

1). □□□□□□□□□□

□□□□□□□□□□□□□□□□



1-1). TODO: □□□□□□□□

- TODO: □□□□□□

2). 開発者(Developer)環境

RuntimeSample(開発者)環境

2-1). 環境

Unity(開発者)環境

- [Runtime](#) (開発者)
 - 開発者環境
 - 開発者環境MVP (Model View Presenter)開発者環境Presenter開発者環境
- [Sample](#) (開発者)
 - UI開発者環境
 - 開発者環境
 - 開発者環境MVP (Model View Presenter)開発者環境

Runtime/
├─ Model/
└─ View/

Samples/Scripts/
├─ Define/ (enum)
├─ Generated/ (OpenAPI)
├─ LifetimeScope/ (VContainerDI)
├─ Model/ (MVPModel)
├─ Parameter/ (開発者環境)
├─ Presenter/ (MVPPresenter)
├─ Repository/ (開発者環境 - 開発者環境)
└─ View/ (MVPView)

Runtime (開発者)

Version (開発者)

 PLATEAU.SNAP.App 0.0.1

Support (開発者)

 Unity 2022.3.44f1 or higher

Dependencies (開発者)

 PLATEAU.SNAP.Server 0.0.1  AR Foundation 5.1.5  Google ARCore XR Plugin 5.1.5  ARCore Extensions 1.22.3
 UniTask 2.5.10  R3 1.2.9  Geometry 0.0.5  Mesh2d 0.0.9  Triangulation 0.0.8

Dependencies (□□□□□□□)

2-2). Installation (□□□□)

Installation (Runtime)

- ```
{
 "dependencies": {
 "jp.synesthesias.plateau.snap": "https://github.com/Synesthesias/PLATEAU-SNAP-App.git?path=SnapForUnity/Assets/Synesthesias.Snap",
 "com.cysharp.r3": "1.2.9",
 "com.cysharp.unitask": "2.5.10",
 "org.nuget.r3": "1.2.9",
 "com.google.ar.core.arfoundation.extensions": "https://github.com/google-ar/arcore-unity-extensions.git#1.47.0",
 "com.ishape.geometry": "https://github.com/iShapeUnity/Geometry.git#0.0.5",
 "com.ishape.mesh2d": "https://github.com/iShapeUnity/Mesh2d.git#0.0.9",
 "com.ishape.triangulation":
 "https://github.com/iShapeUnity/Triangulation.git#0.0.8"
 },
 "scopedRegistries": [
 {
 "name": "OpenUPM",
 "url": "https://package.openupm.com",
 "scopes": [
 "com.cysharp",
 "org.nuget"
]
 }
]
}
```

```
PackageManager\Sample\src\main\AndroidManifest.xml
Package/manifest.json
```

```
{
 "dependencies": {
 "com.unity.addressables": "1.22.3",
 "com.unity.localization": "1.5.4",
 "com.unity.textmeshpro": "3.0.6",
 "jp.hadashikick.vcontainer": "https://github.com/hadashiA/VContainer.git?
path=VContainer/Assets/VContainer#1.16.8",
 "org.nuget.polly": "8.5.2"
 }
}
```

## 2-3). Localization

- Localization
- - Edit > Project Settings > Localization ( ) > Create
  - Add Locale
  - Japanese (Code > ja)
  - English (Code > en)
  - Add Locales

## 2-4). Addressables

- Addressables
- Addressables
- Addressables
  - Window > Asset Management > Addressables > Groups > Create Addressables Settings > Create Addressables Settings
- Addressables Project View Inspector > Addressables
  - Group
  - Default Local Group

Assets/Samples/Synesthesias PLATEAU SNAP/< >/Sample App/Addressables

## 2-5).

ScriptableObject

- EnvironmentDevelopment ( )

Assets/Samples/Synesthesias.Snap/Resources/Environment/EnvironmentDevelopment.asset

- EnvironmentRelease (環境リリース)

Assets/Samples/Synesthesias.Snap/Resources/Environment/EnvironmentRelease.asset

## 環境リリースの作成

- Project View環境リリースの作成
- Create > Synesthesias > Snap > Sample > EnvironmentScriptableObject

## 環境リリースの登録

- Project ViewRootLifetimeScopeprefabEnvironment Scriptable Object 環境リリースの登録  
ScriptableObject環境リリースの登録
- 環境リリース Environment Development (環境リリース) 環境リリース

Assets/Samples/Synesthesias.Snap/Resources/VContainer/RootLifetimeScope.prefab

## 2-6). ARCore環境リリース (iOS)

- Unity環境リリースの作成
- Edit > Project Settings
- XR Plug-in Management > iOS環境リリースの作成
  - Apple ARKit 環境リリースの作成
- XR Plug-in Management > ARCore Extensions
  - iOS Support Enabled環境リリースの作成
  - Geospatial環境リリースの作成

## ARCoreAPI環境リリースの作成

- API環境リリースの作成git環境リリースの作成
- API環境リリースの作成環境リリースの作成
- API環境リリースの作成
  - Android: Keyless環境リリースの作成
  - iOS: API Key環境リリースの作成
- API環境リリースの作成環境リリースの作成.gitignore環境リリースの作成

/ProjectSettings/ARCoreExtensionsProjectSettings.json

## ARCoreAPI環境リリースの作成

- 環境リリースの作成ARCoreAPI環境リリースの作成
  - <https://developers.google.com/ar/develop/authorization?hl=ja&platform=unity-arf#api-key-unity>

- Project Settings > XR Plugin-in Management > ARCore Extensions
- iOS Authentication Strategy API Key
  - Authentication Token
  - iOS API Key

## 2-7). API

- (Unity) API
  - <https://github.com/Synesthesias/PLATEAU-SNAP-Server>
- API API

## API

- API git
- API
- API

## .gitignore

- git
  - Assets/Resources/GitIgnore
- .gitignore
- (gitignore)

Assets/Resources/GitIgnore

## API

- Assets/Resources/GitIgnore
- Create > Synesthesias > Snap > Sample > ApiKeyScriptableObject
- ScriptableObject API
  - End Point
    - URL
  - Api Key Type
    - API
    - Bearer, X-API-Key
  - Api Key Value
    - API
- Project View ScriptableObject Api Configuration ScriptableObject &

## 2-8). iOS



### 3). 貢献者(Contributer)について

- 貢献者(Contributer)について
- 貢献者(Contributer)の役割
- [Wiki](#) 貢献者(Contributer)について

#### 3-1). 環境

- iOS環境
- Mac環境
- Unity Windows環境

#### 3-2). 環境構築

Unity Samples~ 環境構築(環境構築)について [Unity Samples](#) 環境構築(環境構築)について

#### Windows環境構築

Windows環境構築

```
git config core.symlinks true
```

Mac環境構築

#### 3-3). API環境構築(環境構築)

- API環境構築
- API環境構築

```
// .NET環境構築
brew install dotnet
```

```
// 環境構築
cd path/to/PLATEAU-SNAP-App
```

```
// .NET環境構築(.config/dotnet-tools.json環境構築)
dotnet new tool-manifest
```

```
// .NET環境構築(docfx環境構築(dotnet-tools.jsondocfx環境構築))
dotnet tool install docfx
```

```
// DocFx環境構築(環境構築)
vi .config/dotnet-tools.json
```



```
// ـــــــــــــــــــــــــــــــــ.NETــــــــــــــــ
dotnet tool restore

// DocFxــــــــــــــــ(ــــــــــــــــــــــــــــــــdocsــــــــــــــــــــــــــــــــ)
dotnet docfx init -o docs

// APIــــــــــــــــــــــــcspjــــــــــــــــ(ــــــــ)
vi docs/docfx.json
```

## DocFxــــــــــــــــــــــــــــــــ

ــــــــــــــــــــــــAPIــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــــ 2.78.2 ــــــــ 2.75.3 ــــــــــــــــــــــــــــــــــــــــ

## DocFxــــــــــــــــــــــــــــــــ(ــــــــــــــــ)

```
// ــــــــــــــــ
Name (mysite): Synesthesias.Snap API Document

// APIــــــــــــــــــــــــــــــــ
Generate .NET API documentation? [y/n] (y): *ــــــــــــــــــــــــــــــــ

// Unityــــــــــــــــــــــــــــــــ(cspjــــــــــــــــ)
.NET projects location (src): ./SnapForUnity

// ــــــــــــــــــــــــــــــــ
Markdown docs location (docs): *ــــــــــــــــــــــــــــــــ

// ــــــــــــــــــــــــــــــــ
Enable site search? [y/n] (y): *ــــــــــــــــــــــــــــــــ

// PDFــــــــــــــــــــــــــــــــ
Enable PDF? [y/n] (y): *ــــــــــــــــــــــــــــــــ

// ــــــــــــــــ
Is this OK? [y/n] (y): *ــــــــــــــــــــــــــــــــ
```

## APIــــــــــــــــــــــــــــــــcspjــــــــ

| docs/[docfx.json](#)↗

```
{
 "$schema":
```

```
"https://raw.githubusercontent.com/dotnet/docfx/main/schemas/docfx.schema.json",
 "metadata": [
 {
 "src": [
 {
 "src": "../SnapForUnity",
 "files": [
- "**/*.csproj"
+ "Synesthesias.Snap.Generated.Sample.csproj",
+ "Synesthesias.Snap.csproj",
+ "Synesthesias.Snap.Sample.csproj"
]
 }
],
 "dest": "api"
+ "allowCompilationErrors": true
 }
]
}
```

### 3-4). APIドキュメントを生成する

APIドキュメントを生成する \**.csproj* ファイルを指定する

UnityプロジェクトのUnityプロジェクトのディレクトリにIDEをインストールする

UnityプロジェクトのUnityプロジェクトのディレクトリに \**.csproj* ファイルを指定する

```
// .NETをインストール
brew install dotnet

// 作業ディレクトリを移動
cd path/to/PLATEAU-SNAP-App

// .NETをインストール
dotnet tool restore

// 不要なファイル削除
rm -rf docs/_site docs/api docs/docs

// ymlファイルの生成
dotnet docfx metadata docs/docfx.json

// htmlファイルの生成
dotnet docfx build docs/docfx.json
```

```
// pdf to yaml/html
dotnet docfx docs/docfx.json --serve --port <port>
```

## 3-5). OpenAPI Generator Client

- OpenAPI Generator Client [spec.json](#) Client
- OpenAPI Generator Client

```
// openapi-generator
brew install openapi-generator
```

```
// OpenAPI
cd path/to/OpenAPI
```

```
//
rm -rf ./Generated.Client/
```

```
//
openapi-generator generate -i spec.json -g csharp -c config_client.json -o ./Generated.Client
```

## README.md

- `Generated.Client` README.md Dependencies

### ## Dependencies

- [Json.NET](<https://www.nuget.org/packages/Newtonsoft.Json/>) - 13.0.2 or later
- [System.ComponentModel.Annotations](<https://www.nuget.org/packages/System.ComponentModel.Annotations>) - 5.0.0 or later

## README.md 2

- Json.NET (Newtonsoft.Json)
  - 13.0.2
- System.ComponentModel.Annotations
  - 5.0.0

using Polly

## openupm-cli

- `Polly` をインストールする

```
// npx
brew install npx
```

```
//
// (openupm-cli)
npx openupm-cli search Polly
```

```
//
npx openupm-cli info org.nuget.polly
```

- `npx openupm-cli search Polly` で `org.nuget.polly` を検索
- `npx openupm-cli info org.nuget.polly` で `org.nuget.polly` のバージョン 8.5.2 を確認

## manifest.json

- `packages-lock.json` を生成する
- `packages-lock.json` と `manifest.json` を比較する
- `Polly` をインストール

```
{
 "dependencies": {
 "org.nuget.polly": "8.5.2"
 }
}
```

## packages-lock.json

- `packages-lock.json` を生成する
- `org.nuget.polly` の `README.md` の `Dependencies` の `Annotation` を `packages-lock.json` に追加

```
{
 "dependencies": {
 "org.nuget.polly": {
 "version": "8.5.2",
 "depth": 0,
 "source": "registry",
 "dependencies": {
 "org.nuget.polly.core": "8.5.2"
 }
 }
 }
}
```

