

第6章

Gitでバージョン管理

～リベース編～

我々「今日も順調♪あ、他の人が新しいコミットをしたぞ！早く取り込まないと置いてきぼりに…」
Git「未来からやり直せばよかろう。常に最先端じゃよ。」

今回の目的1

- 最新に対して再コミット (*rebase*)
複数ブランチで運用中の履歴がキレイになります。

自分のブランチで行ったコミットを、別のブランチ(任意のコミット地点)に対して全てコミットし直します。



今回の目的2

- 競合の解決 ([conflict/mergetool](#))
リベースやマージを行った際に、他の人と同じ箇所を変更してコミットし合った場合に競合します。

どちらの変更を適用するか、
または手動で編集して競合を
解決する必要があります。

同じファイルを同時に
変更しないのが安全です。

```
Hunk 1: 行 0-0
0 0      enemy.transform.position = new Vector3(
1 + <<<<<<<<<< 36d3416457366712c4d126f901b7cfc0694a7d2d
1 2      +          x += 2f; ← master側の内容
3 + ++++++
4 +          enemy.tag = "Enemy";
5 +          x += 4f; ← bullet側の内容
6 + +>>>>>>> [add] Bullet
2 7      }
```

今回の完成図

bulletブランチ作成

グラフ	説明
	[update] EditorSettingsをGit用に設定
	Revert "[modify]右に向かってジャンプするように修正"
	Merge branch 'stage1'
	[add] ランダムな位置に敵を配置
	[add] 追従カメラを実装

rebase!!

グラフ	説明
	[update] Stage, Player jump.
	[add] Bullet
	[add] Enemy tag
	[update] EditorSettingsをGit用に設定

グラフ	説明
	[add] Bullet
	[add] Enemy tag
	[update] Stage, Player jump.
	[update] EditorSettingsをGit用に設定

bulletで2コミット
masterで1コミット

bulletが先端になる
(masterの内容を包含)

コミット内容

1. (bullet) Enemyタグ追加。
 - ・Unityでタグ入力
2. (bullet) Bullet実装。
 - ・PrefabをProjectにD&D
 - ・Playerを修正してアタッチ
 - ・EnemySpawnerを修正してタグ設定
3. (master) StageとPlayer修正。
 - ・CubeのRigidbody設定
 - ・Playerの右ジャンプ修正
 - ・EnemySpawnerの敵配置修正

masterにマージして完成

グラフ	説明
	Merge branch 'bullet'
	[add] Bullet
	[add] Enemy tag
	[update] Stage, Player jump.
	[update] EditorSettingsをGit用に設定
	Revert "[modify]右に向かってジャンプするように修正"

bulletブランチの作成 (branch)

1. master(最新コミット)を
右クリックして、
ブランチを選択します。



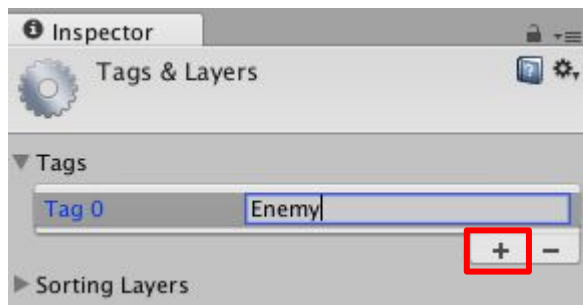
2. bulletブランチを
作成します。



3. bulletがアクティブ
(チェックアウト)

Enemyタグの追加

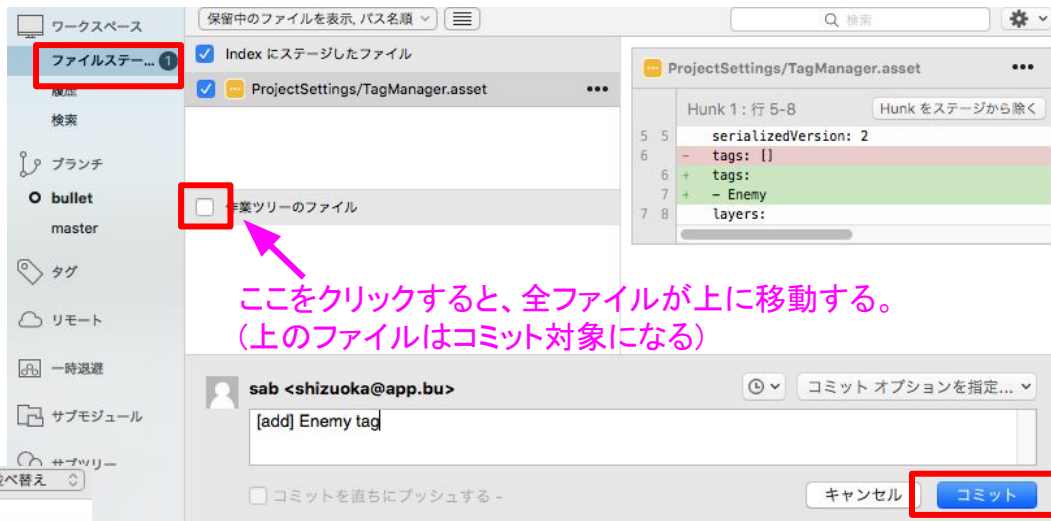
1. Edit → Project Settings → Tag and Layers を選択します。
2. 「+」ボタンをクリックして、Enemyを入力します。



Enemyタグの追加

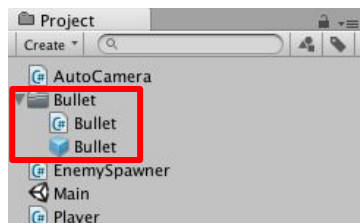
3. “[add] Enemy tag”
とコミットします。

4. bulletが1コミット
先行します。



Bulletの実装 (Bulletの発射)

1. BulletフォルダをProjectにD&Dします。



2. Player.csに2箇所追加します。

```
public Transform Bullet;
```

```
if (!_isJump) return;
```

```
Instantiate(Bullet).transform.position =  
_transform.position;
```

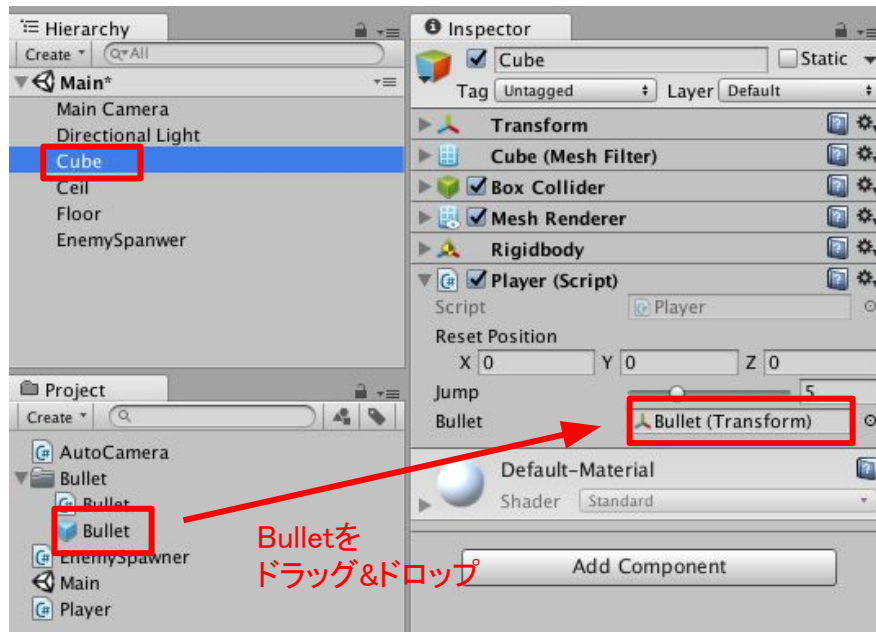
```
12 [Range(3f, 10f)]  
13 public float Jump = 5f;  
14  
15 private bool _isJump;  
16  
17 public Transform Bullet;  
18  
19 void Awake()  
20 {  
21     _transform = GetComponent<Transform>();  
22     _rigidbody = GetComponent<Rigidbody>();  
23 }  
24  
25 void Update()  
26 {  
27     _isJump = _isJump || Input.GetMouseButtonDown(0) || Input.GetM  
28  
29     if (!_isJump) return;  
30     Instantiate(Bullet).transform.position = _transform.position;  
31 }  
32
```


Bulletの実装 (Bulletの発射)

3. Cubeを選択します。
PlayerのBulletに
Bulletプレハブをセットします。

4. プレイ開始します。

クリックしたときに
Bulletが発射されることを
確認しましょう。



Bulletの実装 (Enemyの破壊)

5. EnemySpawnerを修正します。

```
enemy.tag = "Enemy";
```

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class EnemySpawner : MonoBehaviour {
5
6     void Start () {
7         var x = 2f;
8         while (x < 50)
9         {
10            var enemy = GameObject.CreatePrimitive(PrimitiveType.Cube);
11            enemy.transform.position = new Vector3(x, Random.Range(0, 10), 0);
12            enemy.tag = "Enemy";
13            x += 4f;
14        }
15    }
16 }
17 |
```

6. プレイ開始します。

Enemyに衝突すると
破壊されることを確認します。

Bulletの実装

7. “[add] Bullet”
とコミットします。

8. bulletが2コミット
先行します。

Graph showing commit history:

- bullet [add] Bullet [add] Enemy tag
- master [update] EditorSettingsをGit用に設定
- Revert "[modify]右に向かってジャンプするように修正"
- Merge branch 'stage1'

File Explorer showing staged files:

- Index にステージしたファイル
- Assets/Bullet.meta
- Assets/Bullet/Bullet.cs
- Assets/Bullet/Bullet.cs.meta
- Assets/Bullet/Bullet.prefab
- Assets/Bullet/Bullet.prefab.meta

Commit Dialog:

- Commit message: [add] Bullet
- Commit options: [add] Bullet
- Buttons: キャンセル, コミット

Assets/Bullet.meta file content:

```
1 + fileFormatVersion: 2
2 + guid: 7ec2453ee091441c0b281c699b54e0fd
3 + folderAsset: yes
4 + timeCreated: 1479768462
5 + licenseType: Free
6 + DefaultImporter:
7 +   userData:
8 +   assetBundleName:
9 +   assetBundleVariant:
```

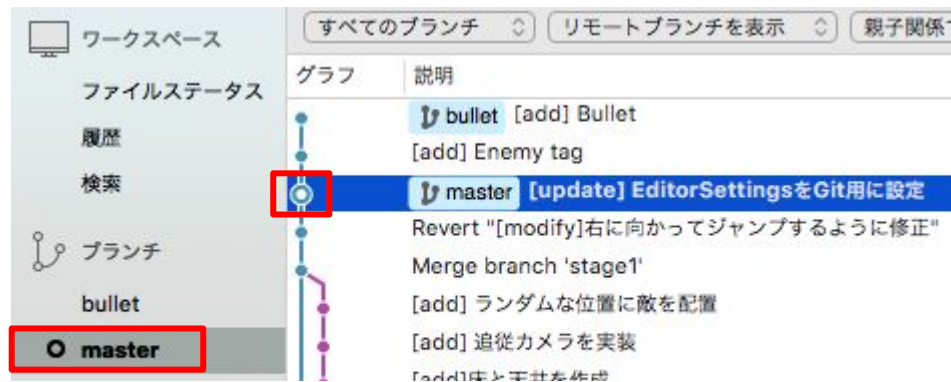
(!)実際は8ファイルです

StageとPlayerの修正

1. masterブランチをチェックアウトします。

以下を想定してみましよう。

- ・自分がbulletブランチ作業してコミット。
- ・他の人がmasterブランチ作業してコミット。



StageとPlayerの修正

2. EnemySpawner.csを修正して、出現数を増やします。

```
4 public class EnemySpawner : MonoBehaviour {  
5  
6     void Start () {  
7         var x = 2f;  
8         while (x < 100) → 50 → 100  
9             {  
10            var enemy = GameObject.CreatePrimitive(PrimitiveType.Cylinder);  
11            enemy.transform.position = new Vector3(x, Random.Range(-3f, 5f));  
12            x += 2f; → 4f → 2f  
13        }  
14    }  
15 }
```

StageとPlayerの修正

3. Player.csを修正して、右ジャンプを実装します。

```
_rigidbody.velocity = new Vector3(4f, Jump);
```

```
28 void FixedUpdate()  
29 {  
30     if (!_isJump) return;  
31     _isJump = false;  
32       
33       
34     _rigidbody.velocity = new Vector3(4f, Jump);  
35 }
```

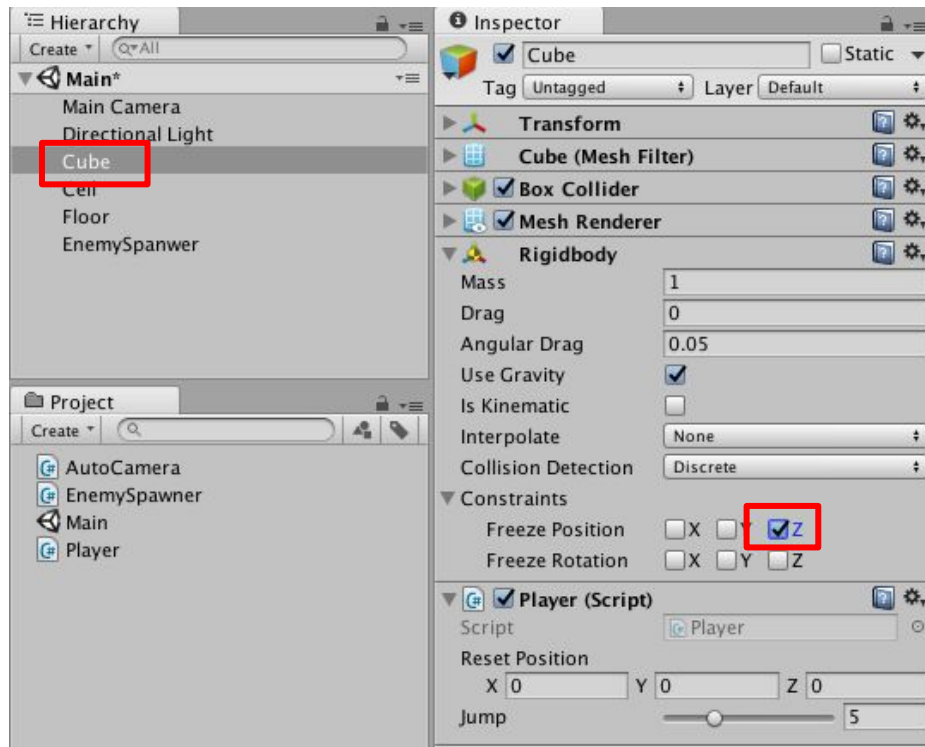
StageとPlayerの修正

4. Cubeを選択します。
RigidbodyのFreeze PositionのZをチェックします。

CubeがZ方向で
移動なくなります。
(Z座標が固定)

5. プレイ開始します。

右方向にジャンプすることを
確認します。



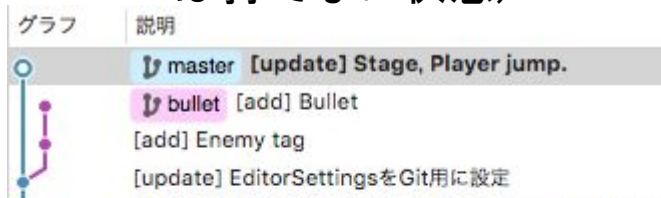
StageとPlayerの修正

6. “[Update] Stage, Player jump.”
とコミットします。

```
Assets/EnemySpawner.cs
Hunk 1: 行 7-9
7 7     var x = 2f;
8 8 -   while (x < 50)
8 8 +   while (x < 100)
9 9     {

Hunk 2: 行 11-13
11 11     enemy.transform.position = new Vector3(x
12 12 -   x += 4f;
12 12 +   x += 2f;
13 13     }
```

7. masterが1コミット
先行します。
(bulletは別次元にいるので
bulletは打てない状態)



sab <shizuoka@app.bu>

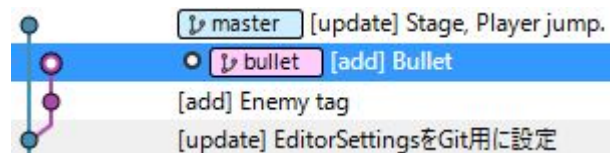
[update] Stage, Player jump.

コミットを直ちにプッシュする -

キャンセル コミット

リベース (*rebase*)

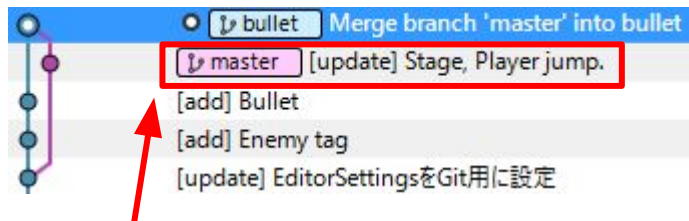
1. bulletブランチをチェックアウトします。



以下を想定してみましょ。

- ・他の人がmasterブランチにコミットした内容を自分のブランチに取り込みたい。(マージで実現可能)
- ・でも、履歴はキレイにしたい。(リベースで実現可能)

マージ



masterの最新コミットをbulletにマージして取込

リベース



masterの最新コミットに対してbulletを再コミットして適用

リベース (*rebase*)

2. masterを右クリックして、リベースを選択します。



bullet(アクティブ): 最新masterに再コミットする
master: 最新コミット (どこにリベースしたいか)

リベース (rebase)

3. 競合が発生します。
masterとbulletで同じ箇所を修正していることが原因です。

Assets/EnemySpawner.cs

```
Hunk 1 : 行 0-0
0 0      enemy.transform.position = new Vector3(x, Random.Range(-3f, 5f));
1 +<<<<<<<< 36d3416457366712c4d126f901b7cfc0694a7d2d
1 2 +      x += 2f;
3 +>>>>>>>
4 +      enemy.tag = "Enemy";
5 +      x += 4f;
6 +>>>>>>> [add] Bullet
2 7      }
```

master側の内容 (自分)

bullet側の内容 (相手)

競合したけど自動解決した箇所

! マークが競合したファイル

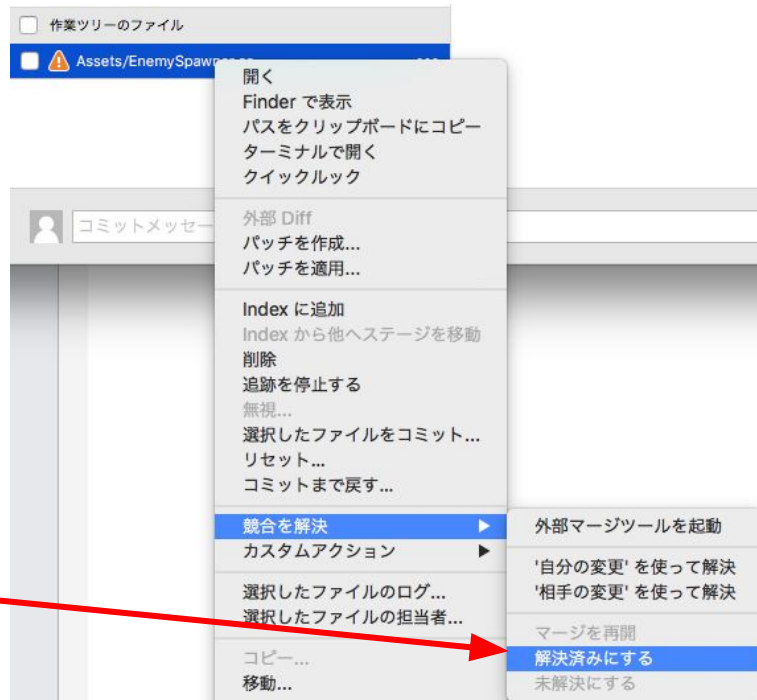
「masterが自分!？」
最新masterに対して、bullet
の2つのコミットをし直してい
る。コミット相手はbullet。

リベース (rebase)

4. 今回は競合を手動で修正します。

```
4 public class EnemySpawner : MonoBehaviour {  
5  
6     void Start () {  
7         var x = 2f;  
8         while (x < 100)  
9             {  
10            var enemy = GameObject.CreatePrimitive(  
11            enemy.transform.position = new Vector3(  
12            enemy.tag = "Enemy";  
13            x += 2f;  
14            }  
15        }  
16    }
```

5. 解決済みにします。



リベース (*rebase*)

6. 操作 → リベースを続ける を選択します。

リベース中の場合のみ、このメニューが
選択可能な状態になっています。

リベース中に競合が発生した場合の対処方法

- ・競合を解決する。
- ・リベースを中止して、実行前に戻す。



リベース (*rebase*)

7. リベースが完了して、履歴が直線になります。(キレイ)

グラフ	説明
	bullet [add] Bullet
	[add] Enemy tag
	master [update] Stage, Player jump.
	[update] EditorSettingsをGit用に設定
	Revert "[modify]右に向かってジャンプするように修正"
	Merge branch 'stage1'
	[add] ランダムな位置に敵を配置

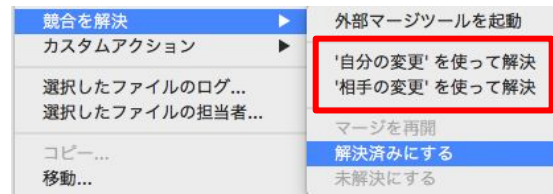
8. プレイ開始します。

Cubeが右にジャンプできて(masterのコミット)、
Bulletも発射できること(bulletのコミット)を確認しましょう！

補足

- 競合からは逃げられない
同じ箇所を変更した時点で競合は発生してしまうため、
リベースだからではなく、マージした際にも発生します。
同じ箇所を変更しないことが一番の予防策です。

- 競合をひとまず対処する
競合ファイルの右クリックから
自分か相手の内容で解決して、
リベースを続け、ひとまず完了させます。
その後、ファイルの内容を確認して手動で修正しましょう。



リベース前にtmpブランチを作成してバックアップしておきましょう。
(コミット修正編をP4参照)

第6章 クリア

最強無敵のGitで複数ブランチで作業しながら最新コミットを取り込んで、履歴をキレイにすることができました。積極的にブランチを活用していきましょう。

次回は、チェリーピックを使って、別のブランチから必要なコミットだけ取り出してみましよう。