

第1章

Gitでバージョン管理

～今すぐ使う編～

必須ではない。だが、最強無敵の味方Git。
SourceTreeを使って、最短で今すぐ使う！

今回の目的

- とにかくGitを使う！
難しいことは全カスルー、まずは**Gitの力を得よう！**
- 最低限の設定だけ！
Mac+Unityの開発を前提でやるよ！
- シンプルな環境で！
ローカルPCで**自分1人**でGit管理する前提でやるよ！

こうなる！

1. Gitでバージョン管理する前

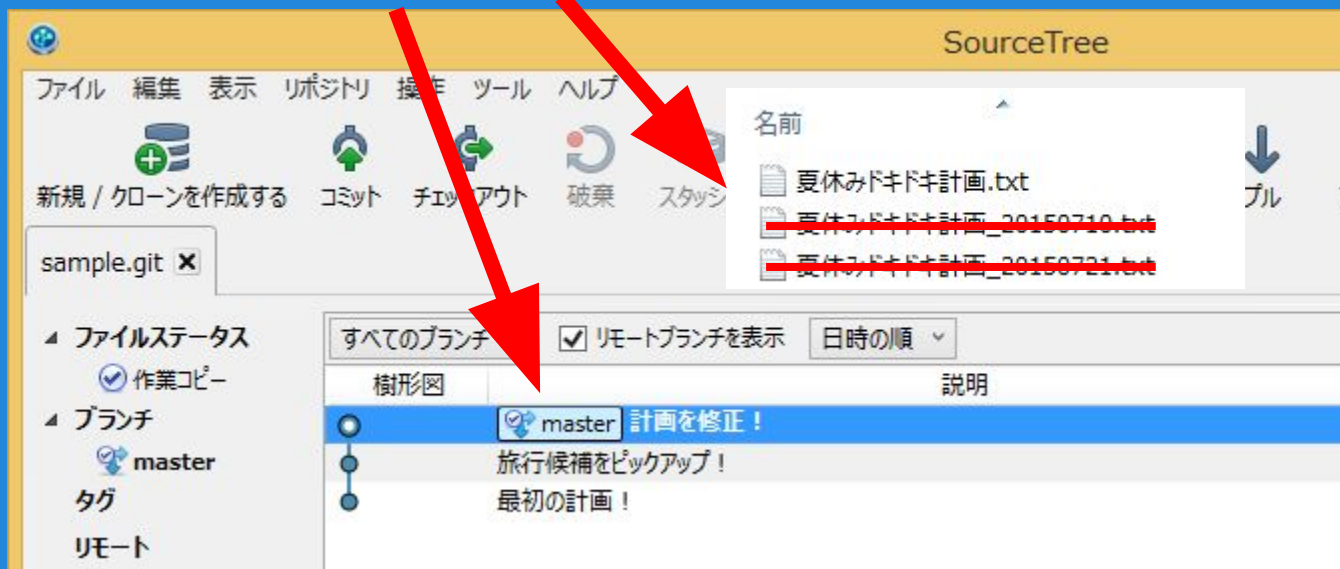
2. Gitでバージョン管理した後

悲惨な日付管理(T_T)

名前

夏休みドキドキ計画.txt
夏休みドキドキ計画_20150710.txt
夏休みドキドキ計画_20150721.txt

Gitを使いやすくしたソフトがSourceTree



SourceTreeのインストール

1と2だけあれば、使えます。

1. Atlassianのアカウントを作る(無料)。
2. Gitは、SourceTree内蔵を使う。
3. ユーザー情報は、自由(作成者名として残る)。
4. Mercurialは、不要。
5. SSH設定は、PuTTY/Plinkを使う。
6. SSHキーは、設定しない。

Gitの初期設定

● リポジトリの作成

作業フォルダにリポジトリを作って、Git管理できるようにする。

- Unityの新規プロジェクトを作成。
- [新規 / クローンを作成する] → リポジトリを作成。

● gitignoreの設定

ログや一時ファイルなど**不要なファイル**を管理対象外にする。

<https://github.com/github/gitignore>「github gitignore」でぐぐればOK。

- gitignore/Unity.gitignore
- gitignore/Global/OSX.gitignore

Gitの初期設定



gitignore

それぞれのリポジトリで設定する。

- gitignore/Unity.gitignore
- [設定] → [詳細]タブ → リポジトリ固有の無視リスト。



gitignore_global

共通の設定で、全てのリポジトリに反映される。

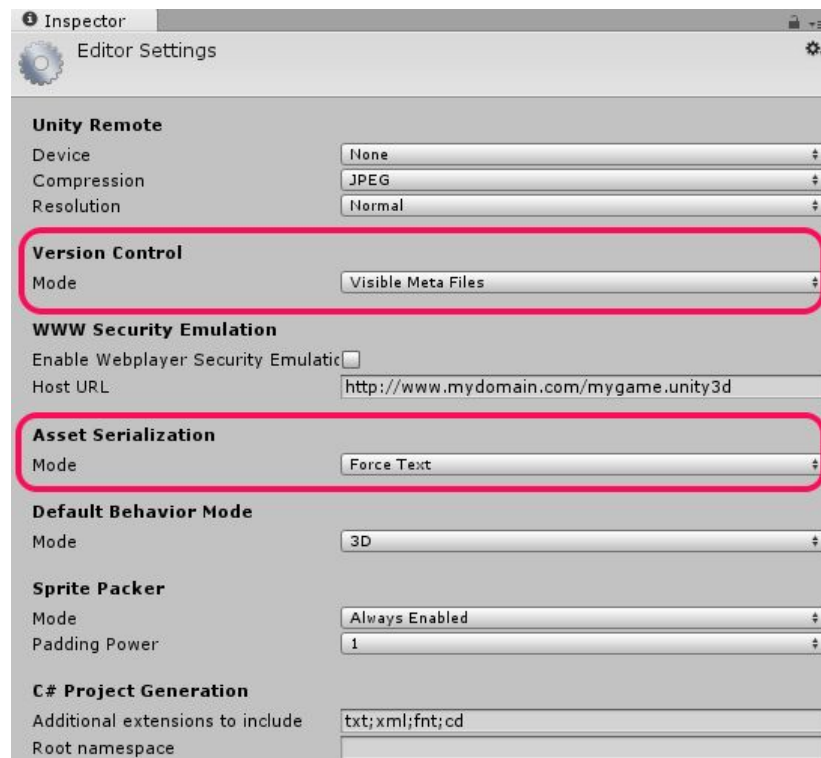
- gitignore/Global/OSX.gitignore
- [ツール] → [オプション] → [Git]タブ
→ グローバル無視リスト → ファイルを編集。

Unityの初期設定

● **Version Control**
Mode: **Visible Meta Files** に設定。

● **Asset Serialization**
Mode: **Force Text** に設定。

初めてのコミットをする前に、
必ず設定しておこう！



基本操作

● ひたすらコミット

現時点のファイルをGitに記録管理させる。コメントも書ける。

- 記録した時点のファイルに、いつでも戻せる。
- 前回コミットに対して、変更点も見れる。
- **コミットする直前に、必ずUnityでSave Sceneせよ！**

● 過去のファイルに戻す

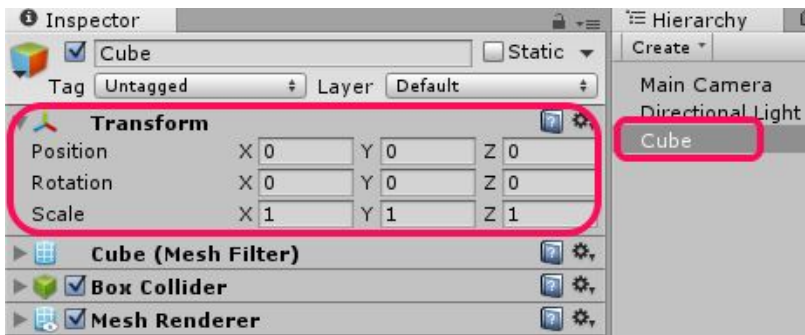
[ログ]タブ → 好きなコミットをダブルクリック。

● 過去のファイルを取得する

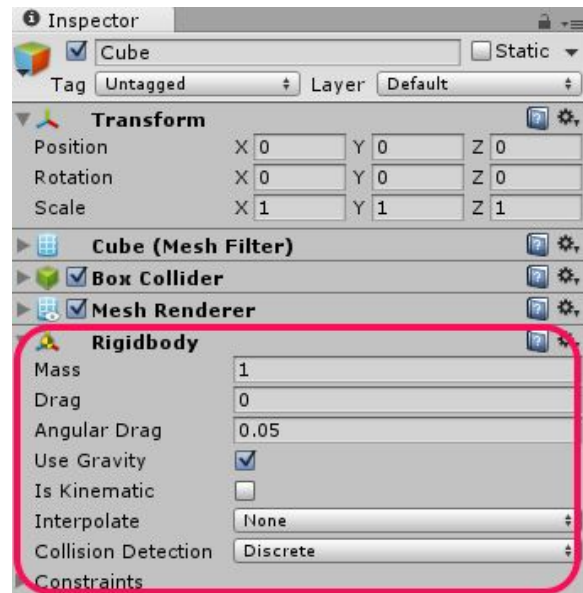
[ログ]タブ → 好きなコミットを右クリック → アーカイブ。

UnityでGitを試す (シーンの編集)

1.Cubeを作成して、コミットする。



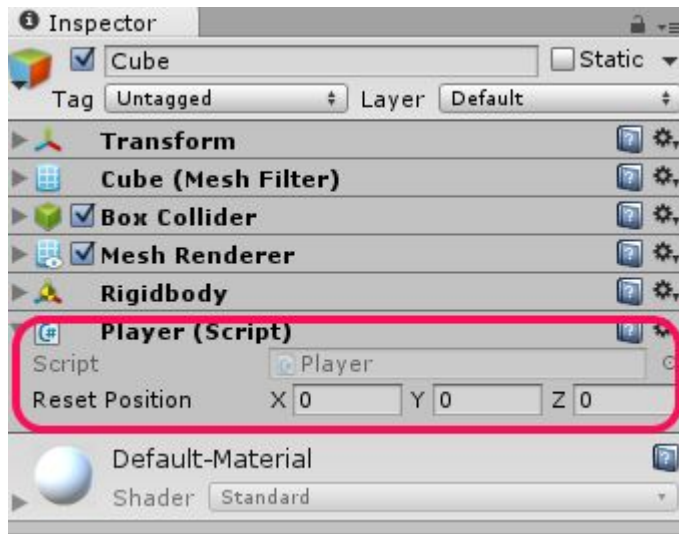
2.CubeにRigidbodyをアタッチして、コミットする。



シーンの保存は忘れずに！

UnityでGitを試す (コードの編集)

3. CubeにPlayerスクリプトをアタッチして、コミットする。



4. ジャンプを実装して、コミット。

```
1 using UnityEngine;
2
3 public class Player : MonoBehaviour
4 {
5     [Tooltip("リセット後の位置")]
6     public Vector3 ResetPosition;
7
8     private Transform _transform;
9     private Rigidbody _rigidbody;
10
11     [Tooltip("ジャンプする高さ")]
12     public float Jump = 5f;
13
14     private bool _isJump;
15
16     void Awake()
17     {
18         _transform = GetComponent<Transform>();
19         _rigidbody = GetComponent<Rigidbody>();
20     }
21
22     void Update()
23     {
24         _isJump = _isJump || Input.GetMouseButtonDown(0);
25     }
26
27     void FixedUpdate()
28     {
29         if (!_isJump) return;
30         _isJump = false;
31
32         _rigidbody.velocity = Vector3.up * Jump;
33     }
34
35     void OnBecameInvisible()
36     {
37         _transform.position = ResetPosition;
38     }
39 }
```

まとめ

● 今回使ったコマンド

- リポジトリの作成。(init)
- コミット。(add, commit)
- 過去に戻る。(checkout)
- 過去のファイルを取得。(archive)

● バージョン管理システムGitについて

第4回「SourceTreeによるGit入門」の資料を参考にしてください。

第1章クリア

最強無敵のGitが仲間になりました！ やったね！
どんな環境で開発をするときも、必ず使って慣れていこう！

次回は、アmend・ブランチ・リセットを使って、
コミットを修正したり、過去の状態に戻したりするよ！