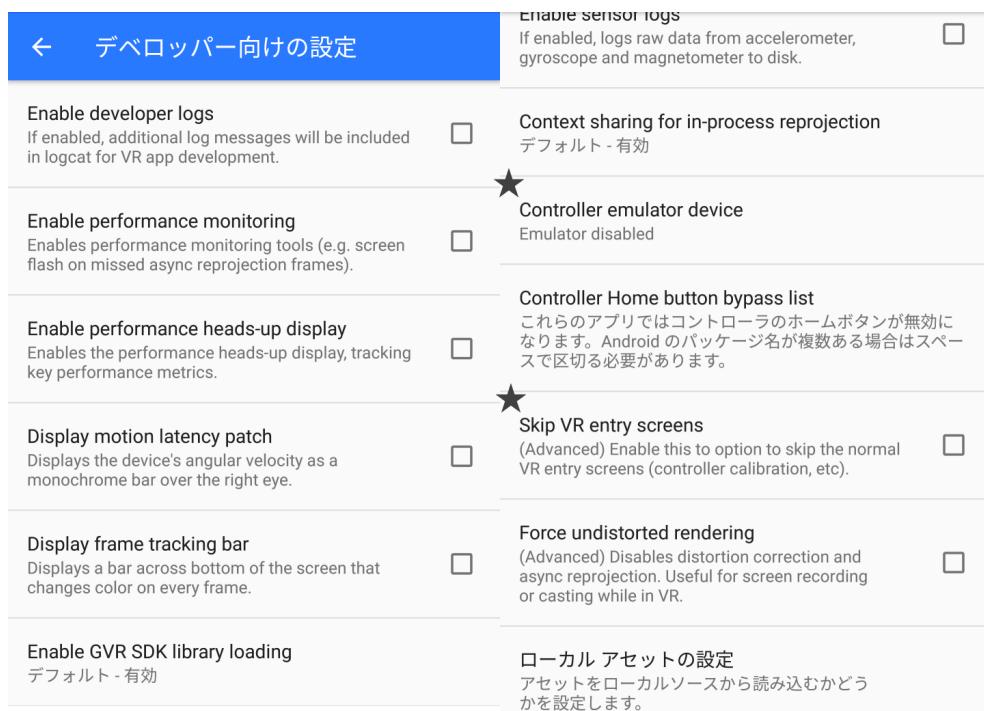
前節の Daydream の初期セットアップの最後、コントローラのペア設定画面（図 3.1-右）まで進めると、実は初期セットアップとしてはコントローラの設定を残して完了しているため、Daydream アプリを終了して、再度起動すると設定が完了したかのようにホーム画面が表示されます。



▲図 3.5 デベロッパー向け設定の有効化

ホーム画面の左上のハンバーガーメニュー(≡)から設定画面(図 3.5-左)を呼び出し、「ビルト バージョン」を7回タップすると(図 3.5-中)、「デベロッパー向けの設定」隠しメニューが表示されるようになります(図 3.5-右)。

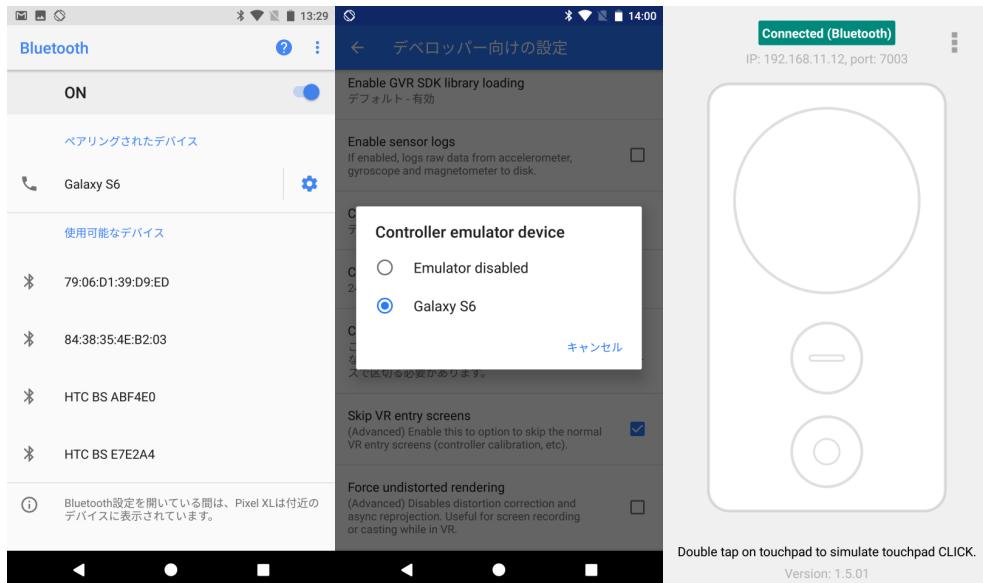
^{*3} The Controller Emulator | Google VR | Google Developers <https://developers.google.com/vr/daydream/controller-emulator>



▲図 3.6 デベロッパー向けの設定

ここまで、Daydream View とコントローラを使わないまま、セットアップを続けています。実際にアプリを起動した際にコントローラとのペアリングを要求されるのを回避するために、デベロッパー向けの設定中の、「Skip VR entry screens」を有効にしておきましょう（図 3.6）。

Daydream コントローラエミュレータ



▲図 3.7 コントローラエミュレータのセットアップ

続けて、コントローラエミュレータのセットアップを行い、動かしてみましょう。

エミュレータアプリは別 Android デバイスで

Daydream コントローラのエミュレータは、専用の Android アプリが公開されています。Daydream 母艦となる Android デバイスとは別の Android デバイスを用意してください。Android 4.4 KitKat 以降であれば動作します。Daydream-ready phone がもう 1 台必要というわけではありません。

Bluetooth のペアリング設定

まず、2 台の Bluetooth 設定を開いて、ペアリングを行います（図 3.7-左）。この際、エミュレータ側のデバイス名を区別がつきやすい名前に変えておくとわかりやすいです。

次に、Daydream のデベロッパー向け設定の「controller emulator device」を選択し、ペアリング設定を行ったデバイスを選択します（図 3.7-中）。

前述の公式サイト^{*4}から、「Controller Emulator」の apk をダウンロードし、インストールします^{*5}。このアプリは、Bluetooth 経由の他、WiFi や USB ケーブル経由で Unity や Unreal Engine などのエディタ上でのコントローラ操作のエミュレートにも使えます。

コントローラエミュレータを試す

セットアップが完了したら、実際に動かしてみましょう。Daydream アプリを立ち上げ、右下の VR ゴーグルアイコンをタップします。「Skip VR entry screens」設定が有効になっていると、本物のコントローラのペアリングも、ホームボタンの押下も要求されません。

二眼ステレオの VR モードになったら、もう一台の Android で「Controller Emulator」アプリを起動します。エミュレータアプリの表示が「Connected (Bluetooth)」になったら、正常に動作しています。うまくいかない際には、Bluetooth の設定や、アプリの起動順番に気をつけつつ、何度かやり直してみてください。アプリ UI 上にも説明文言が表示されていますが、タッチパネル部のクリックのエミュレートには、アプリ上のダブルタップで対応しています。

Daydream View がなくとも

ここまで手順で、Daydream View がなくとも、Daydream-ready phone またはその設定が可能な Android デバイスがあれば Daydream の世界をちょっと覗いてみたり、Daydream アプリ開発を試してみることができます。ただし、あくまでお試しで。Daydream の真の力はきちんとした環境でぜひ体験してください。

^{*4} The Controller Emulator | Google VR | Google Developers <https://developers.google.com/vr/daydream/controller-emulator>

^{*5} インストール手順は省略します。Android の SDK が導入済みであれば「adb install」を、一般ユーザは「提供元不明のアプリのインストール」を試してみてください。

3.3 Daydream View 互換機 BOBOVR Z5 を試す



▲図 3.8 BOBOVR Z5 Daydarem Edition

Daydream View 互換機とは

前章で述べた通り、Daydream View のヘッドユニットは Cardboard と同等のもので、コントローラは BLE で（比較的）シンプルな情報を送信しているだけです。このため、ハックの上、クローンが作れるということです。

BOBOVR Z5

Cardboard 向け VR ゴーグルや、独自の VR のプラットフォームを展開している BOBOVR から、BOBOVR Z5^{*6}という Daydream View 互換機が登場しました（図 3.8）。\$59 で AliExpress 経由で購入しましたが、今は Amazon^{*7}でも取り扱いがある模様です。

ヘッドユニットのレンズまわりに難あり

Daydream のコントローラ部は問題なく動作しました。タッチパネル部の形状の違いから操作感が違うのですが、許容範囲だと思います。

ただし、公式サイトにも書かれていますが、旧型の Daydream View の視野角である

^{*6} BOBOVR Z5 <http://www.bobovr.com/product/bobovrz5/>

^{*7} Amazon.co.jp : BOBOVR Z5 Daydream View VR ゴーグル <https://www.amazon.co.jp/dp/B071GYQR8V>

90度を120度に広げたヘッドユニット部の独自拡張がよろしくなく、端が歪んで見えます。首を振ると顕著に感じられ、気持ち悪く感じます。

結論としては、Daydream互換機として動作はしましたが、格別安いというわけでもなく、あまりオススメはできない機種です。日本特有の例のアレの話もありますし、おとなしく新型のDaydream Viewの発売を待つのが良さそうです。

第 4 章

Unity で Daydream アプリ開発の入り口を覗く

さて、概要を知って、使い方を知ったその次は、Daydream アプリ開発の入り口を覗いてみましょう。

Daydream も発表されて約 1 年半、リリースされて約 1 年経ちました。Google VR SDK（略称：GVR SDK）として、様々なライブラリやツールが揃ってきております。古い情報も残っていたり、Cardboard 向けや iOS 向けの話もあり、さあ冒険に出かけよう、と足を踏み出しても迷子になりやすい状況とも言えます。

この章では、入り口からの歩き方として、はじめに試すべきサンプルやツールの紹介をします。

4.1 Hello, Daydream

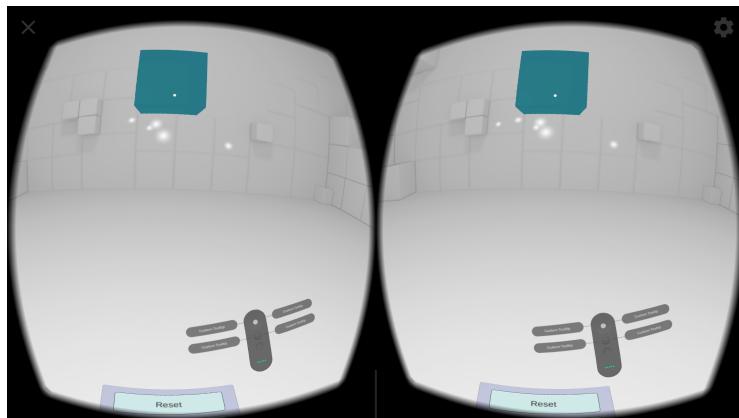
まずは Hello World からです。

Google VR SDK はネイティブ（Java, C/C++）や Unreal Engine もサポートしていますが、Unity の情報が一番多く、試す際にはオススメのプラットフォームです。本章では、とくに断りがなければ Unity のバージョンは「2017.2.0f3」を、GVR SDK のバージョンは「v1.100.1」を利用しています。

基本的には以下のページの手順に従います。

- Get started with Google VR in Unity on Android
 - <https://developers.google.com/vr/unity/get-started>
1. 新規 3D プロジェクトを作成し、「GoogleVRForUnity_1.100.1.unitypackage」をダウンロードしてインポート

2. API の自動アップグレード確認ダイアログが出るため、了承して実行
3. 「GoogleVR > Demos > Scenes」配下の「GVRDemo」シーンを開く
4. 「Build Settings」を開いて、「GVRDemo」シーンをビルド対象に追加、Platform を「Android」に「Switch Platform」を実行
5. 「Player Settings」を開いて、以下の設定を変更
6. 「Other Settings」より「Package Name」を適当な文字列に変更
7. 同じく「Other Settings」より「Minimum API Level」を「Android 7.0 'Nougat' (API level 24)」に変更
8. 「XR Settings」の「Virtual Reality Supported」をチェックして有効に、「Virtual Reality SDKs」の「+」ボタンを選択して、「Daydream」を追加
9. 「Build and Run」を実行、apk の名前は適当に入力



▲図 4.1 GVRDemo サンプルシーン

うまくいくと、灰色の空間上に、赤色のキューブが浮かんでいるアプリが立ち上がりります。Daydream コントローラを（ホーム長押しで）有効にして、ポインタをあててみましょう。赤色のキューブが青緑に色に変化します（図 4.1）。キューブをポインタで示して、タッチパネルをクリックすると、部屋内のどこかにワープします。

4.2 Instant Preview

前節で紹介した GVRDemo には Daydream 向けのステレオ表示とヘッドトラッキングの（自動的な）制御から、コントローラの接続状態のハンドリング、コントローラ操作での 3D オブジェクトに対する Ray（光線）制御によるヒット判定とメニュー操作と、基本

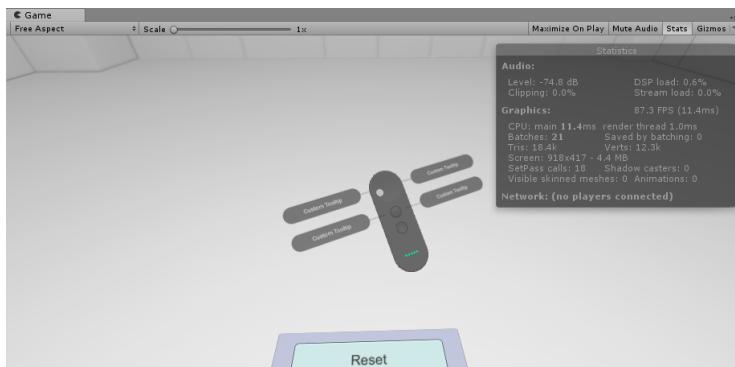
が詰まっており、根幹としてはこれが Daydream の全てです。

さて、つづけて具体的なアプリを開発していきましょう、となった際に、毎回ビルドと Android デバイスへの転送、インストールを待って…というサイクルを繰り返すのは効率的ではありません。できれば Unity エディタ上で再生して、開発サイクルを回していくたいですよね。

Unity エディタ上で頭の振りやコントローラ操作のシミュレーション

まず、Unity エディタの Game ウィンドウに対する操作で、頭を振ったり、Daydream コントローラの操作がシミュレートされています。GVRDemo のプレビュー再生中に、以下の操作を試して見てください。

- Alt + マウス移動：頭の回転を操作
- Ctrl + マウス移動：頭の傾きを操作
- Shift + マウス移動：Daydream コントローラの傾きを操作
- Shift + マウス左クリック：Daydream コントローラのタッチパッドのクリック
- Shift + マウス中央クリック：Daydream コントローラの Home ボタンのクリック（recenter 扱い）
- Shift + マウス右クリック：Daydream コントローラの App ボタンのクリック
- Shift + Ctrl + マウス移動：Daydream コントローラのタッチパッド上でのスワイプ操作



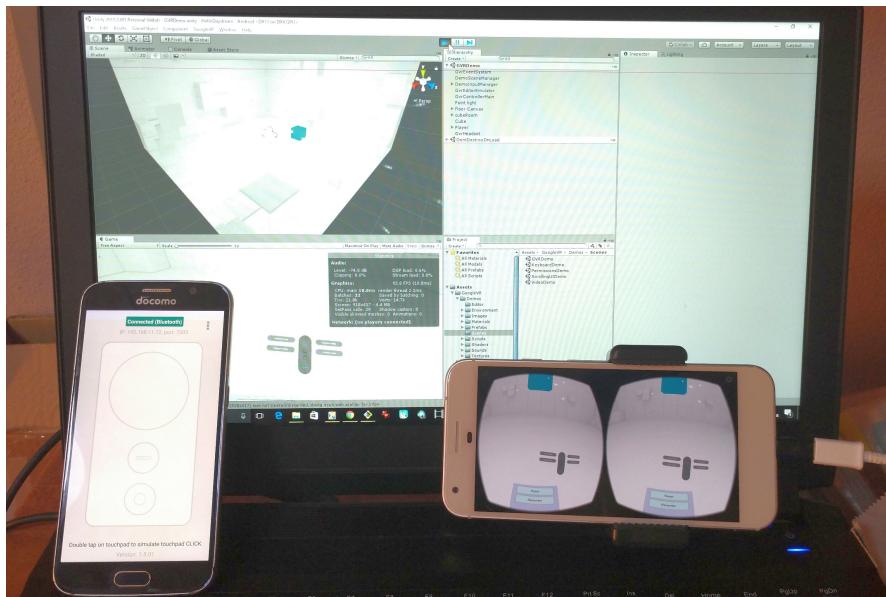
▲図 4.2 Unity エディタ上で Daydream コントローラのタッチパッドシミュレーションの様子

ひととおりの動作ができるようになっています。少々操作は複雑ですが、タッチパッドのスワイプ操作も可能です（図 4.2）。

ヘッドセットを被ってプレビューしたい

Unity エディタ上では平面しかありませんので、ヘッドセットを被った上で立体で確認したいですし、首振りやコントローラ操作はもう少し直感的に試したいところです。そこで Instant Preview の登場です。

- Instant Preview
 - <https://developers.google.com/vr/tools/instant-preview>



▲図 4.3 Instant Preview と Controller Emulator セットでの動作の様子

Instant Preview は、専用のアプリを Andorid デバイス上で動作させ、Unity エディタと通信させてこれを実現するものです。画像は Unity エディタから Instant Preview アプリへの映像ストリーミングで実現します。また、逆方向で Daydream コントローラの操作による入力も可能です。コントローラの操作については前章で紹介した Controller Emulator アプリによる Daydream コントローラのエミュレートにも対応しています（図 4.3）。USB・Wi-Fi 経由いずれも使えます。ただし、Wi-Fi 経由については映像ストリー

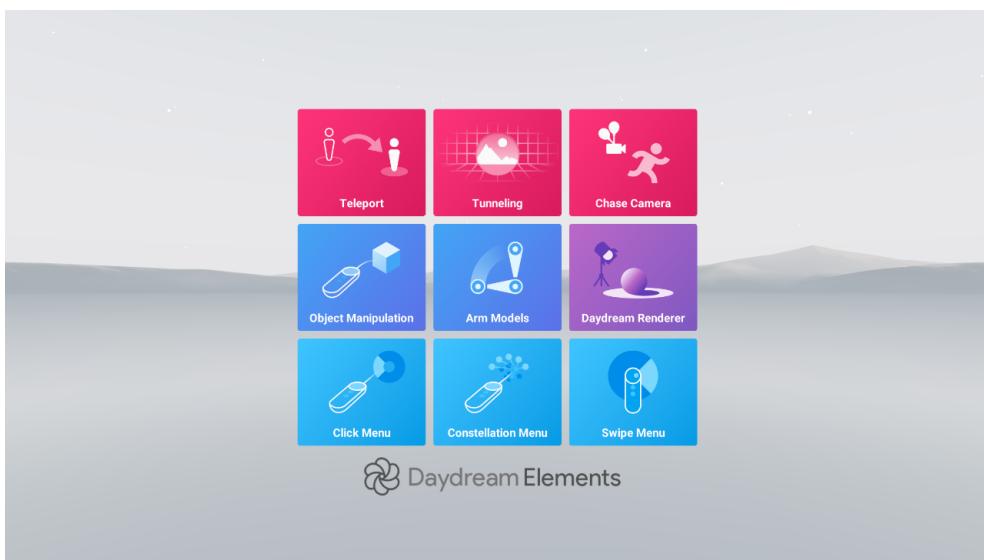
ミングのための通信帯域の観点や、特定のマルチキャストアドレスとポートによる通信が可能なネットワークである必要があり、USB 経由の方がお手軽です。

前節で動かした GVRDemo シーンに、Instant Preview に必要な Prefab がすでに含まれています。Daydream 母艦となる Android デバイスを USB で接続して、Unity エディタで再生すると、必要なアプリのインストールと起動は自動的に行われ、映像の再生がすぐに始まります。

コントローラについては、実デバイスであれば Home ボタンの長押しを、Controller Emulator であればアプリを起動して Bluetooth の接続を待つだけです。

adb (Android Debug Bridge) コマンドまわりを使っているため、Unity エディタから Android アプリのビルドとインストール・実行が正常に動く環境であれば問題なく動作するはずです。うまくいかない際には紹介したページ内に「Troubleshooting in Unity」という項目がありますのでそちらを参照ください。

4.3 Daydream Elements



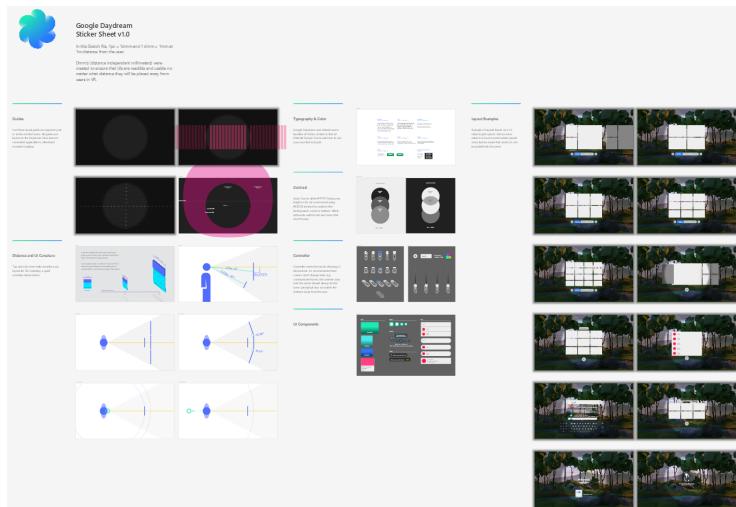
▲図 4.4 Daydream Elements 起動画面メニュー

- Daydream Elements Overview
 - <https://developers.google.com/vr/elements/overview>

Daydream Elements は Daydream のサンプル集です。アプリは Google Play で公開されている他、ソースコードも GitHub で公開されています。

ものすごくよくできたサンプル集で、以下に特徴をあげます。

- GVR SDK のライブラリ群の一つとして位置づけられ、多数のデモサンプルが含まれる
- 数多のデモサンプルは 1 つのアプリにまとまっていて、そのメニューや階層構造と各デモ間の遷移も操作感が大変よい
- 各デモは丁寧な説明看板と遊び心がたっぷり詰まったインターフェクションのあるプレイアブルなものになっている
- 統一感のある UI で Daydream のデザインガイドラインが直感的にわかるようになっている



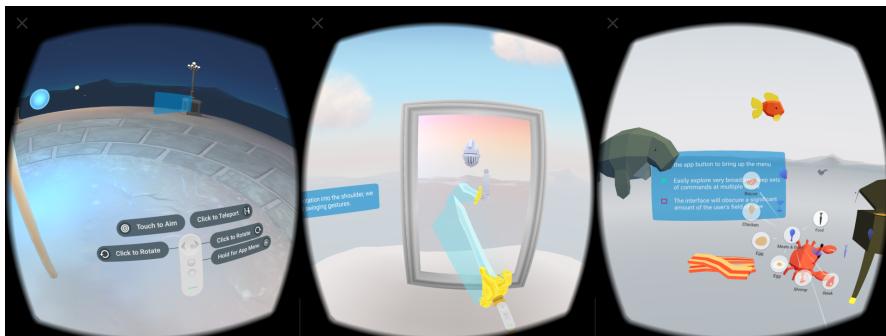
▲図 4.5 Daydream デザインガイドライン資料 (sketch ファイル) の一部

Daydream のデザインについては、ガイドラインがあります。Sticker Sheet というページ^{*1}から、sketch ファイルが 1 つ（図 4.5）と、Google I/O で解説した YouTube 動画へのリンクが貼られています。

Daydream のホームアプリはもちろん、Daydream Elements アプリや後述する Media App Template アプリもこのガイドラインでデザイン統一されています。一度目を通して

*1 Design | Sticker Sheet <https://developers.google.com/vr/design/sticker-sheet>

おくとデザインガイドラインとその根拠がわかります。



▲図 4.6 Daydream Renderer (左)・Arm Models (中)・Constellation Menu (右)

光源数などに制限がある、モバイル向けに最適化された動的なライティングを行う Daydream Renderer、様々な移動やメニュー・インタラクションのデモ、傾きのみの 3DoF ながら腕の構造のエミュレーションによって擬似的に 6DoF に近い動きを実現する Arm Model とそのパラメータを設定し、確認するデモなど、多数のデモが含まれています。ぜひそれぞれのドキュメントを読みながら確認してみてください。

4.4 VR Video と Daydream Media App Template アプリ

GVR video plugin

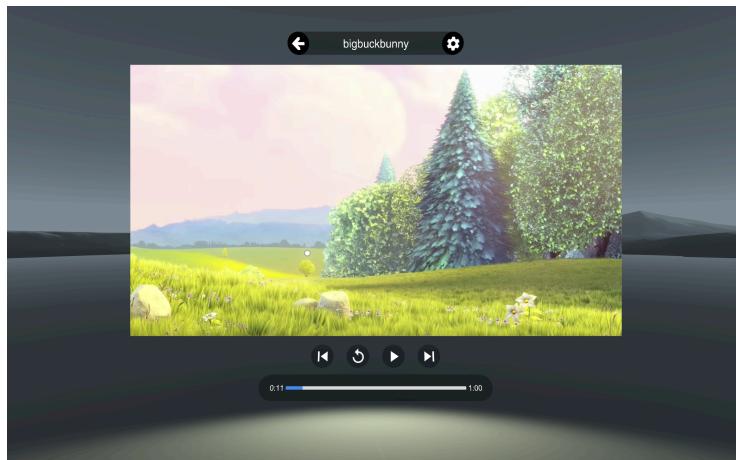
Daydream Elements と似た位置づけですが、GVR SDK for Unity には「GVR video plugin」という動画再生ライブラリが含まれます。

VR 動画として必要な、「モノラル/Top-Bottom のステレオ/Left-Right のステレオ」×「フラット /180 度/360 度」の 3x3 の 9 パターンの動画再生に対応しています^{*2}。ExoPlayer をベースにしており、シークやサムネイル取得も含め、ローカルファイルの再生のほか、HLS や DASH のような adaptive streaming^{*3}にも対応しています。

^{*2} フォーマットやスペックについては「第 2 章「Daydream を知る」」で述べた、CDD と Daydream の要件の項を参照ください。

^{*3} 直接の意味はネットワーク状態に応じて再生品質を制御するストリーミングのことです。ただし、ここでは「近年のいい感じのストリーミング技術」ぐらいの意味が近いと思われます。

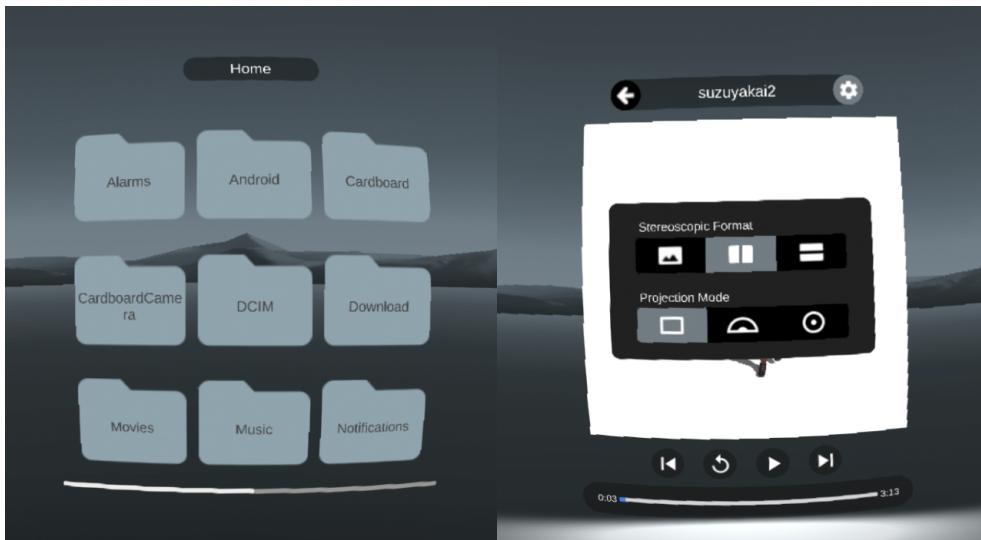
Daydream Media App Template



▲図 4.7 Daydream Media App Template アプリでの 2D 動画再生中画面

この GVR video plugin にも、Daydream Elements と同様に、メディア再生アプリのテンプレートとなるひととおりの機能の揃ったサンプルアプリ（図 4.7）と unitypackage 形式のリリースパッケージ、そしてもちろんソースコード一式が GitHub で公開されています。

- Daydream Media App Template
 - <https://developers.google.com/vr/unity/media-app-template>
- GitHub: googlevr/media-app-template: Daydream Media App Template
 - <https://github.com/googlevr/media-app-template>



▲図 4.8 ファイラー表示（左）と VR 動画のためのステレオ・投影設定メニュー（右）

特徴を以下に述べます。

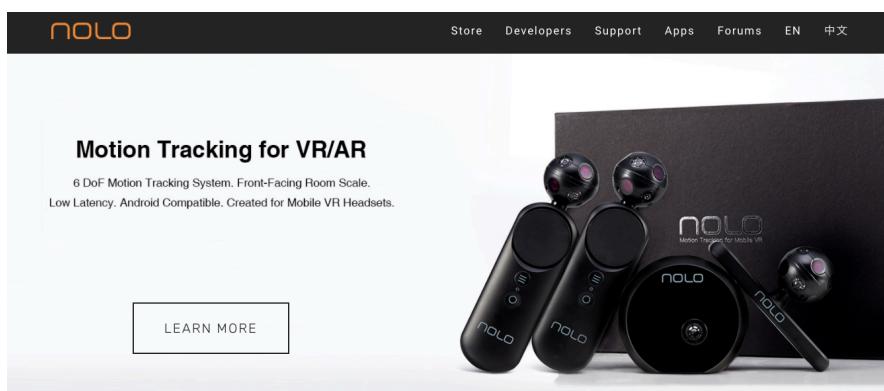
- メディアライブラリにアクセスするための Permission をユーザに確認するフロー実装に対応
- ファイラーとしてフォルダ表示と移動、ファイルサムネイル表示とそれらの選択に対応（図 4.8-左）
- 静止画・動画の再生に対応
 - 静止画またはフラット動画は縦スワイプによる拡大縮小、ドラッグによる表示位置の移動に対応
 - VR 動画としてのフォーマット選択メニューでステレオの Top-Bottom/Left-Right や 180 度/360 度の切り替えにも対応（図 4.8-右）
 - 動画のシークに対応、ポインタのオーバーレイによるシーク位置のサムネイルプレビューにも対応
- Daydream のデザイン・つくり観点でのガイドラインを満たしている

VR 動画はドラッグによって水平方向の回転も可能です*4。

*4 縦方向（ピッチ）の回転にも対応しているとベストだったのですが、それぐらいは自分で改造すればいいってことですよね、はい。

ローカルに VR 動画の資産がいくつかある方はぜひ試してみてください。サンプルアプリとは思えないレベルで作り込まれているなど感じました。ストリーミングサービスのバックエンドがあるなど、特定目的の Daydream メディアアプリを開発したい際には、大変有力な手助けになると思われます。

4.5 6DoF モーショントラッキングシステム NOLO を試す NOLO と Kickstarter



▲図 4.9 NOLO 公式サイト

NOLO^{*5}は6DoF のモーショントラッキングシステムです(図 4.9)。ベースステーション、ヘッドセットマーカー、2つのモーションコントローラからなるセットです。モバイル形態で利用時には、完全にケーブルレスで、フロント方向限定ながら2つのコントローラを含むルームスケールでのポジショントラッキングを実現します。PC 経由で利用時にはケーブルが1本だけ伸びますが、RiftCat + VRidge^{*6}を利用することで、低遅延の映像ストリーミングにて、SteamVR のアプリをプレイすることができます。

2017 年の 1 月末に Kickstarter^{*7}でのクラウドファンディングが開始され、NOLO 一式のコースが\$89～\$109 とそれなりの低価格で設定されました。Cardboard や Gear VR 環境のスマートフォンは持っていても、HTC Vive や Oculus Rift などの PC VR 環境

*5 NOLO - Motion Tracking for VR <https://www.nolovr.com/>

*6 VRidge - Play PC VR on your Cardboard <https://riftcat.com/vridge>

*7 NOLO: VR Motion Tracking for Mobile and SteamVR Play by Lisa Zhao — Kickstarter <https://www.kickstarter.com/projects/243372678/nolo-affordable-motion-tracking-for-mobile-and-ste>

を持っていないユーザにとって、NOLO は価格面においても期待された選択肢の一つでした。

ケーブルレスでポジショントラッキング付き、「6DoF のモバイル VR」としては Daydream standalone や Oculus の Santa Cruz や、ARKit/AR Core + Daydream が競合デバイスと考えられますが、NOLO はその先駆者と言えます。NOLO のもう一つの形態である PC 経由については、HTC Vive のケーブルが新型で細くなったことや、Steam VR 対応の Windows MR の登場もあって残念ながら今後はかなり厳しそうです。

モバイルモードでアプリを試す



▲図 4.10 モバイルモードでの NOLO 一式

さて、そんな NOLO ですが、back して届いたものを長らく放置していたのですが、いまさらになって試してみました。ここからは試した内容についてお届けします。

NOLO の公式サイト、「Support」内に英語版の「Product Manual」が公開されています^{*8}。初期セットアップの「Pairing」と、今回は「Android Phone」(Cardboard 相当と思われます)を選択します。

ベースステーションの電源を入れてペアリングモードにした後、コントローラ 2つとヘッドセットマーカーの 3 ユニットをペアリングしていきます。ここまで操作もマニュアル通りで、LED の表示も親切であり、とくにトラブルなくセットアップが終わりました。NOLO のモバイルモードは Android の USB Host 機能を使っていますので、ヘッド

^{*8} User Guide — NOLO - Motion Tracking for VR <https://www.nolovr.com/user-guide-en>

セットマーカーと Android を付属の USB ケーブルで繋ぎます。Gear VR のヘッドセットと同様の仕組みですが、NOLO には Gear VR のような統合管理を担当するコンパニオンアプリがないため、このあたりから操作難易度が上がってしんどくなっています。

NOLO Tetris^{*9}というアプリが公式サイトからダウンロードできる他、Google Play でも公開されているようなので試して見ましょう。



▲図 4.11 ヘッドセットマーカー接続時の確認ダイアログ（左）と NOLO Tetris のプレイ画面（右）

手順としては以下の通りです。決して少なくない上に、とくにヘッドセットマーカーを USB で接続時に表示されるダイアログの OK がタップしづらくてたまりません（図 4.11-左）。

1. ベースステーションの電源を入れる
2. コントローラの電源を入れる
3. ヘッドセットマーカーを VR ゴーグルにセットする
4. スマートフォンとヘッドセットマーカーを専用の USB ケーブルで接続し、USB Host 機能の確認ダイアログで OK をタップする
5. スマートフォンを VR ゴーグルにセットする

^{*9} NoloTetris - Google Play <https://play.google.com/store/apps/details?id=com.lyrobotix.nolodaydream>

6. トラッキングがうまくいっていなかったら 2. あたりからやりなおし

空中に浮かんでいるブロックから好きなを選んで盤面に近づけると、通常のテトリスのようにハマリこみます（図 4.11-右）。ちゃんと横ラインを消すと消えるので確かにテトリスの模様です。

トラッキングがうまく行く勝率は 3 割ぐらいで、一度うまくいくとその後しばらくは安定します。ただし、そもそもすべてがうまくいってる状態でもヘッドセットもコントローラもプルプルと震え^{*10}、数秒に 1 回程度、10cm から 30cm ほどスイーツと滑ります。これは厳しい。

他のアプリも試して見ます。もう一つの宇宙戦争っぽいアプリはコントローラが動かず、SDK for Android Java のサンプルアプリはスクリーンショット例のようにきちんとログは出してくれず、正常に動いていないように見えました。Forum を探しているうちに見つけた NOLO GUN というアプリは動くには動いたのですが、銃をもったり弾をこめたりする操作性が悪すぎてとても辛かったです。

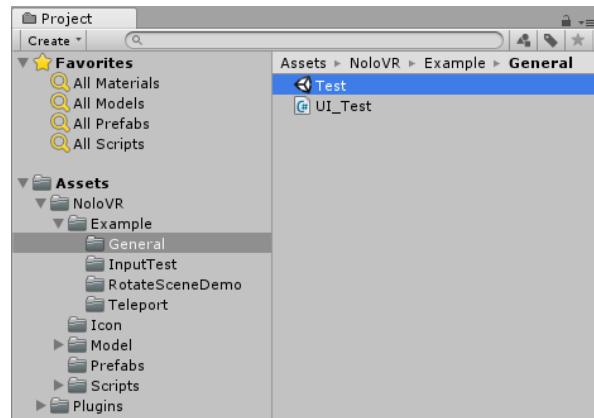
NOLO + Daydream

さて、そんなプルプルにもスイーツにもめげずに、Daydream で動くか試してみました。空中や水中で浮いてるようなケースに限定すればそれなりに楽しめる可能性がありますからね！ 基本的には GitHub 上で公開されている NOLO-Unity-SDK の「Get Started」^{*11} の手順の通りです。この際に、Daydream コントローラは使わないため、前章で述べた「VR Entry Flow」の無効化設定は ON にしておくことをオススメします。

Unity のバージョンは 2017.2.0f3 で確認しています。NoloVR_SDK_1.1.7_20171017.unitypackage をダウンロードしてインポートします。

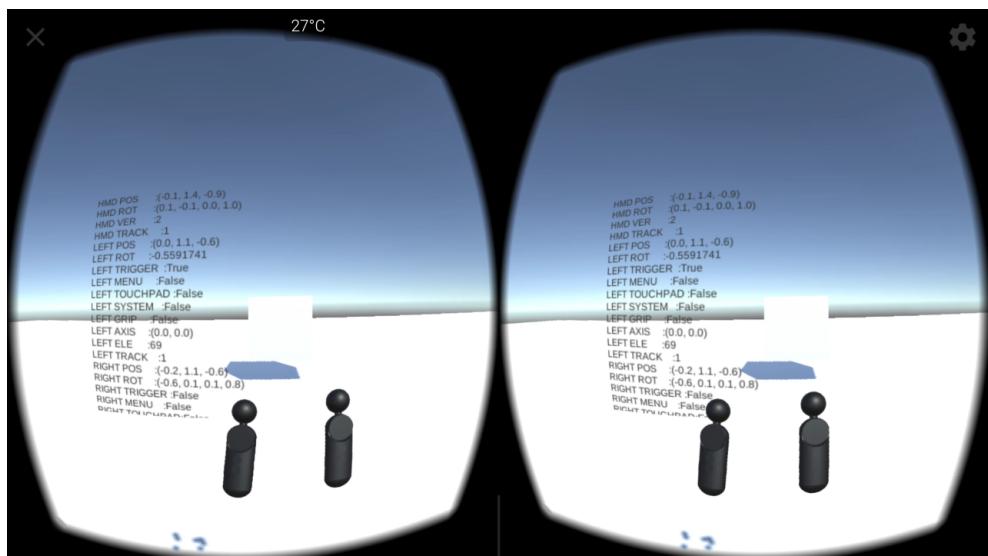
^{*10} ヘッドセットの位置認識がプルプル震えるとどうなるかというと、VR 空間上で世界がプルプルと震えます。SaGa2 の非常階段から最下層みたいな感じです。

^{*11} NOLO-Unity-SDK/GetStarted.md https://github.com/NOLOVR/NOLO-Unity-SDK/blob/master/Docs/en_us/GetStarted.md



▲図 4.12 SDK 内 Example フォルダ配下の Test サンプルシーンを選択

SDK 内、Exapmple フォルダ配下にはいくつかのサンプルシーンが含まれています。ひとまず SDK としてコントローラやヘッドセットマーカーの空間上の座標や向き、ボタンの状態などの情報が表示される、「NoloVR > Example > General > Test」シーン（図 4.12）を試してみます。



▲図 4.13 Test シーンの動作画面

前述の NOLO Tetris と同様に、プルプル揺れたりスイーツと水平方向に滑る現象は（残念ながら）とくに変わりはありませんでしたが、各ユニットの座標・傾き、ボタンの状態はきちんととれていることが分かりました（図 4.13）。首振り制御の競合などが懸念された Daydream としての動作もとく問題ないようです。API の動作が素直だと何かを作ってみようという気になりますね。

そんなわけで将来の猶予はあまりありませんが、可能性を感じました。決して一般ユーザにはオススメはできませんが、機会があればちょっとだけ試す分には面白いデバイスだと思います^{*12}。

^{*12} 相性問題が多くセットアップ難易度が高すぎるとか、不安定ですぐ落ちるとか、ハック口がまったく開いていないとか、そういうデバイスではないということです。ガジェットのクラウドファンディング結果としては満足しています。

あとがき

youten です。FGO に満足してきたのでガルパに手を出して、全バンドランク 20 にしてついでにアズールレーンに手を出しました。Live2D の力強さを感じます。

Daydream View がやっと日本に来る！ ということで Daydream 特集号になりました。4 月から（正確には 6 月末から）はじまった無職生活ですが、何か名刺になりそうなアプリを作ろう、その経験の中でもう少し価値のある具体的なコードが載る本を…と考えていたのですが、まったくそんな感じにはなりませんでした。Publishing（アプリ品質基準ルールと Play 審査の話）まわりも全く書けませんでした。今年度も半年すぎましたので、下半期は本気出していこうと思います！

NOLO はなかなか将来の厳しそうな子でした。ARKit/ARCore ポジショントラッキングがきっと仇をとってくれると思います。それではまた、どこかでお会いできましたら。

Twitter: @youten_redo

Web: ReDo -Refrigerator Door- れいぞうこのドア <http://greety.sakura.ne.jp/redo/>

VR コンテンツのつくりかた 2

2017 年 10 月 21 日 技術書典 3 版 v0.9.0

2017 年 10 月 23 日 v0.9.1

著 者 youten

(C) 2017 ReDo れいぞうこのドア