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

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

L つくよう 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)



Gitを使いやすくしたソフトがSourceTree



SorceTreeのインストール

<u>1と2だけあれば、使えます。</u>

- 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。

- O 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。

<u>https://github.com/github/gitignore</u> [github gitignore]でぐぐればOK。

- O gitignore/Unity.gitignore
- O 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。
- O 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、全てのリポジトに反映される。
- O gitignore/Global/OSX.gitignore
- O [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。



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!

Inspector		
Editor Settings		۵,
Unity Remote		
Device	None	\$
Compression	JPEG	\$
Resolution	Normal	\$
Version Control		
Mode	Visible Meta Files	\$
Enable Webplayer Security Emula Host URL	ttichttp://www.mydomain.com/mygame.unity3d	
Asset Serialization		
Mode	Force Text	;
Default Behavior Mode		
Mode	3D	\$
Sprite Packer		
Mode	Always Enabled	\$
Padding Power	1	\$
C# Project Generation		
Additional extensions to include	txt;xml;fnt;cd	
Root namespace		



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。

- O 記録した時点のファイルに、いつでも戻せる。
- 〇 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。



シーンの保存は忘れずに!

2.CubeにRigidbodyをアタッチして、コミットする。

o Inspector	_	_	_		10		n in the second se
💋 🗹 Cube				-		JSt	atic
Tag Untagged		\$	Layer	Default			
🙏 Transform							1
Position	Х	0	Y	0	Z	0	1.212
Rotation	Х	0	Y	0	Z	0	
Scale	Х	1	Y	1	Z	1	
📋 Cube (Mesh	Fil	ter)					1
😡 🗹 Box Collider	(
🛃 🗹 Mesh Rende	rer	-					
🔉 Rigidbody	20						12 4
Mass	1						
Drag	0						
Angular Drag	0	.05					
Use Gravity	V	1					
Is Kinematic	C]					
Interpolate	1	lone					
Collision Detection	0	iscret	te				
Constraints	25						

UnityでGitを試す(二一^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。 using UnityEngine; public class Player : MonoBehaviour [Tooltip("リセット後の位置")] public Vector3 ResetPosition; private Transform transform; private Rigidbody rigidbody; [Tooltip("ジャンプする高さ")) 11 12 public float Jump = 5f; 13 14 15 16 private bool isJump; void Awake() 17 transform = GetComponent<Transform>(); 18 19 _rigidbody = GetComponent<Rigidbody>(); 20 21 22 E 23 24 25 26 27 E 28 29 30 void Update() isJump = isJump || Input.GetMouseButtonDown(0); void FixedUpdate() if (! isJump) return; isJump = false; 31 32 33 34 35 36 37 38 39 rigidbody.velocity = Vector3.up * Jump; void OnBecameInvisible() transform.position = ResetPosition;



● 今回使ったコマンド

- O リポジトリの作成。(init)
- O コミット。(add, commit)
- O 過去に戻る。(checkout)
- O 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、アメンド・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!