

第7章

Gitでバージョン管理 ～チェリーピック編～

我々「このコミットだけを別のブランチに適用したいけど、簡単にできればなあ…。」

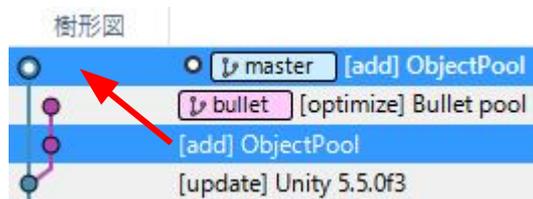
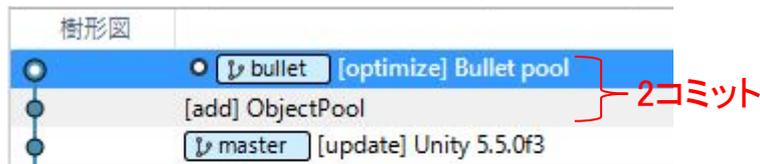
Git「コミットをつまみ食いすれば、お主のものじゃよ。」

今回の目的

- 任意のコミットを別のブランチに適用 ([cherry-pick](#))
過去のコミットや別のブランチのコミットを自分のブランチに適用できます。

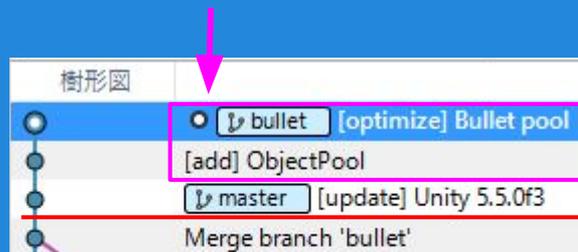
(例)

bulletブランチのコミットをmasterブランチに適用

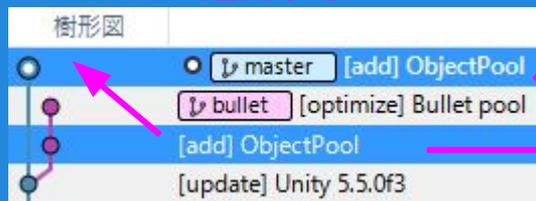


今回の完成図

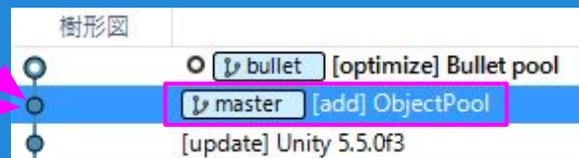
bulletに2コミット作成する



bulletの1つ目のコミットをmasterに適用する



bulletをmasterにリベースする



cherry-pick!!

同じ内容のコミットは消えて、bulletブランチは1コミットだけになる

コミット内容

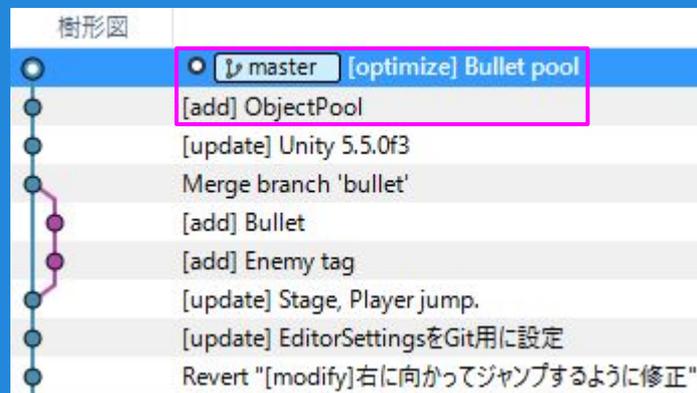
1. [add] ObjectPool

- ObjectPool.csを追加。
(生成したGameObjectを再利用するためのクラス)

2. [optimize] Bullet pool

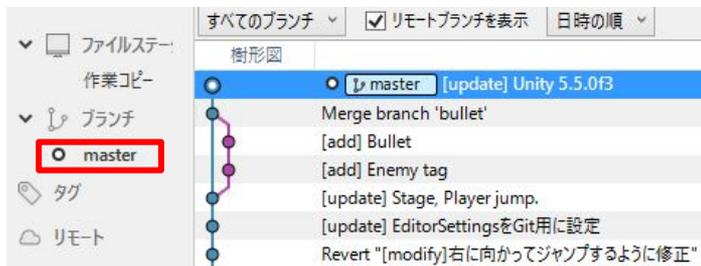
- Player.csでBulletの生成を修正。
- Bullet.csでBulletの破棄を修正。
(ObjectPoolを使ってBulletを再利用する)

masterにマージして完成！

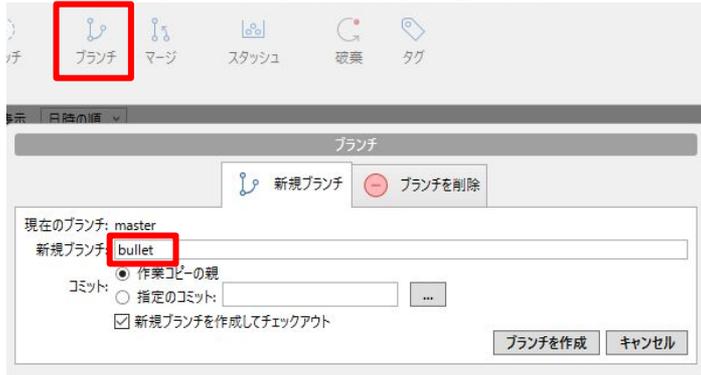


bulletブランチの作成 (branch)

1. masterをチェックアウトします。

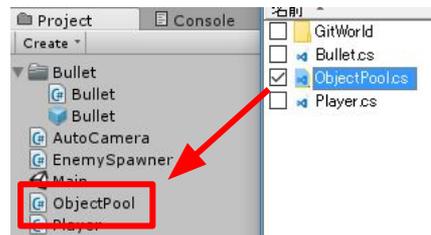


2. bulletブランチを作成します。

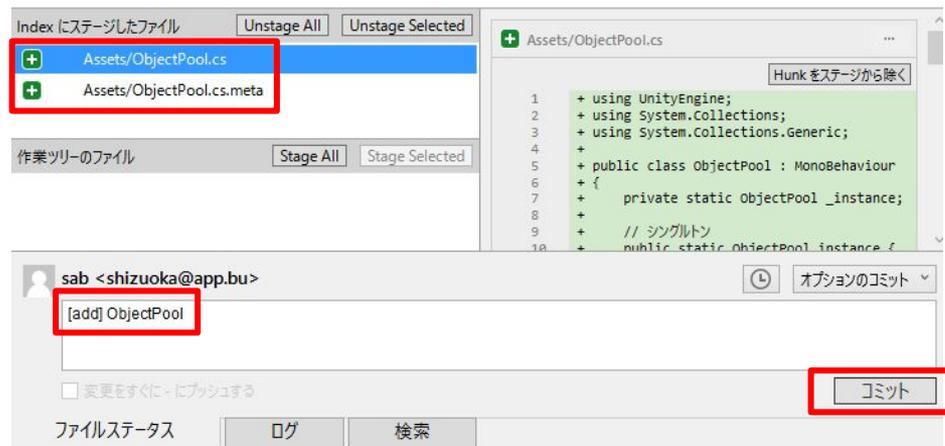


ObjectPool.csの追加

1. ObjectPool.csをProjectにD&Dして追加します。

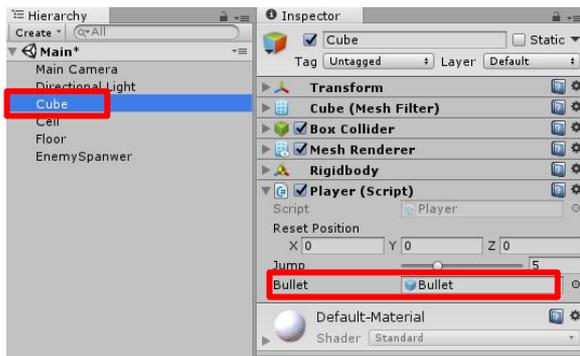


2. “[add] ObjectPool”とコミットします。



Bulletを再利用する 1/2

1. Player.csでBulletの生成を修正します。
2. CubeのPlayerにBulletのプレハブを再設定します。



```
Assets/Player.cs
Hunk をステージ
14 14
15 15     private bool _isJump;
16 16
17 17     - public Transform Bullet;
17 17     + public GameObject Bullet;
18 18
19 19     void Awake()
20 20     {
28 28
29 29         if (!_isJump) return;
30 30
31 31     - Instantiate(Bullet).transform.position = _transform.position;
31 31     + ObjectPool.instance.GetGameObject(Bullet).transform.position = _transform.position;
32 32
33 33     }
34 34     void FixedUpdate()
```

```
public GameObject Bullet;
```

```
ObjectPool.instance.GetGameObject(Bullet).transform.position = _transform.position;
```

ObjectPoolから再利用できるBulletを取得します。

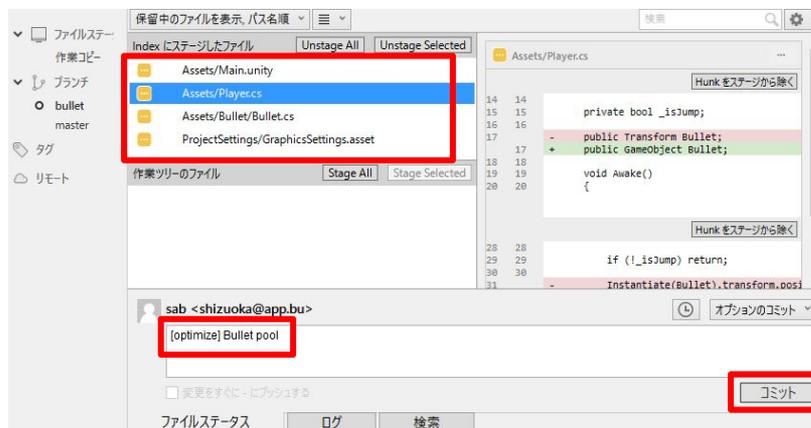
Bulletを再利用する 2/2

3. Bullet.csでBulletの破棄を修正します。

```
ObjectPool.instance.ReleaseGameObject(gameObject);
```

```
Assets/Bullet/Bullet.cs
13 13     if (other.tag != "Enemy") return;
14 14
15 15     Destroy(other.gameObject);
16 16     Destroy(gameObject);
17 17     ObjectPool.instance.ReleaseGameObject(gameObject);
18 18 }
19 19
20 20 void OnBecameInvisible()
21 21 {
22 22     Destroy(gameObject);
23 23     ObjectPool.instance.ReleaseGameObject(gameObject);
24 24 }
```

4. “[optimize] Bullet pool”とコミットする。

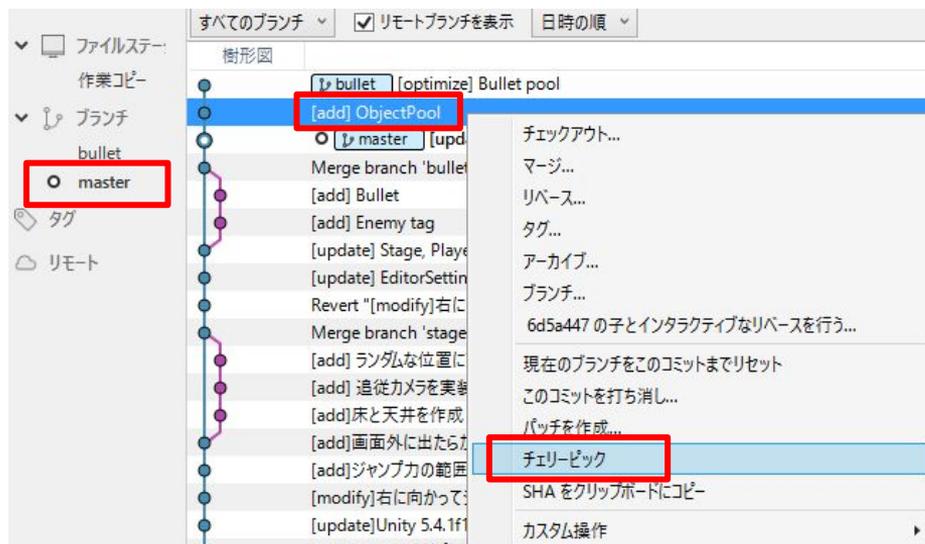


チェリーピック (cherry-pick)

1. masterをチェックアウトし、“[add] ObjectPool”を右クリックしてチェリーピックを選択する。

(以下を想定してみましよう)
bulletの作業は続けたいが、
masterに1つだけ先に
コミットしたい。

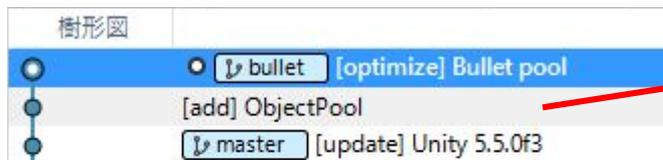
master: 全員の最新ブランチ
bullet: 自分の作業ブランチ



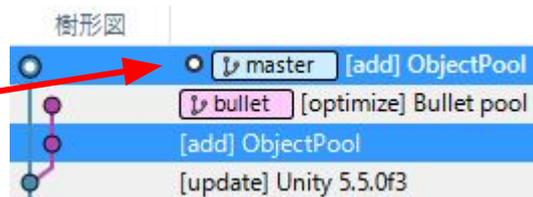
チェリーピック (cherry-pick)

2. masterに”[add] ObjectPool”がコミットされたことを確認します。

チェリーピック前

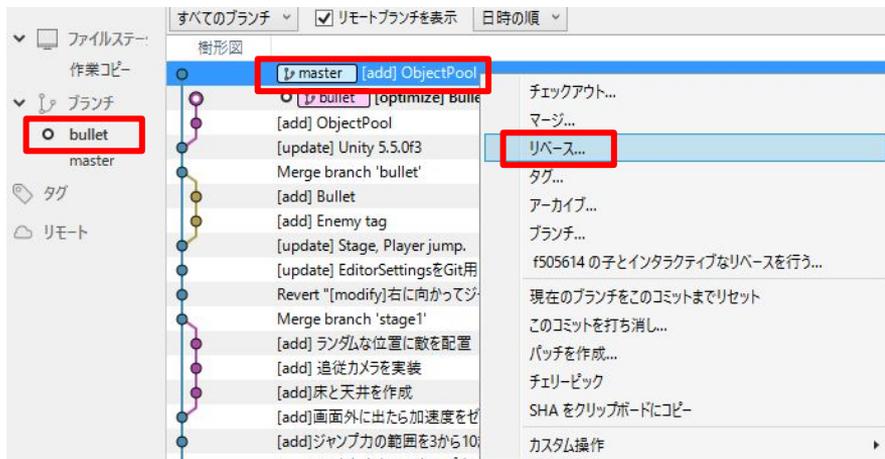


チェリーピック後



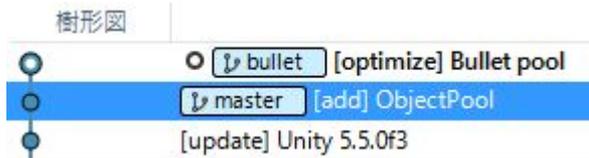
リベース (rebase)

1. bulletをチェックアウトし、masterを右クリックしてリベースを選択します。



最新のmasterに対して
bulletの2コミットを再適用します。

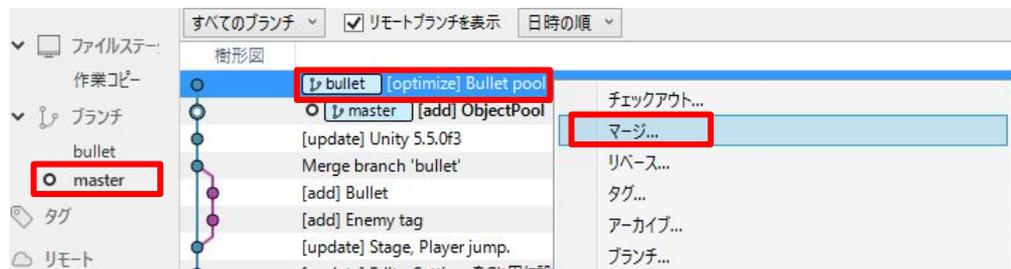
<bulletの親コミット>
適用前: [update] Unity 5.5.0f3
適用後: [add] ObjectPool



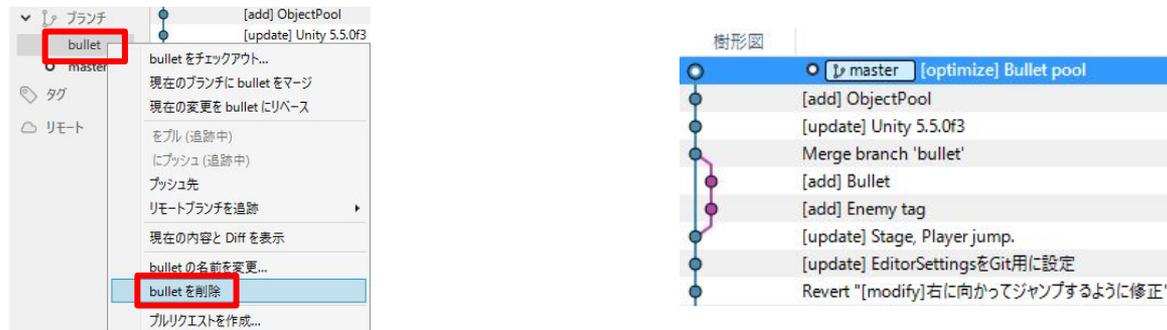
結果、[add] ObjectPoolの内容は同じなので消えてなくなり、
bulletは[optimize] Bullet poolだけになります。

マージ (merge)

1. masterをチェックアウトし、bulletを右クリックしてマージを選択します。



2. bulletを削除する。



補足1

- 同じ内容のコミットが適用される
 - ・コミット内容が既に適用済みの場合は、差分がないためコミットされません。
 - ・差分のある箇所(行)のみ適用されます。
 - ・差分の箇所に競合が起きる可能性もあります。
- 手軽に色々な使い方がある
 - ・必要なコミットを好きな順番で新しいブランチに取り出せる。
 - ・新規ブランチにチェリーピックした後に「最後のコミットを上書き(amend)」で途中のコミットを修正したりできる。

↓

インタラクティブなリベースという機能で実現できます。

補足2

- ObjectPool.csについて
2Dシューティングゲーム チュートリアル
<https://github.com/unity3d-jp-tutorials/2d-shooting-game/wiki>
 - シングルトン
 - 複数のプレハブに対応
- UniRxのObjectPool基底クラス
UniRx/ObjectPool.cs neuecc
<https://github.com/neuecc/UniRx/blob/master/Assets/Plugins/UniRx/Scripts/UnityEngineBridge/Toolkit/ObjectPool.cs>
 - Queueで効率良くPool
 - 毎フレーム少しずつ生成して指定の数だけプリロード

第7章 クリア

最強無敵のGitで好きなコミットをつまみ食いできました。自由に適用したり戻したりできるように、有効にコミットしていきましょう。

次回は、インタラクティブなリベースで過去のコミットを修正して整理してみましよう。