Unity利用規約

# まえがき

## 参考ページ

[Unity開発者が複数人で開発を進める上で覚えておくと幸せになる９つの事](http://tsubakit1.hateblo.jp/entry/20140613/1402670011#プロジェクト設定で注目すべきポイント)

## 基礎知識

# 目次

1. [設定](#_1.設定)
2. [Assets内フォルダ構成](#_2.Assets内フォルダ構成)
3. [ファイル、フォルダの上書き/名前の変更/移動](#_ファイル、フォルダの上書き/名前の変更/移動)
4. [Gitを使用する時の注意](#_Gitを使用する時の注意)
5. テスト/統合
6. [当たり安定(プログラマ向け)](#_当たり判定(プログラマ向け))

# 1.設定

[メニュー] -> [Edit] -> [Project Settings] -> [Editor] を開いて、[Asset Serialization]を[Force Text]に設定されている事を確認してください。

# 2.Assets内フォルダ構成

1. プロジェクト名 : ゲーム用フォルダ
   1. Audios : BGMやSEなどのサウンドデータを置く
   2. Materials : メッシュやパーティクルのマテリアルを置く
   3. Meshs : fbx等メッシュデータを置く
   4. PhysicsMaterials : 物理用マテリアルを置く
   5. Prefabs : Object（.prefab）を置く
      1. Resources : Resources.Loadで読み込む.prefabを置く
   6. Scenes : Title,StageSelect等の.sceneファイルを置く
   7. Scripts : スクリプトを置く
      1. Editor : エディター用のスクリプトを置く
   8. Textures : テクスチャを置く
   9. Users : 各チームメンバーがテスト用に自由に使っていいフォルダ。

各フォルダ内に関しては更に使用するオブジェクトの種類でフォルダ分けする事

例 : Audios-PlayerCharacter-Jump.mp3, Textures-Enemy-Enemy1icon.jpg,

Prefabs-Resources-PlayerCharacter-Bullet.prefab等

## (プログラマ)Resourcesフォルダについて

Resources.Load関数で読み込むファイルを置きます。

親フォルダがどこでもアクセスできるので、同名のファイルがあった場合どちらが読み込まれるかわからないためさらにフォルダ分けするなど注意してください。

# ファイル、フォルダの上書き/名前の変更/移動

参考 : [Unityのプロジェクトは何故壊れるのか。原因と対策](http://madnesslabo.net/utage/?page_id=5640)

## 上書き

ファイルを上書きする場合、Unity外（エクスプローラー）で上書きしてください。

Unity上に同名ファイルをドラッグアンドドロップすると新しいファイルとして読み込まれてしまいます。

## 名前の変更

名前を変更する場合、Unity上で名前を変更してください。

Unity外で名前を変更するとファイルがなくなったと判断されます。

### (プログラマ)クラス名の変更

Unity上でスクリプト名を変更した後、クラス名を変更してください。

## 移動

ファイル、フォルダを移動する場合、Unity上で移動してください。

# Gitを使用する時の注意

## コミット

ファイルを追加した場合、同名の.metaファイルが生成されるので一緒にコミットしてください。

### .metaファイルについて

*metaファイルは、Unityプロジェクトにインポートした際に生成され、プロジェクト上でのファイルの振る舞いについての情報を持つ。　例えば、Textureの圧縮フォーマット、Modelの設定等々。*

# 5.テスト/統合

## フォルダ/命名規則

テスト用のスクリプト/シーン/オブジェクトを作成する場合、Usersフォルダ以下の自分の名前フォルダ内に自由に作成してください。

その場合、ファイル名の前もしくは後ろにTestを必ずつけてください。

## スクリプトの場合

Test名前空間でテスト用スクリプトを囲んでください。

例：namespace Test

{

class SceneTest～

}

# 当たり判定(プログラマ向け)

詳しくは<https://docs.unity3d.com/ja/current/Manual/Layers.html>を参照してください。

## コード

衝突判定はレイヤーで行ってください。

例:

// Start()時

int layerMaskEnemy = LayerMask.NameToLayer(“Enemy”);

// OnCollider～(Collider collider)時

if(collider.gameobject.layer == layerMaskEnemy)