

# 第3章

# Gitでバージョン管理

## ～ハンク編～

私「実装完了、コミット！ん…ここは戻したいな。」  
Git「容易い御用じゃ、行単位で戻してやろう。」

# 今回の目的

- ハンクの変更を破棄 (`checkout -p`)  
変更を破棄したい行を選択して破棄します。  
コミットしたくない部分だけを変更前に戻せます。
- コミットの分割 (`add -p`)  
コミットしたい行を選択してコミットします。  
一度に多くの変更作業をした時でも、複数回に分けてコミットできます。

すべてのブランチ	リモートブランチを表示	親子関係で並べ替え
グラフ	説明	
	<b>master [update]Unity 5.4.1f1</b> [add]Playerスクリプトにジャンプを実装。左クリックと右クリックに対応。 [add]Playerスクリプトを作成して、Cubeにアタッチ [add]CubeにRigidbodyをアタッチ [add]Cubeをシーンに作成 init	

- ・ソース修正する！
- ・不要な変更を破棄する！
- ・3つに分けてコミットする！

グラフ	説明
	<b>master [add]画面外に出たら加速度をゼロに設定</b> [add]ジャンプ力の範囲を3から10までに制限を追加 [modify]右に向かってジャンプするように修正 [update]Unity 5.4.1f1

これは破棄するよ(コミットしない)

```

11 11 [Tooltip("ジャンプする高さ")]
12 12 + [Range(3f, 10f)]
12 13 public float Jump = 5f;
13 14
14 15 private bool _isJump;
15 16
16 17 void Awake()
17 18 {
18 19     _transform = GetComponent<Transform>();
19 20     _rigidbody = GetComponent<Rigidbody>();
20 21 }
21 22
22 23 void Update()
23 24 {
24 25     _isJump = _isJump || Input.GetMouseButtonDown(0) || Input
25 26 }
26 27
27 28 void FixedUpdate()
28 29 {
29 30     if (!_isJump) return;
30 31
31 32     _isJump = false;
32 33
33 34 - _rigidbody.velocity = Vector3.up * Jump;
34 35 + _rigidbody.velocity = Vector3.up * Jump + Vector3.right;
35 36 }
36 37 void OnBecameInvisible()
37 38 {
38 39     _transform.position = ResetPosition;
40 40 + _rigidbody.velocity = Vector3.zero;
41 41 + Debug.Log(Time.time + " 画面外に出たよ", this);
39 42 }

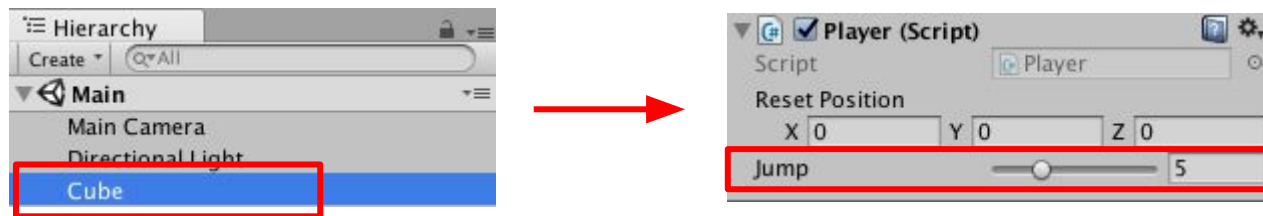
```

# ソースを修正1 (Player.cs)

1. ソースを修正します。(1行追加します。)

```
11 11 [Tooltip("ジャンプする高さ")]  
12 12 + [Range(3f, 10f)]  
12 13 public float Jump = 5f;  
13 14  
14 15 private bool _isJump;
```

2. Cubeを選択し、InspectorビューのPlayerを確認します。
3. Jumpが3~10までの範囲で選択できるようになります。

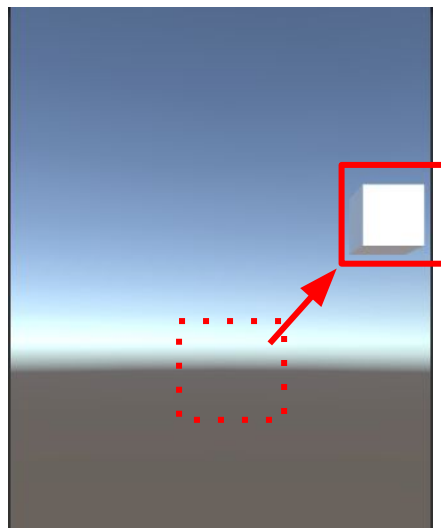


# ソースを修正2 (Player.cs)

1. ソースを修正します。 (“ + Vector3.right;” を書き加えます。)

```
31 32     _isJump = false;
32 33
33  -     _rigidbody.velocity = Vector3.up * Jump;
34  +     _rigidbody.velocity = Vector3.up * Jump + Vector3.right;
34 35 }
```

2. シーンを再生して、  
左クリックでCubeがジャンプします。
3. Cubeが右側に向かって  
ジャンプするようになります。



# ソースを修正3 (Player.cs)

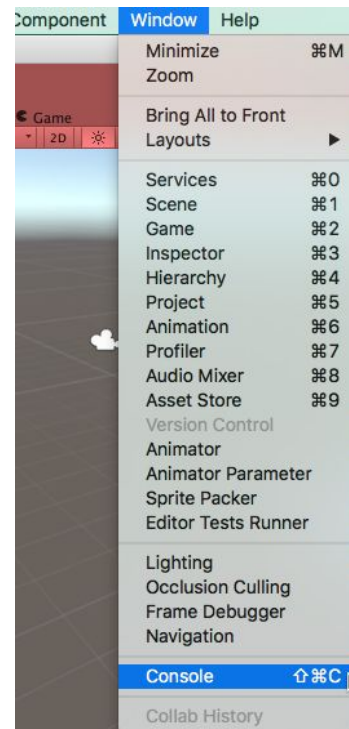
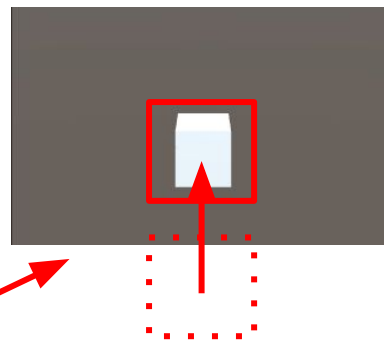
1. ソースを修正します。(2行追加します。)

```
37 38     {  
38 39         _transform.position = ResetPosition;  
40 +         _rigidbody.velocity = Vector3.zero;  
41 +         Debug.Log(Time.time + " 画面外に出たよ", this);  
39 42     }
```

2. シーンを再生します。

3. Cubeが画面外に出ると  
元に戻り、加速度が0になります。

4. Consoleにログも表示されます。



# ハンクを確認

1. SourceTreeで修正を確認します。
2. 修正が3箇所あり、ハンクが2つあります。

Index にステージしたファイル

Assets/Player.cs

Hunk 1: 行 9-15 Hunk をステージへ移動 Hunk を破棄

```
9 9 private Rigidbody _rigidbody;
10 10
11 11 [Tooltip("ジャンプする高さ")]
12 12 + [Range(3f, 10f)]
12 13 public float Jump = 5f;
13 14
14 15 private bool _isJump;
```

Hunk 2: 行 31-43 Hunk をステージへ移動 Hunk を破棄

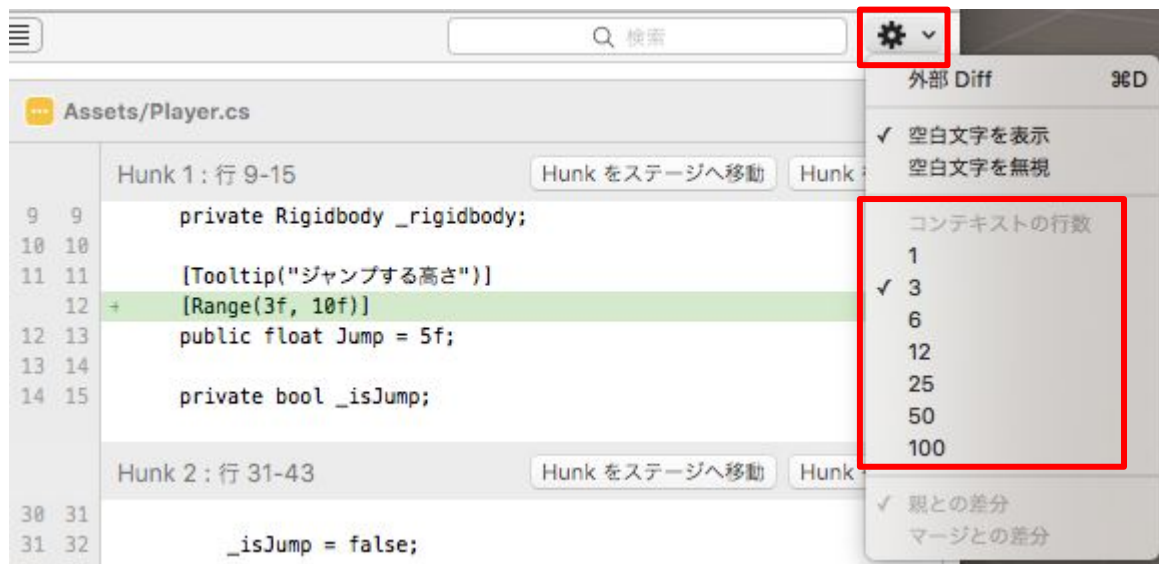
```
30 31
31 32 _isJump = false;
32 33
33 33 - _rigidbody.velocity = Vector3.up * Jump;
34 34 + _rigidbody.velocity = Vector3.up * Jump + Vector3.right;
34 35 }
35 36
36 37 void OnBecameInvisible()
37 38 {
38 39 _transform.position = ResetPosition;
40 40 + _rigidbody.velocity = Vector3.zero;
41 41 + Debug.Log(Time.time + " 画面外に出たよ", this);
39 42 }
40 43 }
```

\ No newline at end of file

# ハンクを確認

3. コンテキストの行数を6に変更します。

修正したコードの  
前後を表示する  
行数が変わります。





# ハンクを確認

4. コンテキストの行数が6に増えたため、ハンクが1つになります。  
修正箇所の前後の表示行数が変わり、ハンクも変わります。

5. コンテキストの行数を3に戻しておきます。

作業ツリーのファイル

Assets/Player.cs

```
8 8 private Transform _transform;
9 9 private Rigidbody _rigidbody;
10 10
11 11 [Tooltip("ジャンプする高さ")]
12 + [Range(3f, 10f)]
12 13 public float Jump = 5f;
13 14
14 15 private bool _isJump;
15 16
16 17 void Awake()
17 18 {
18 19     _transform = GetComponent<Transform>();
19 20     _rigidbody = GetComponent<Rigidbody>();
20 21 }
21 22
22 23 void Update()
23 24 {
24 25     _isJump = _isJump || Input.GetMouseButtonDown(0) || Input.GetMouseButtonDown(1);
25 26 }
26 27
27 28 void FixedUpdate()
28 29 {
29 30     if (!_isJump) return;
30 31
31 32     _isJump = false;
32 33
33 - _rigidbody.velocity = Vector3.up * Jump;
34 + _rigidbody.velocity = Vector3.up * Jump + Vector3.right;
34 35 }
35 36
36 37 void OnBecameInvisible()
37 38 {
38 39     _transform.position = ResetPosition;
40 + _rigidbody.velocity = Vector3.zero;
41 + Debug.Log(Time.time + " 画面外に出たよ", this);
39 42 }
```

# ハンクの変更を破棄 (*checkout -p*)

1. `Debug.Log`の行を選択します。背景色が青色になります。

```
Hunk 2 : 行 28-43      選択した行をステージへ移動  選択した行を破棄
27 28   void FixedUpdate()
28 29   {
29 30       if (!_isJump) return;
30 31
31 32       _isJump = false;
32 33
33 34   -   _rigidbody.velocity = Vector3.up * Jump;
34 35   +   _rigidbody.velocity = Vector3.up * Jump + Vector3.right;
34 35   }
35 36
36 37   void OnBecameInvisible()
37 38   {
38 39       _transform.position = ResetPosition;
39 40   +   rigidbody.velocity = Vector3.zero;
40 41   +   Debug.Log(Time.time + " 画面外に出たよ", this);
39 42   }
40 43   }
```

```
Hunk 2 : 行 28-42      Hunk をステージへ移動  Hunk を破棄
27 28   void FixedUpdate()
28 29   {
29 30       if (!_isJump) return;
30 31
31 32       _isJump = false;
32 33
33 34   -   _rigidbody.velocity = Vector3.up * Jump;
34 35   +   _rigidbody.velocity = Vector3.up * Jump + Vector3.right;
34 35   }
35 36
36 37   void OnBecameInvisible()
37 38   {
38 39       _transform.position = ResetPosition;
39 40   +   rigidbody.velocity = Vector3.zero;
39 41   }
40 42   }
```

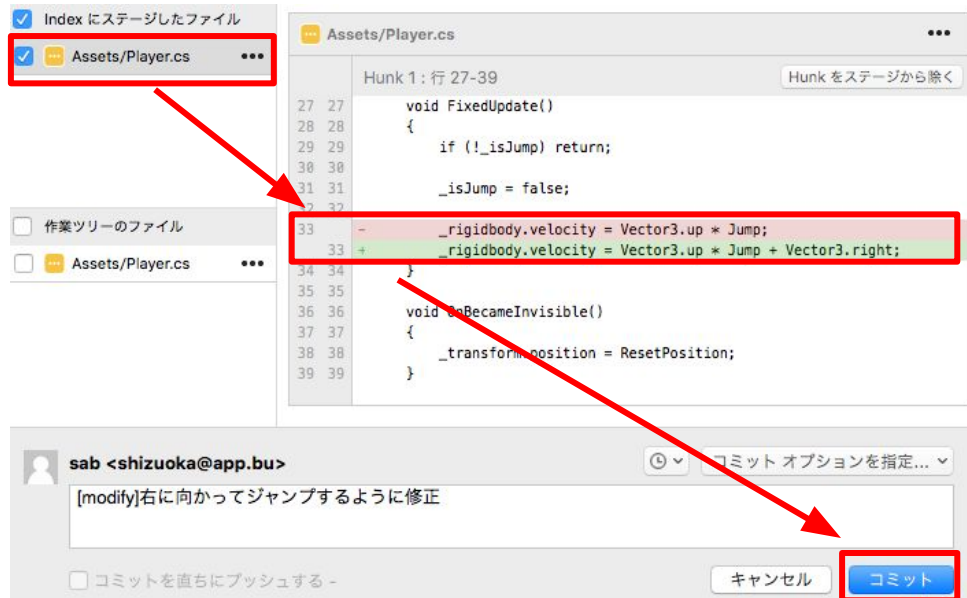
2. 「選択した行を破棄」をクリックすると、変更が破棄されます。

# コミットの分割1 (*add -p*)

1. 右に向かってジャンプするように修正した2行を選択します。
2. 「選択した行をステージへ移動」をクリックします。(破棄ではない)
3. Index にステージしたファイルを選択して、2行ステージされたことを確認します。

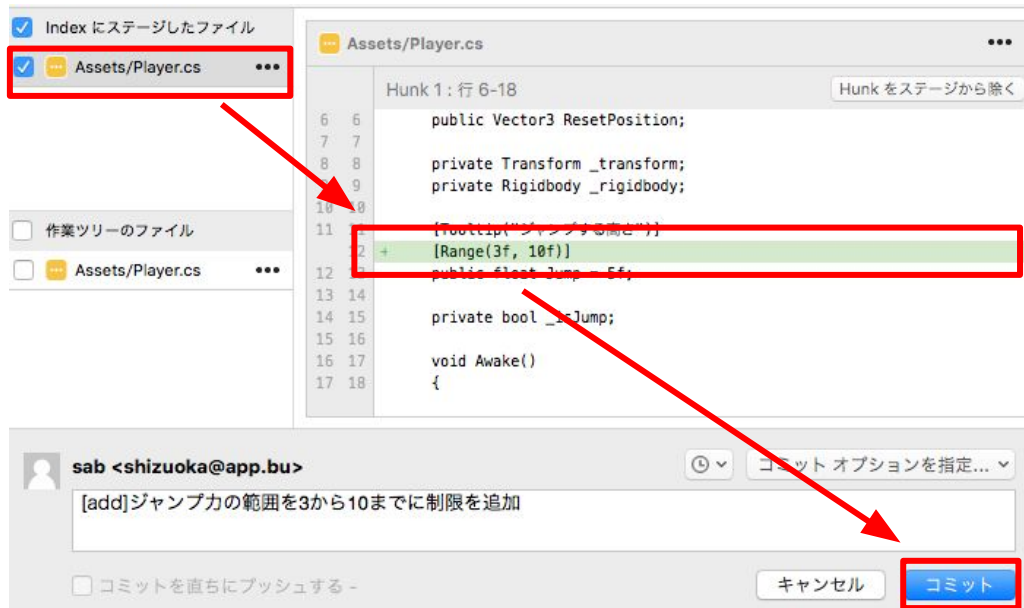
(補足)

1行目:変更前の行を削除  
2行目:変更後の行を追加  
→結果、1行の変更になる。



# コミットの分割2 (*add -p*)

1. ジャンプ力の範囲を3から10までに制限した修正をコミットします。
2. 「Hunk をステージへ移動」をクリックすると、Hunk内の変更が全て対象になります。



# コミットの分割3 (*add -p*)

1. 画面外に出たら加速度をゼロに設定する修正をコミットします。

変更が1箇所しか残っていないため、ファイルごとステージへ移動しても良いです。


The screenshot shows a code editor interface. On the left, a file explorer shows 'Assets/Player.cs' selected. The main editor displays a hunk of code with the following content:

```
Hunk 1: 行 34-42
34 34     _rigidbody.velocity = Vector3.up * Jump + Vector3.right;
35 35     }
36 36
37 37     void OnBecameInvisible()
38 38     {
39 39         _transform.position = ResetPosition;
40 40         _rigidbody.velocity = Vector3.zero;
41 41     }
42 42
\ No newline at end of file
```

The commit dialog at the bottom shows the commit message: '[add]画面外に出たら加速度をゼロに設定'. The 'コミット' button is highlighted with a red box.

# コミットの分割完了

不要な変更を破棄して、  
修正内容別に3回のコミットに分割できました。

グラフ	説明
	<b>master</b> [add]画面外に出たら加速度をゼロに設定 [add]ジャンプ力の範囲を3から10までに制限を追加 [modify]右に向かってジャンプするように修正 [update]Unity 5.4.1f1 [add]Playerスクリプトにジャンプを実装。左クリックと右クリックに対応。 [add]Playerスクリプトを作成して、Cubeにアタッチ [add]CubeにRigidbodyをアタッチ [add]Cubeをシーンに作成 init

# 補足

- Hunk / 選択した行  
「Hunk をステージへ移動」or「選択した行をステージへ移動」、  
「Hunk を破棄」or「選択した行を破棄」、  
「Hunk をステージから除く」or「選択した行をステージから除く」  
が操作できます。
- Hunkと選択した行は、選択範囲が異なるだけです。
- ステージへ移動とステージから除く操作は、  
コミット前なら何度でもできます。

# まとめ

- ハンクの変更を破棄 (`checkout -p`)  
変更を行単位に戻せます。  
混乱した時に、好きな行だけ戻すのもGood！
- コミットの分割 (`add -p`)  
選択した行だけコミットできます。  
修正途中だけど、記録を残したいという時に、  
一部だけコミットをするのもGood！



# 第3章 クリア

最強無敵のGitでコミットを分割したり、一部の変更を破棄できました！

ガンガンコーディングして、ハンクや破棄を活用してどんどんコミットしましょう。

次回は、ブランチ・マージを使って、並行作業をしてみましよう。Gitの本格的な運用になります。