

# Unity git 手順\_完成\_0.02

製作者：AT12B 173 37 柳澤優太

## 大まかな手順説明

## 0.前提

## 1.ProgateでGitのコマンド理解

## 2. Gitのアカウント作成

### 3.Git hubの説明

#### 4. ソースツリーインストール

## 5. Git hubでクローン、フォーク

## 6. ソースツリーの使用

## 7.Unityで開く

## 8.更新差分をSourcetreeでプッシュ

## 9. 差分のプル

## 10.番外編) 本家リポジトリを作成する

## 他のチームがこの資料を使用する際の注意点

## 前提と注意事項

## Git

リポジットリ

Github

本編

#### 4. ソースツリーインストール

## 5. Git hubでクローン、フォーク

## 6. ソースツリーの使用

## 7.Unityで開く

## 8.更新差分をSourcetreeでプッシュ

リモート名：upstream で統一

何か変更したら

## Git hubでプルリクエスト

## 9. 差分のプル

お疲れさまでした！！！！！！！！！！！！！！！！

## 10.番外編) 本家リポジトリを作成する

## コマンドラインでクローン

## ソースツリーでクローン

## 大まかな手順説明

## 0.前提

## 1.ProgateでGitのコマンド理解

<https://prog-8.com/courses>

<https://prog-8.com/git/study/1/1>

→ コマンドを使いこなせなくても、できることと手順を理解して欲しい

## 2. Gitのアカウント作成

<https://prog-8.com/docs/git-env-win>

→ みんなでやる予定

## 3.Git hubの説明

→ みんなでやる予定

## 4.ソースツリーインストール

[https://prog-8.com/blogs/how\\_to\\_use\\_sourcetree](https://prog-8.com/blogs/how_to_use_sourcetree)

## 5.Git hubでクローン、フォーク

## 6.ソースツリーの使用

## 7.Unityで開く

## 8.更新差分をSourcetreeでプッシュ

- Sourcetreeに本家リポジトリのURLを登録します
- Git hubでプルリクエスト

## 9.差分のプル

## 10.番外編) 本家リポジトリを作成する

これはチームで一人できる人がいればOKです

本家リポジトリを作成するので絶対にやることにはなる。

## 他のチームがこの資料を使用する際の注意点

- “4.ソースツリーインストール”までやったら、チーム内の誰か**1人が番外編**をやってください。
- そこで作った本家リポジトリを使用して制作してってください。
- このPDFだけですべて完結していないので、外部サイトへのアクセスがあったらそこで学んでください。
- わからなかったら、調べてください。それでもわからなかったらチーム306の柳澤優太へ相談してください。

## 前提と注意事項

- 以下の通りやったけど駄目だった場合、柳澤か安藤に相談してください。
- Progateでgitの基本的な知識を前提としています
- 最低限 git add → git commit -m "" → git push (→ git pull)の機能を理解してください。
- **作業始める前にPullをしましょう！**

## Git

- git は ソースコードなどの変更履歴を記録・追跡するための分散型バージョン管理システム
- git の操作方法はcmdによるコマンド操作、Sourcetreeによる視覚的な操作などがある

## リポジトリ

- リポジトリ→ファイルなどを記録する場所
- **本家リポジトリ→（ここでは、[AT0139/unity\\_test \(github.com\)](https://github.com/AT0139/unity_test)**
- ー>こっちに変更 <https://github.com/AT0139/kyouyuuyou>
- **自分のリポジトリ→本家リポジトリからForkして作成したリポジトリ**
- **本家リポジトリを自分のリポジトリにコピーして、更新を本家リポジトリに反映する流れ**
- **↑なので本家リポジトリ側の操作は記載していません！**

## Github

- git hubの画面慣れないよな。わかる。でも我慢してくれ。
- git hubの画面は**絶対に日本語翻訳しないでください**

## 本編

### 4.ソースツリーインストール

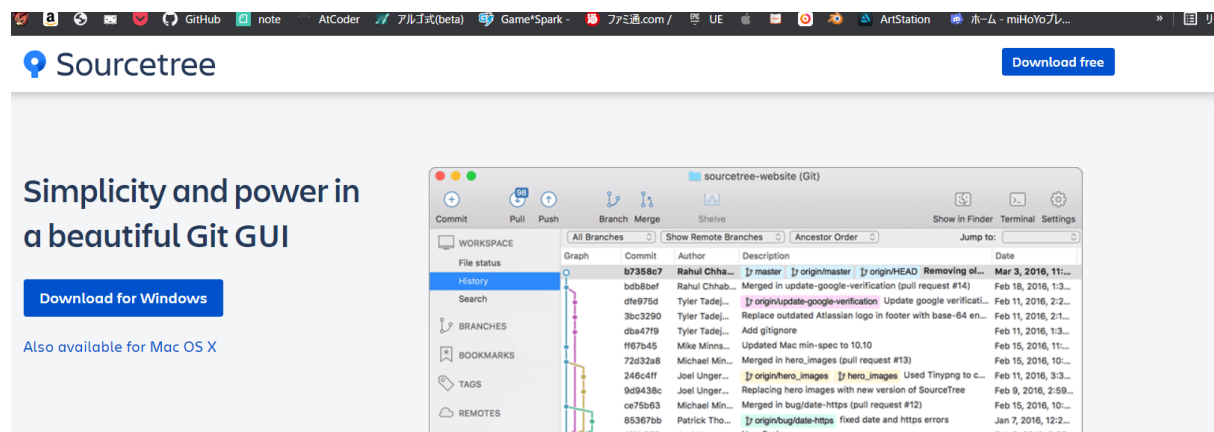
↓優秀サイトProgateで手順が一通り説明されています。

[https://prog-8.com/blogs/how\\_to\\_use\\_sourcetree](https://prog-8.com/blogs/how_to_use_sourcetree)

#### Sourcetree | Free Git GUI for Mac and Windows

As our new Bitbucket space station administrator, you need to be organized. When you make files for your space station, you'll want to keep them in one place and shareable with teammates,

<https://www.sourcetreeapp.com/>

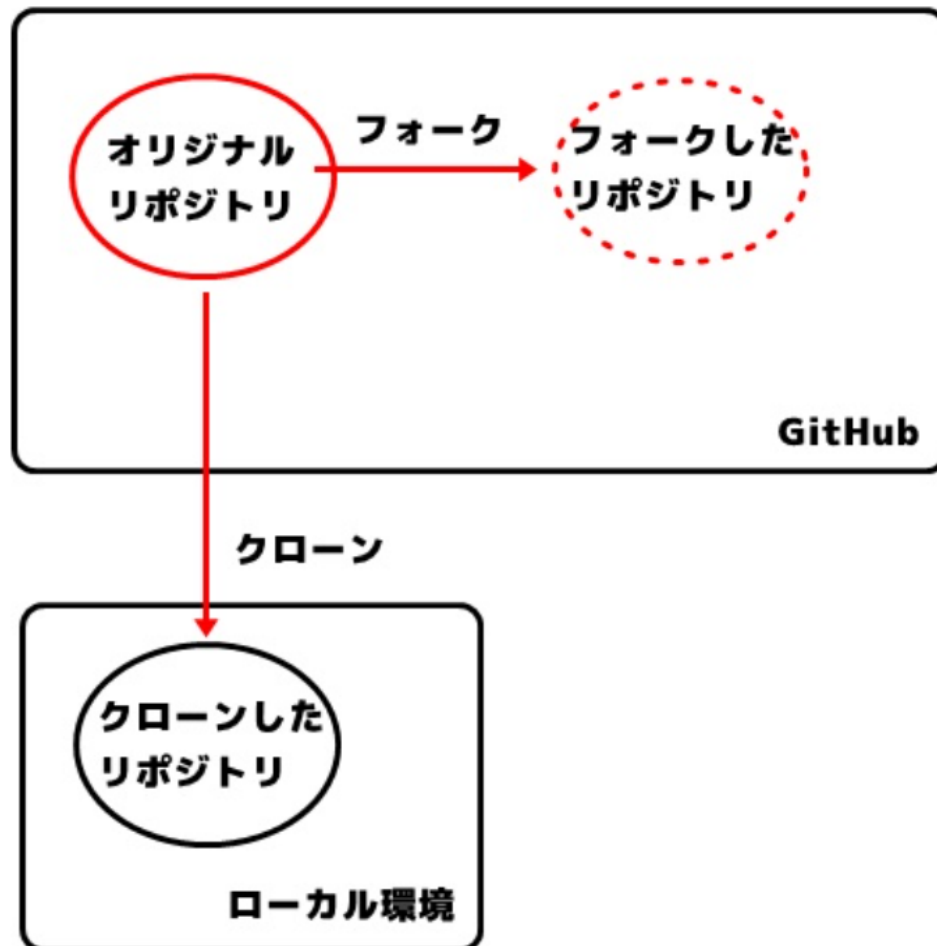


Download free

## 5.Git hubでクローン、フォーク

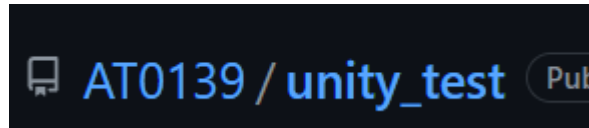
fork（派生）とはある時点のリポジトリをベースとして、GitHub上に別のリポジトリとして作成することです。

→ 共同開発ができるようになります！

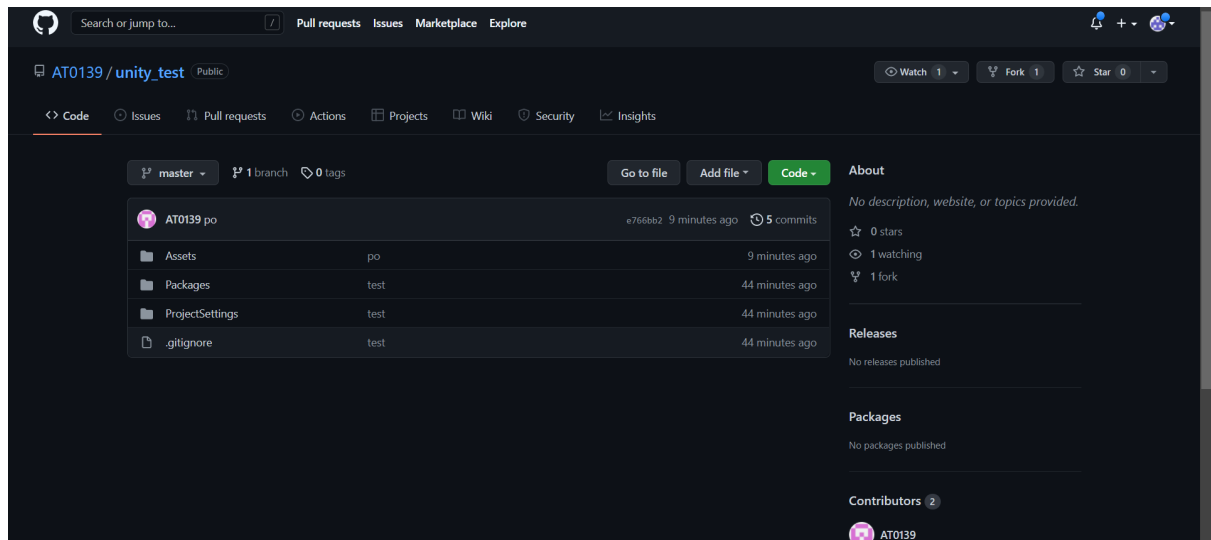


↓（テスト）本家リポジトリに行くと  
（これは使えないけど、テストでやってみてね）

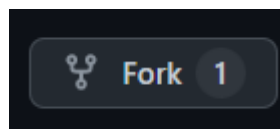
[https://github.com/AT0139/unity\\_test](https://github.com/AT0139/unity_test)



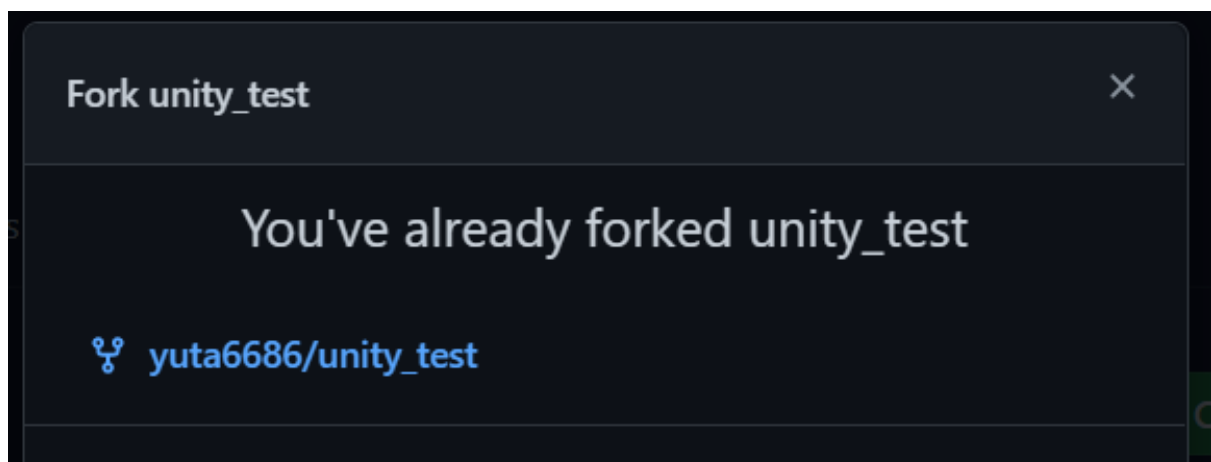
こんな画面になります



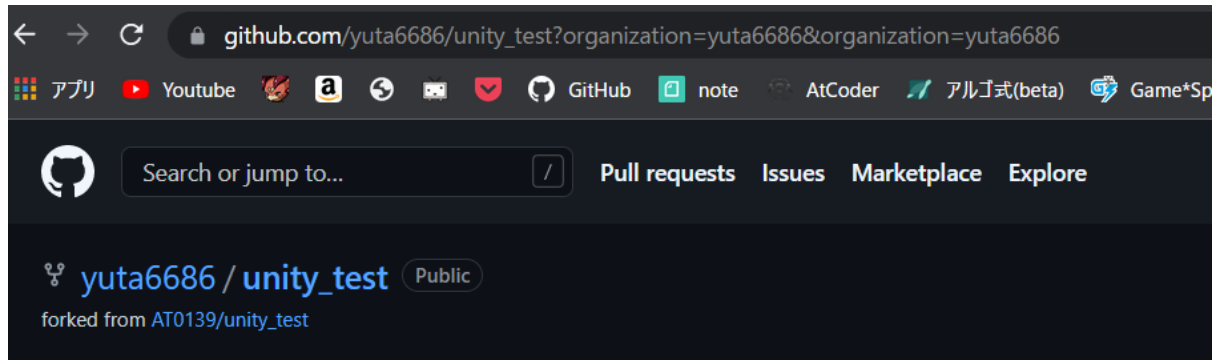
右上のForkボタンをクリック



自分のリポジトリを作成します  
(↓は既に作ってあるw)



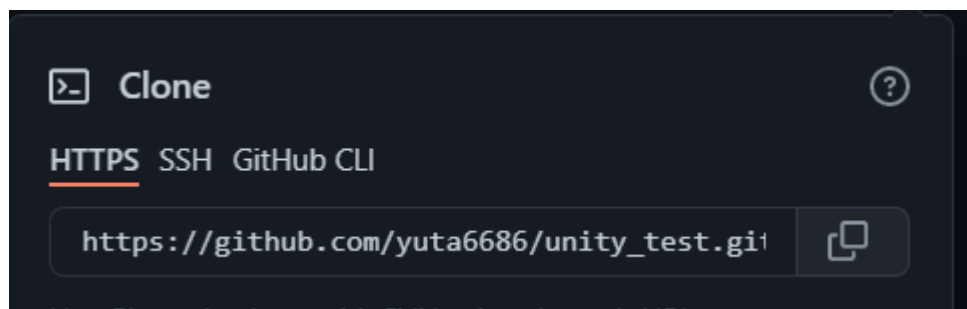
↓自分のリポジトリになった（"URL"と"forked from AT0139/unity\_test"を見ればわかるかな）



緑色のCodeボタンを押して、



URLをコピー



↓コピーペースト

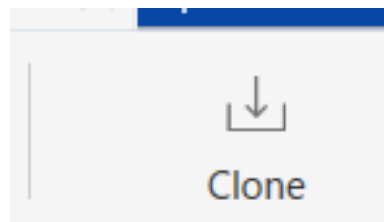
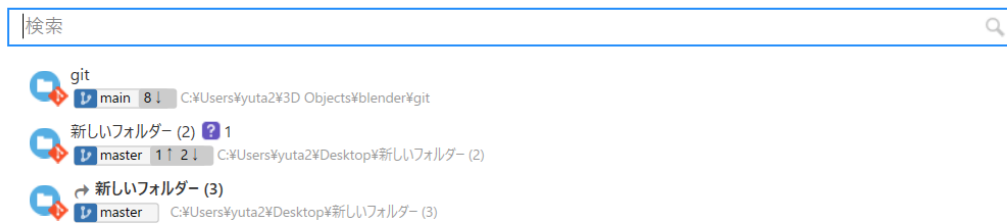


## 6. ソースツリーの使用

このURLをSourceTreeでCloneします



### ローカルリポジトリ





# Clone

Cloning is even easier if you set up a [remote account](#)

参照

リポジトリタイプ: [?](#) パス / URL が指定されていません

参照

Local Folder:

＞ 詳細オプション

クローン

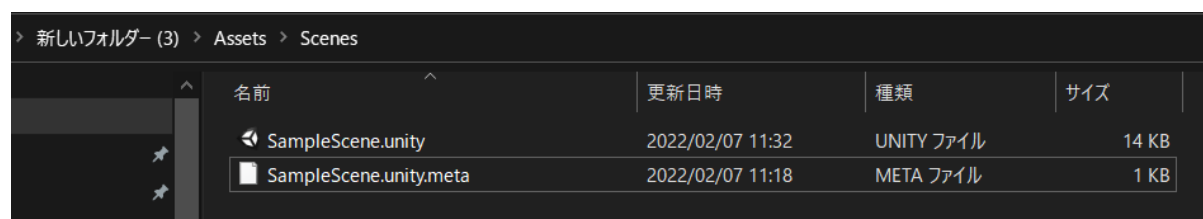
クローンするパスを設定したら


クローン

## 7.Unityで開く

クローンできたら

Assets → Scene → .unity

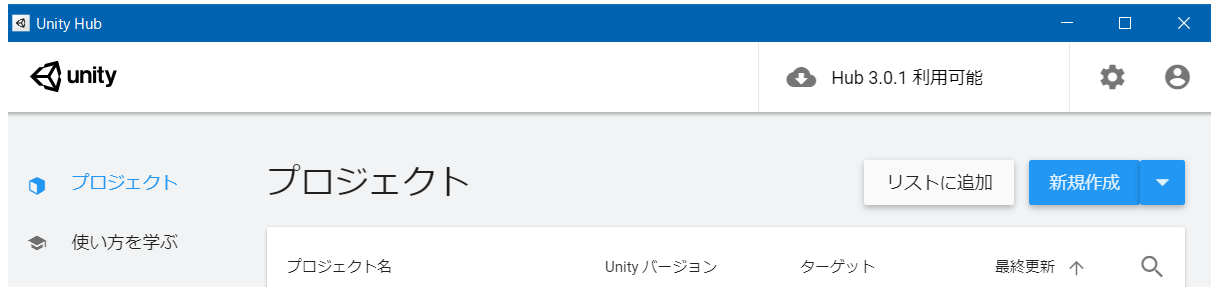


＞ 新しいフォルダー (3) ＞ Assets ＞ Scenes			
名前	更新日時	種類	サイズ
 SampleScene.unity	2022/02/07 11:32	UNITY ファイル	14 KB
 SampleScene.unity.meta	2022/02/07 11:18	META ファイル	1 KB

これを押すと、Hubが開きます

SampleScene.unity

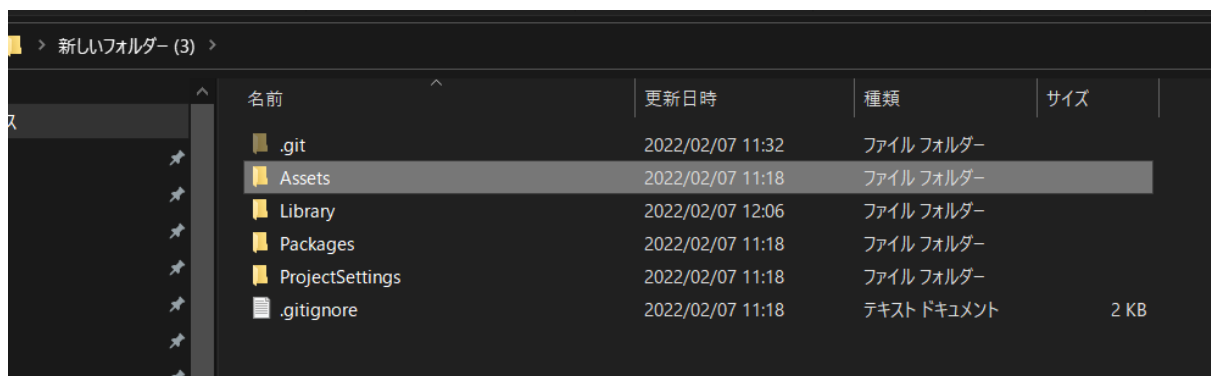
ハブで



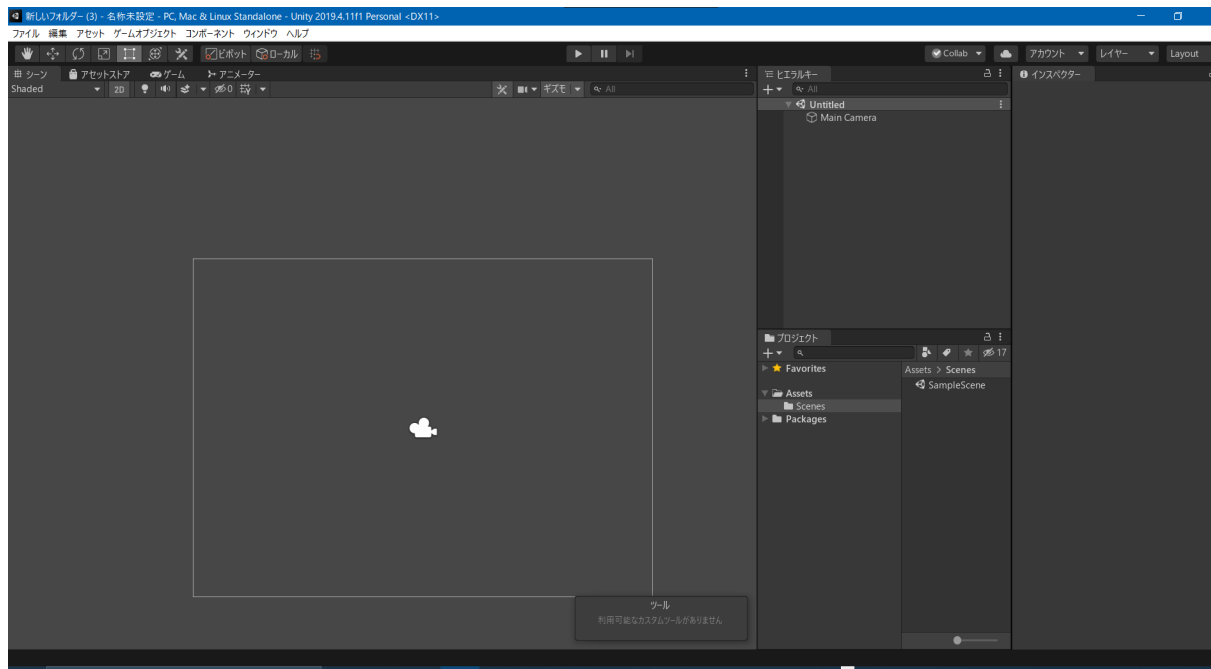
リストに追加で

リストに追加

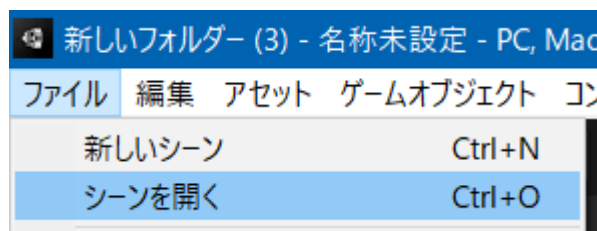
プロジェクトのパスを追加（ここでは新しいフォルダー(3)）



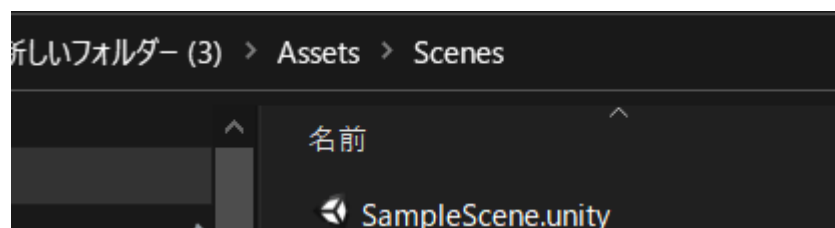
プロジェクト開いてもまだできてません



シーンを開くで



.unityを開く




これで、ちゃんとクローンできたはず。


## 8.更新差分をSourcetreeでプッシュ

まず、Sourcetreeに本家リポジトリのURLを登録します



リポジトリ設定

リモート

詳細

リモートリポジトリのパス

名前	パス
origin	https://github.com/yuta6686/unity_test.git
up	https://github.com/AT0139/unity_test.git

追加

編集

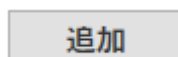
削除

設定ファイルを編集...

OK

キャンセル

追加ボタンをクリック



**リモートの詳細設定**

必要な情報

リモート名:  ☐ デフォルトリモート

URL/パス:

---

外部サービスとの拡張統合オプション

Remote Account:

 **Generic Account**  
Generic Host

Legacy Account Settings:

ホストタイプ:

ルート URL:

ユーザー名:

拡張統合を使うと Bitbucket など外部のホスティングプロバイダとより深く連携することができます。例えば、サイトでリンクを開いたときに既存のクローンを見つけて表示し、そのままプルリクエストを作成することができます。

ここに本家リポジトリのURL↓を入れる  
(これは使えない)

[https://github.com/AT0139/unity\\_test](https://github.com/AT0139/unity_test)

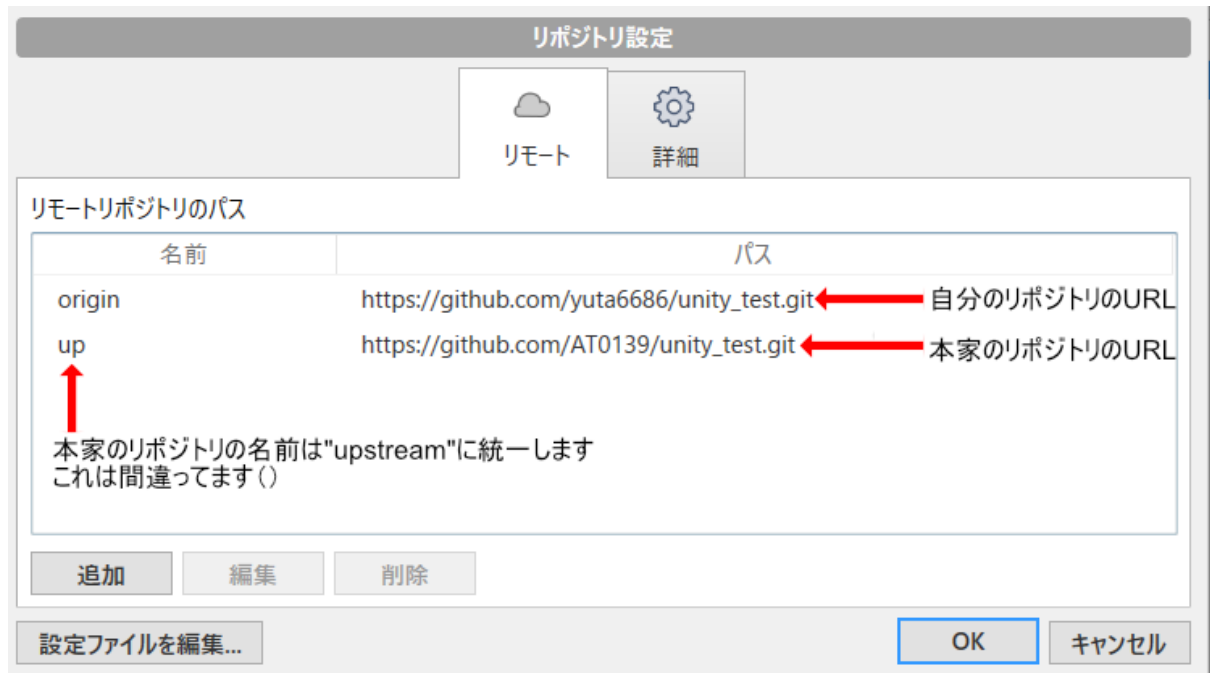
URL/パス:

リモート名:  ☐ デフォルトリモート

URL/パス:

**リモート名：upstream で統一**

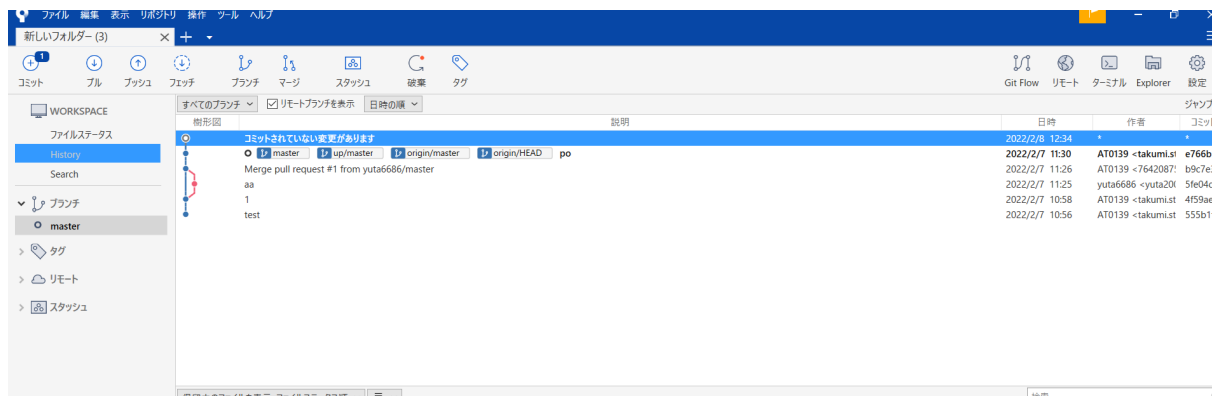
こんな感じになればOK



## 何か変更したら

保存してコミットしてプッシュ

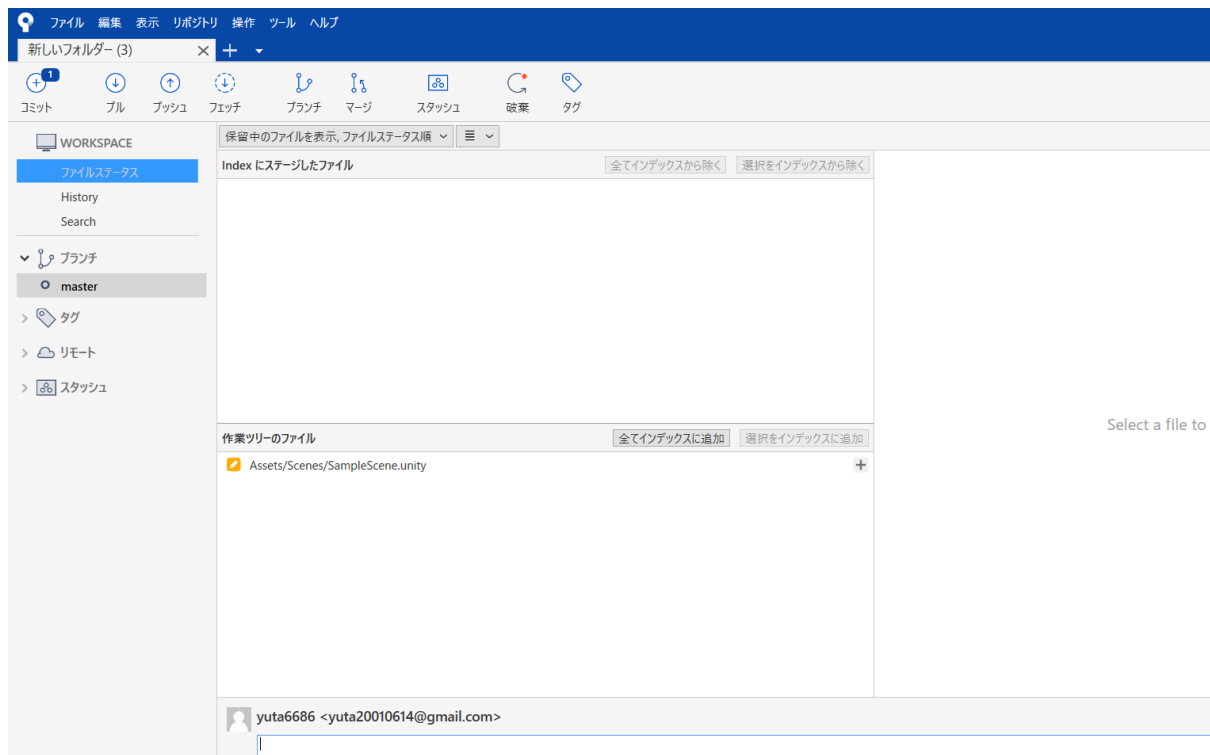
変更があるとコミットに通知が付きます



これをクリック



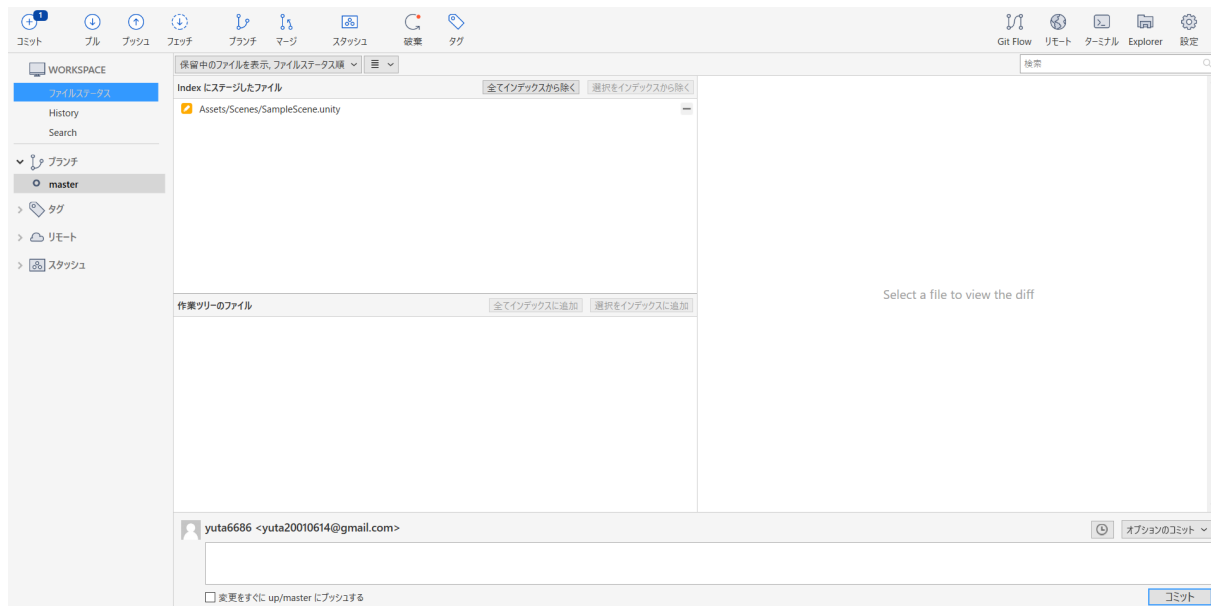
こんな画面



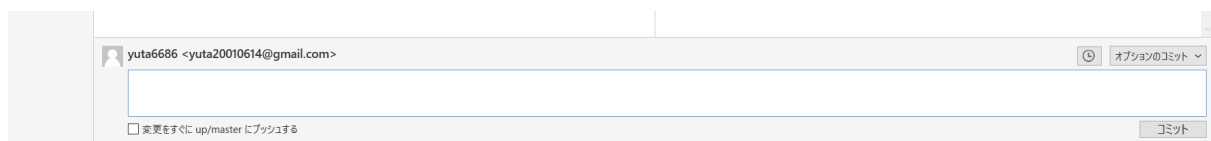
作業ツリーのファイルで共有したファイルを選んで+  
OR すべてインデックスに追加



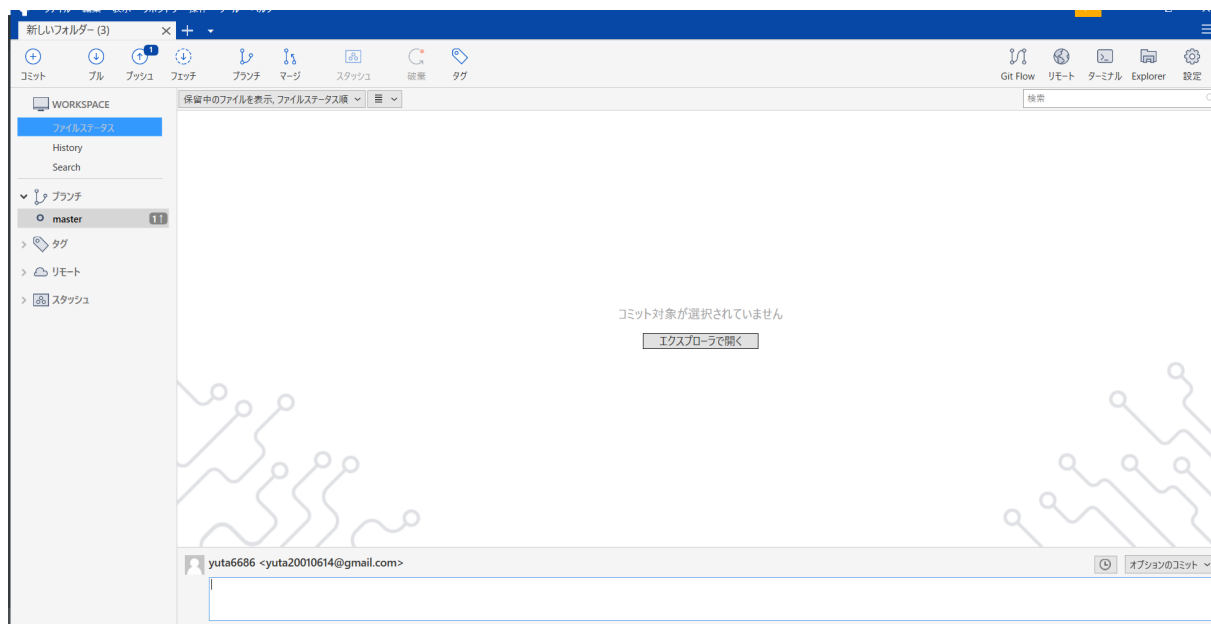
追加すると、上にファイルが移動する



下のここでコミットメッセージを打ちます

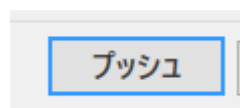


コミットすると



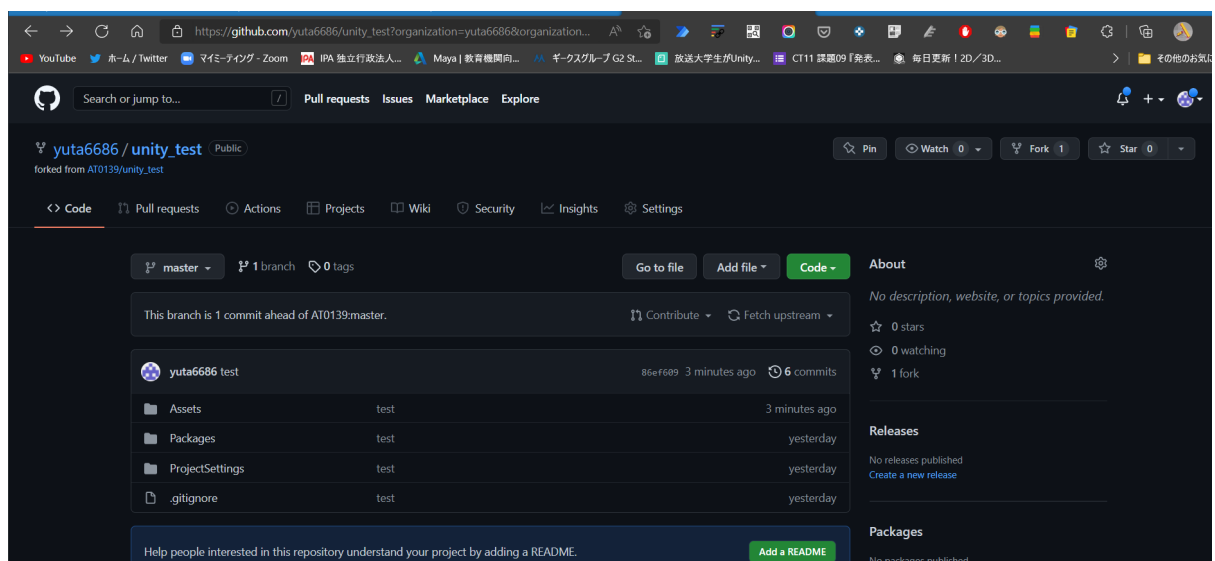


プッシュに通知が付きます

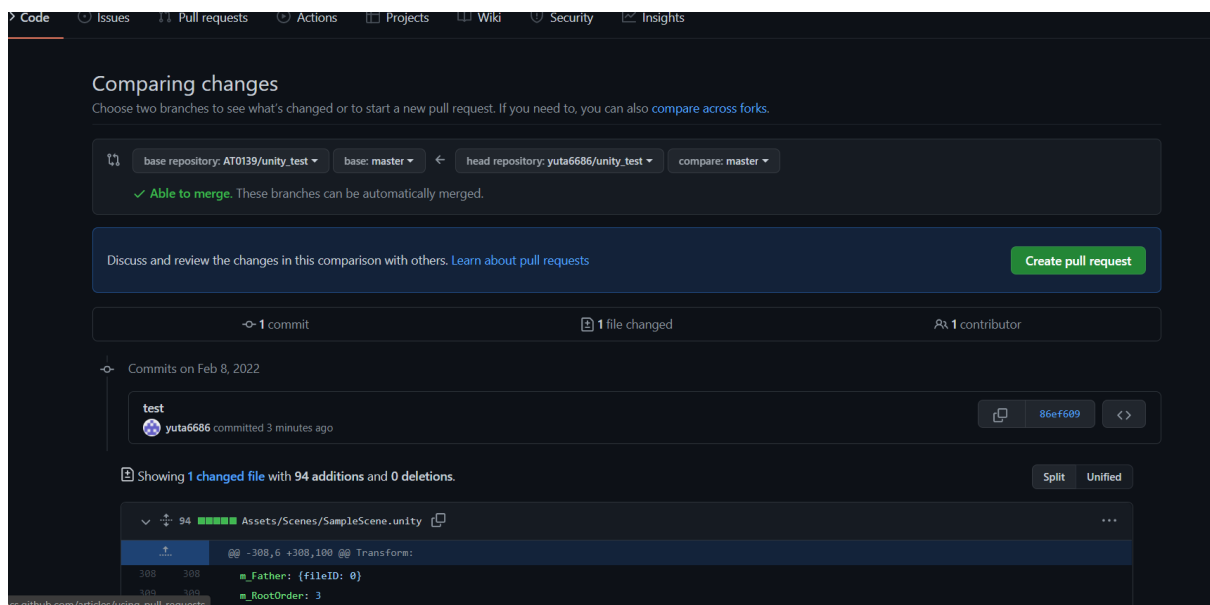
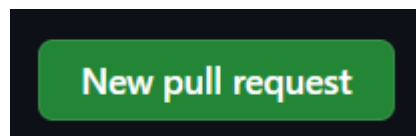
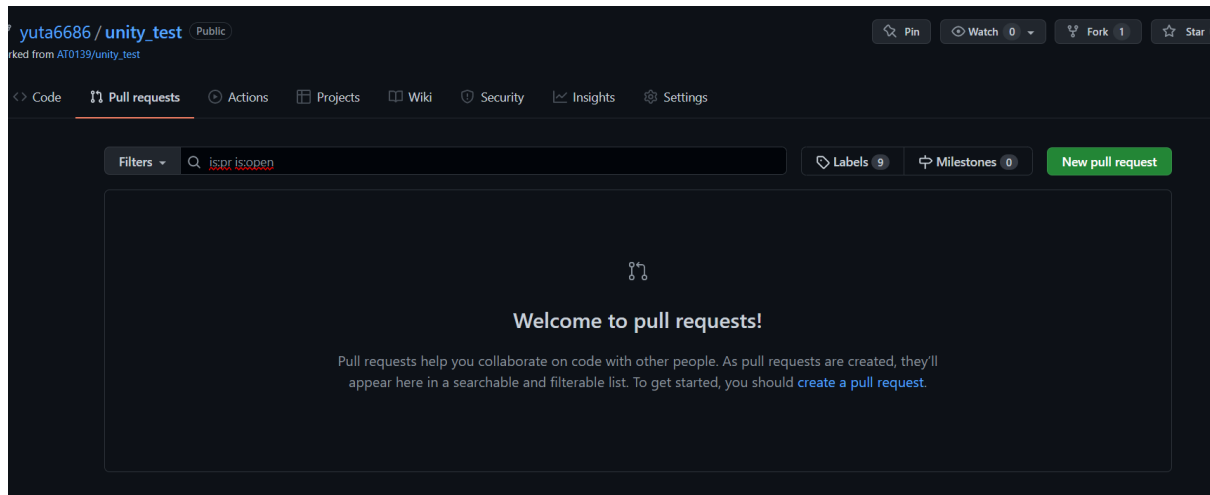
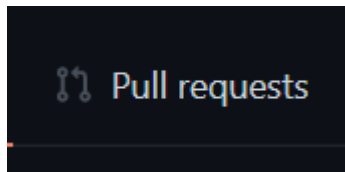


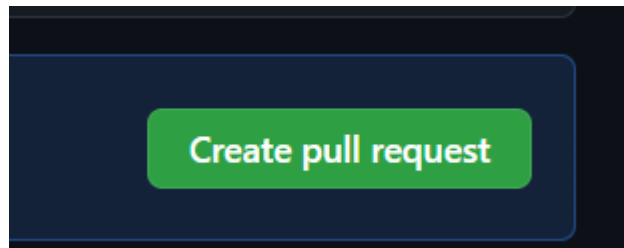
## Git hubでプルリクエスト

まずは自分のリポジトリへ

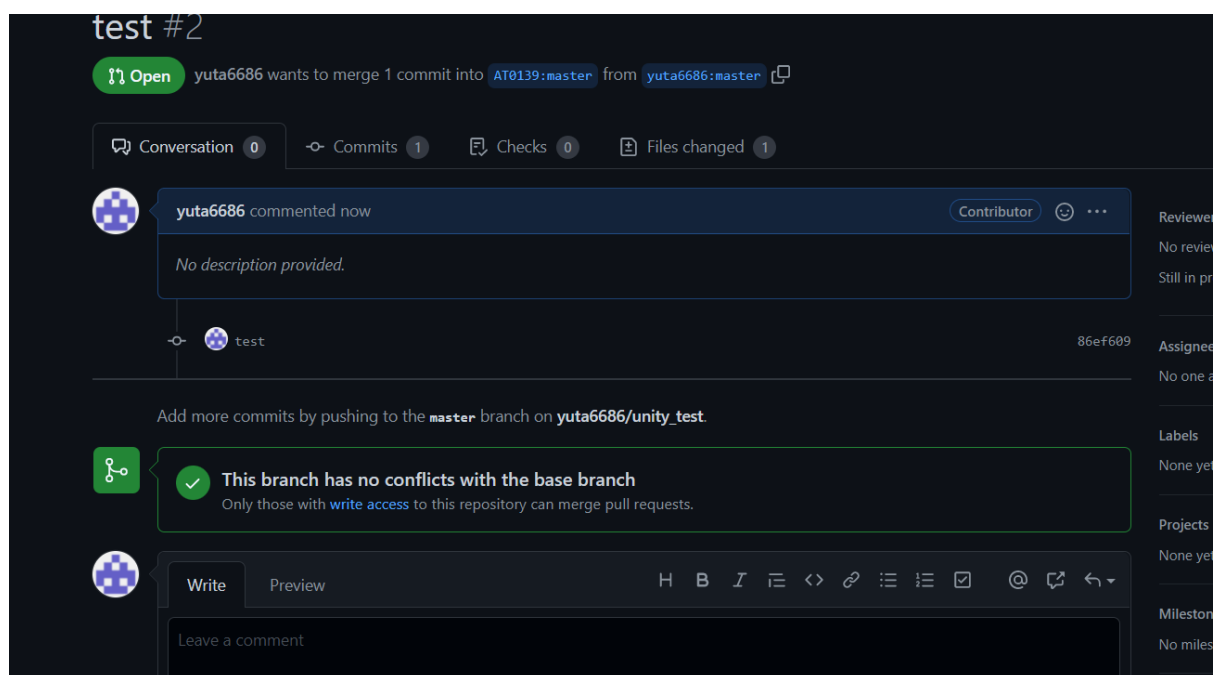


## Pull Requestをクリック





まだ、気を抜かないでください





これ終わったら、


本家リポジトリの主にプルリクエストお願いしますと言ってください

本家画面↓


Add more commits by pushing to the **master** branch on **yuta6686/unity\_test**.












 **Continuous integration has not been set up**  
[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

 **This branch has no conflicts with the base branch**  
Merging can be performed automatically.


**Merge pull request** You can also open [this in GitHub Desktop](#) or view [command line instructions](#).





Write Preview H B I         

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.


 Close pull request **Comment**





 Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).


 **ProTip!** Add `.patch` or `.diff` to the end of URLs for Git's plaintext views.


プルリクエスト対応後↓

## test #2


**Merged** AT0139 merged 1 commit into `AT0139:master` from `yuta6686:master`  now

 Conversation **0**  Commits **1**  Checks **0**  Files changed **1**





yuta6686 commented 2 hours ago Contributor  ...


No description provided.












test 86ef609



AT0139 merged commit `3a61f53` into `AT0139:master` now **Revert**




Write Preview H B I         

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

**Comment**

 Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

## 9.差分のプル



プル

次のリモートからプル: origin

https://github.com/yuta6686/unity\_test.git

プルするリモートブランチ: master 更新

次のブランチにプル: master

オプション

- ☒ マージした変更を即座にコミット
- ☐ マージ元コミットのメッセージをマージ結果のコミットメッセージに含める
- ☐ ファストフォワード可能でも新たなコミットを作成する
- ☐ マージの代わりにリベースする (注意: 変更をプッシュしていない場合に限り使ってください)

プル キャンセル

upstreamね

プル

次のリモートからプル: origin

origin  
up  
master

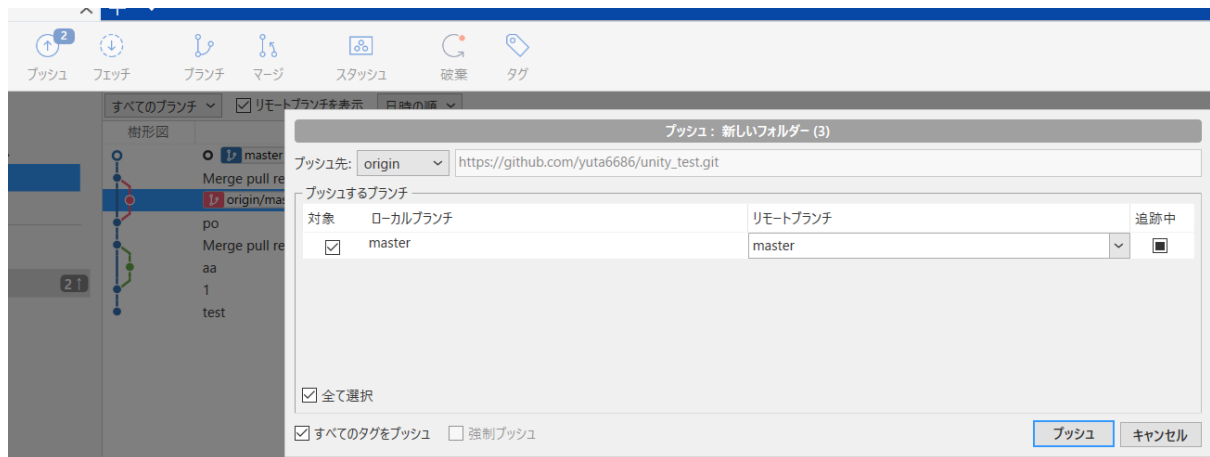
プルするリモートブランチ: master 更新

次のブランチにプル: master

オプション



プルできたら、自分のリポジトリにプッシュしてあげましょう！



プル。以上。終わり！

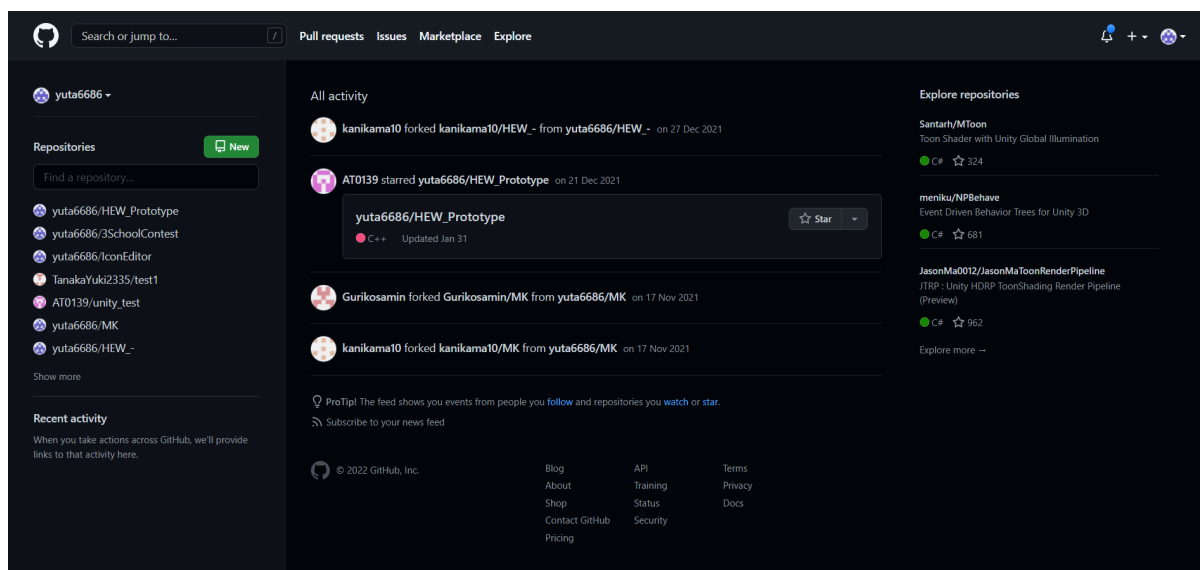
Unity開いてみて変更されてたら完璧！

**お疲れさまでした！！！！！！！！！！！！！！！！**

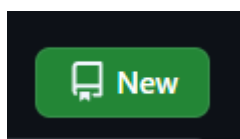
## 10.番外編) 本家リポジトリを作成する

コマンドラインでもSourcetreeでも同じことができます。  
好きな方でやってください。

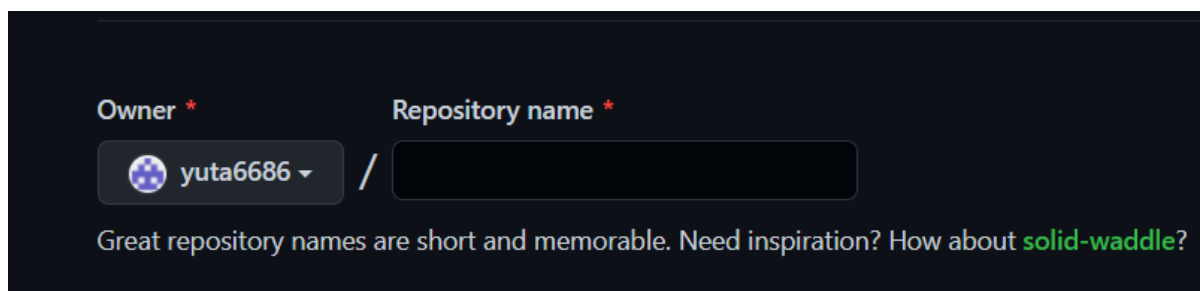
## ↓自身のGitHubページ



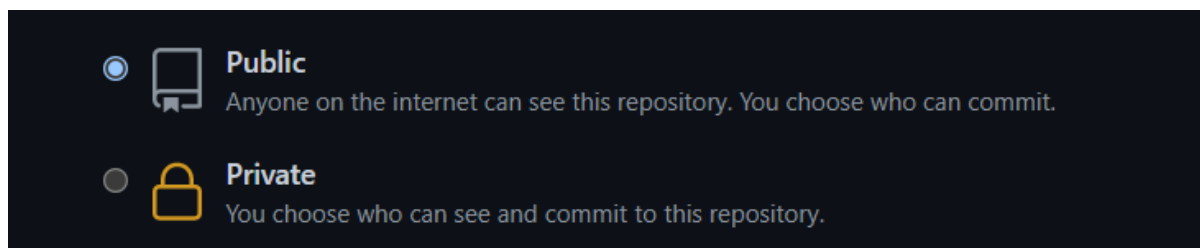
## 左のNewをクリック



## リポジトリ名を入力



## 自分以外に公開するか？ public or private

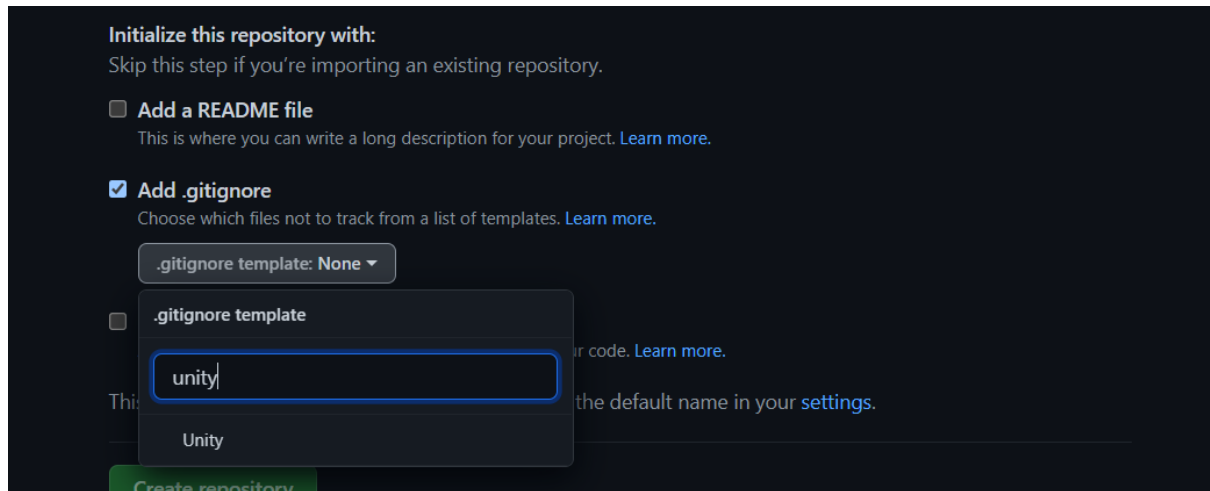


## unityを使用する場合は、Add.gitignoreにチェック

Unityを探して選択

▼ .gitignoreとは？



- 追加したくないファイルを追加しないようにする機能
- Unityを選べば、Unityで使用する保存しなくていいファイルを自動的に弾いて追加してくれる
- 自分も最近知りました。



↓こんな感じでOK





Owner \*      Repository name \*

 yuta6686 / Unity Repo Test 

Great repository names are short and memorable. Need inspiration? How about [solid-waddle?](#)

Description (optional)

☒  **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.


**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

☒ **Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)

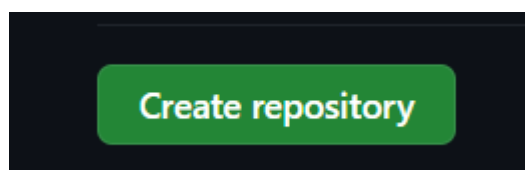
**.gitignore template:** Unity ▼

☐ **Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

**Create repository**

これでリポジトリが作成されます



## コマンドラインでクローン

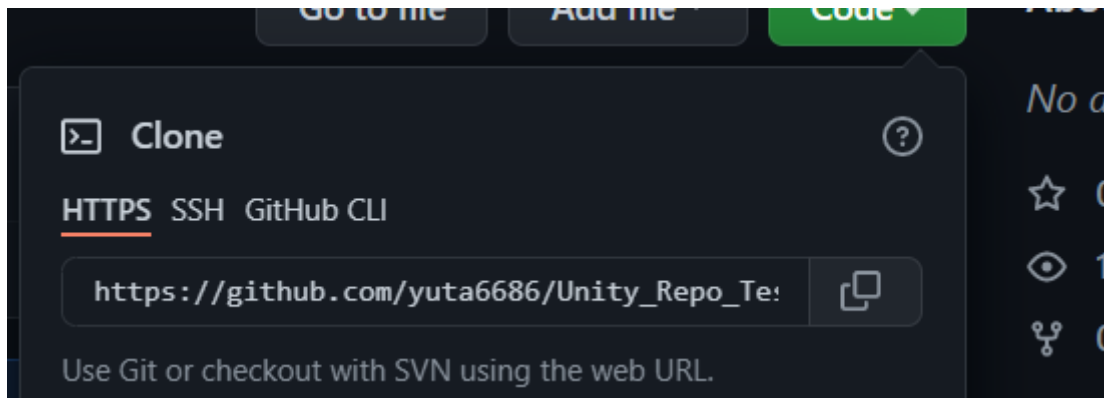
### ▼ 注意事項

- ProgateでGit Bashを使用できるところまでやっておいてください。
- <https://prog-8.com/docs/git-env-win> ← Progateのgit の環境構築のページ

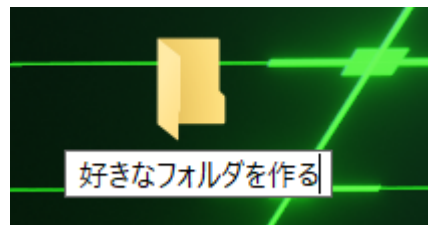
Git Bashを使用できるところまでやっておいてください。

詳細はProgateで。

いつもどおりコピペ



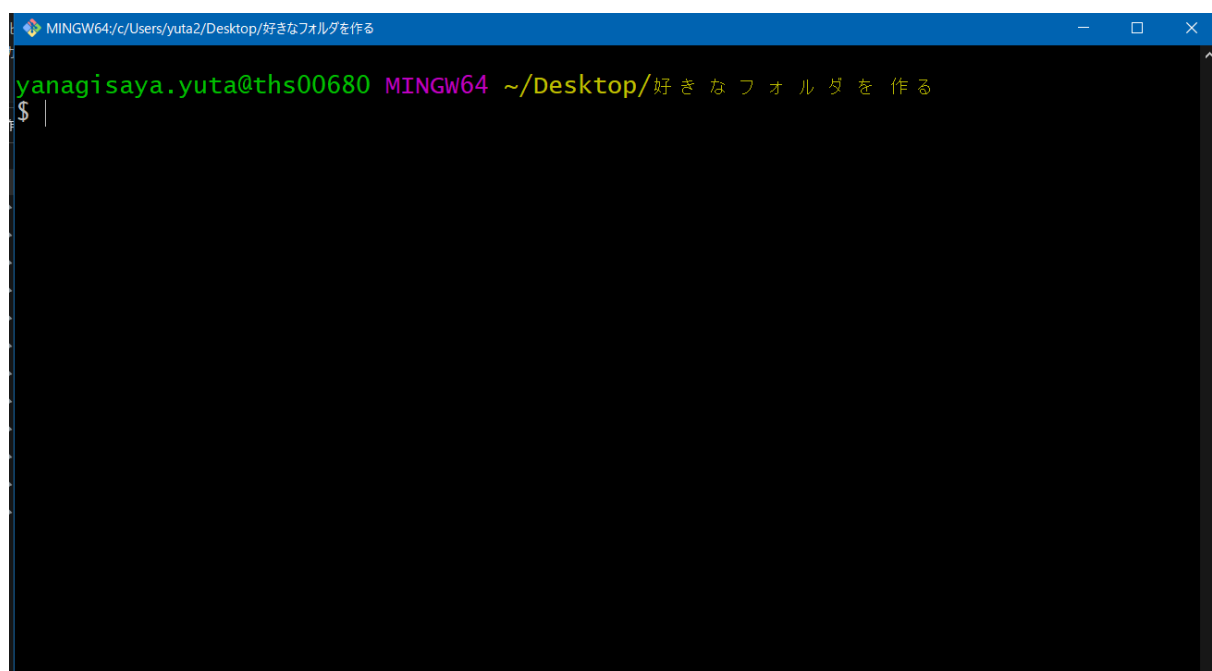
適当にフォルダ作って



開いてGit Bash Hereを選ぶ



気持ち悪い画面が出てきました。



さっそく

git init

と入力してください、初期化です。

```
yanagisaya.yuta@ths00680 MINGW64 ~/Desktop/好きなフォルダを作る
$ git init
Initialized empty Git repository in C:/Users/yuta2/Desktop/好きなフォルダを作る /
.git/
```

続いてクローンしていきます

git clone URL

例) git clone [https://github.com/yuta6686/Unity\\_Repo\\_Test.git](https://github.com/yuta6686/Unity_Repo_Test.git)

▼ Git Bashでペーストできない！

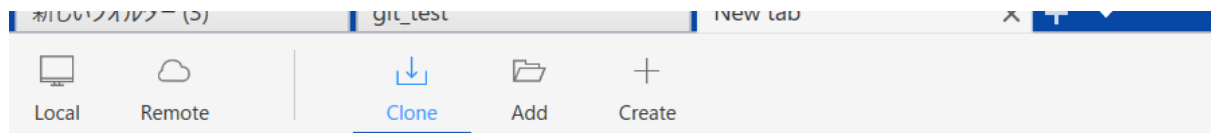
Windows + v を押してください

クリップボード機能をONにしていない場合はONにしてください。

これでコマンドラインでのクローンは完了です。

## ソースツリーでクローン

(こっちのほうが楽です)



# Clone

Cloning is even easier if you set up a [remote account](#)

[参照](#)

リポジトリタイプ: ? パス / URL が指定されていません

[参照](#)

Local Folder:

＞ 詳細オプション

クローン

リポジトリにURLをペースト。

# Clone

Cloning is even easier if you set up a [remote account](#)

参照

リポジトリタイプ:  これは Git リポジトリです

参照

Local Folder:

➤ 詳細オプション

クローン

好きなフォルダを設定してください

参照

クローンボタン押したらクローン完了！