




プロジェクト演習 テーマD

第5回

担当：CS学部 講師 伏見卓恭
連絡先：fushimity@edu.teu.ac.jp

授業の流れ

- 第1回:実験環境の構築 / Pygameの基礎 / Gitの基礎
- 第2回:Pygameによるゲーム開発の基礎 / コード規約とコードレビュー
- 第3回:オブジェクト指向によるゲーム開発 / GitHubの応用
- 第4回:Pygameによるゲーム開発の応用 / 共同開発の基礎
- 第5回:共同開発演習（個別実装） 
- 第6回:共同開発演習（共同実装） 
- 第7回:共同開発演習（成果発表） 

今後の流れ

【第5回】

- ～14時：実装するゲームの相談
 - 共通基本機能と5つ以上の追加機能に分割し、分担を決める
 - コード規約や変数名、クラス設計などの相談
- ～18時：共通基本機能＋担当追加機能の個別実装
 - メンバーで相談しながら実装してもOK
- ～19時：提出物の作成

【第6回】

- ～17時：コードのマージ → ゲーム完成＋README完成
- ～18時：提出物の作成

【第7回】

- ～14時：発表準備（パワポ不要）
- ～18限：成果発表会（デモンストレーション、コードの説明）

サンプルゲームで遊んでみよう

- pygameモジュール内のexamplesにあるサンプルゲームを実行する

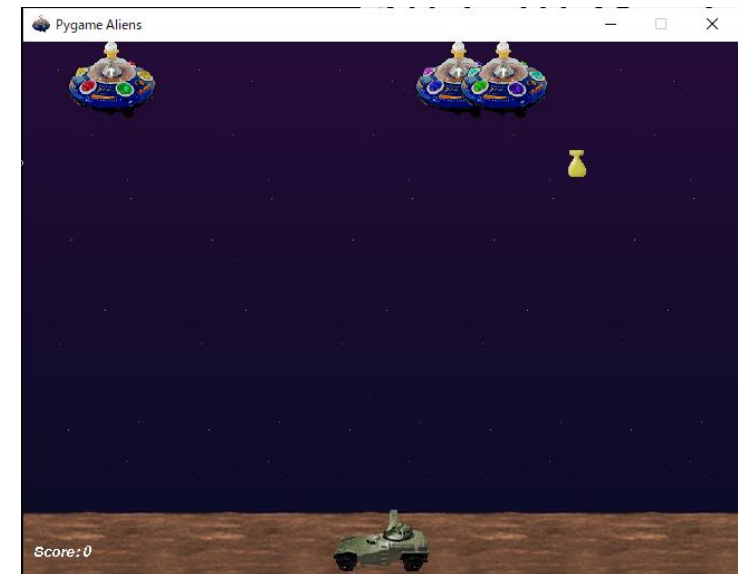
__pycache__	2022/03/29 17:40	ファイル フォルダー	
data	2022/03/29 17:40	ファイル フォルダー	
__init__.py	2022/03/29 17:40	PY ファイル	0 KB
aacircle.py	2022/03/29 17:40	PY ファイル	2 KB
aliens.py	2022/03/29 17:40	PY ファイル	12 KB
arraydemo.py	2022/03/29 17:40	PY ファイル	4 KB
audiocapture.py	2022/03/29 17:40	PY ファイル	2 KB
blend_fill.py	2022/03/29 17:40	PY ファイル	4 KB
blit_blends.py	2022/03/29 17:40	PY ファイル	7 KB
camera.py	2022/03/29 17:40	PY ファイル	3 KB
chimp.py	2022/03/29 17:40	PY ファイル	6 KB
cursors.py	2022/03/29 17:40	PY ファイル	3 KB
dropevent.py	2022/03/29 17:40	PY ファイル	3 KB
eventlist.py	2022/03/29 17:40	PY ファイル	6 KB
font_viewer.py	2022/03/29 17:40	PY ファイル	10 KB

←aliens.pyを実行する

```
(ProjExD) C:\¥Users¥fsmtkys>python -m pygame.examples.aliens
```

↑

「python -m モジュール名」でモジュールのプログラムを実行できる



【参考】pygameに関する情報の確認

- pipコマンドで確認する

```
PS C:¥Users¥admin¥Desktop¥ProjExD2023> pip show pygame
Name: pygame
Version: 2.1.2
Summary: Python Game Development
Home-page: https://www.pygame.org
Author: A community project.
Author-email: pygame@pygame.org
License: LGPL
Location: c:¥users¥admin¥anaconda3¥envs¥projexd¥lib¥site-packages
Requires:
Required-by:
```

←大文字のフォルダ名も
小文字で表示されている

- エクスプローラーでLocationのフォルダを開く
- その下のpygameフォルダ内にpygameモジュールのプログラムがある

注意事項

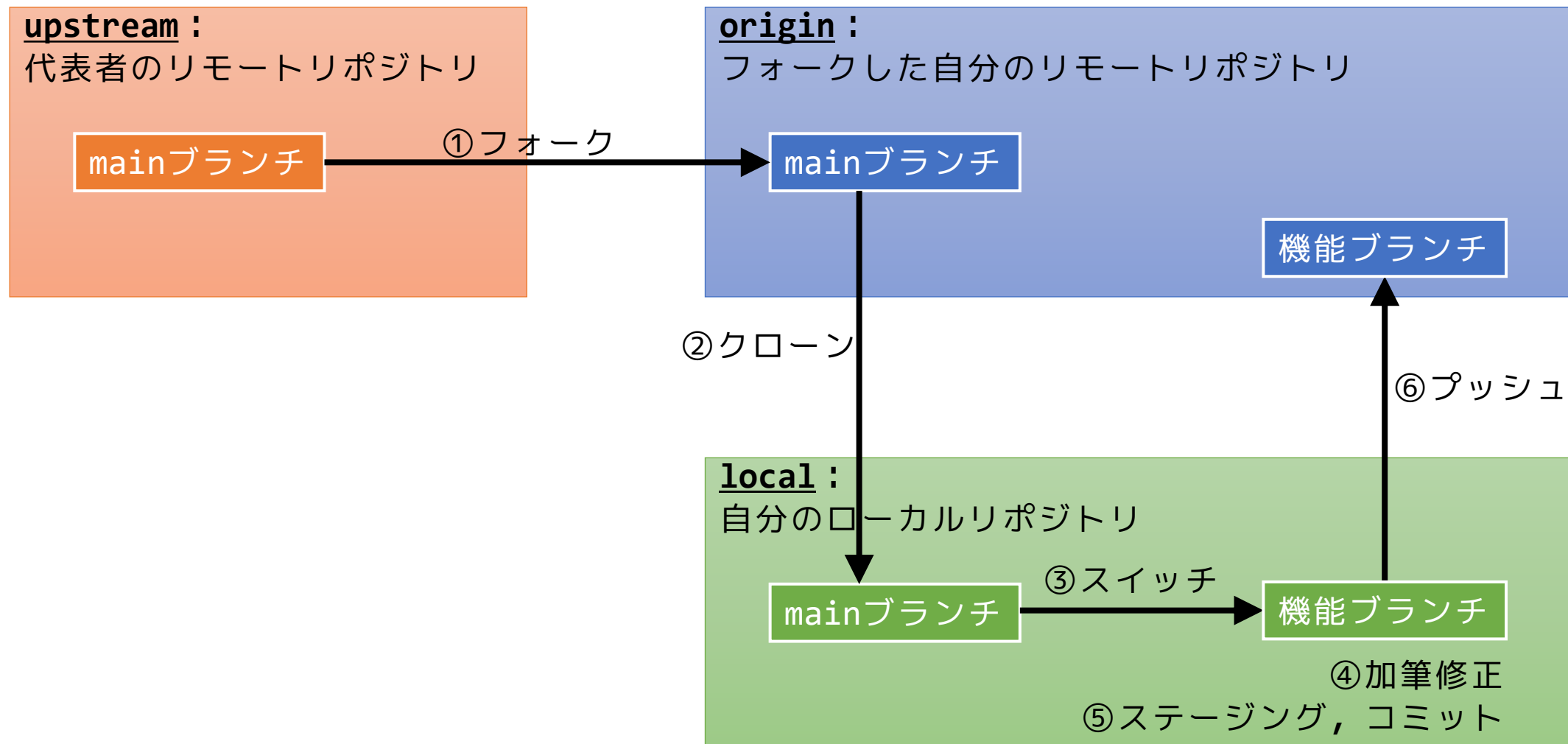
- 条件：pygameを用いること
- 進め方
 - あとでマージすることを念頭に、機能分割、役割分担、コード規約（変数名の付け方など）、クラス設計など念入りに検討・相談して進めること
- ゲームの内容
 - メンバーのプログラミング能力を鑑みて、ゲーム内容を決定
 - 共通基本機能は、背景画像と登場キャラを表示させる程度にした方がよい
 - プロ演の授業で実装したものに機能を追加してもOK
 - ネット上にある記事などを参考にしてもよいが、丸パクリはダメ
 - 差分が小さければ、加点も小さい
 - まったく新しくイチから実装してもOK
 - 複数サイトからコピペしてコードに一貫性がなくなれば、減点になる

作業の流れ（全体像）

1. 代表者を決める ←次回，代表者加点あり
2. 代表者は，メンバーと相談して共通基本機能を実装する
 - 作業フォルダ名：ex05
3. 代表者は，**コード一式**と**README.md**をステージング，コミットする
4. 代表者は，公開リポジトリを作成し，上記コミットをプッシュする
 - 公開リポジトリ名：ProjExD_05
5. その他メンバーは，上記の公開リポジトリをフォークする
6. その他メンバーは，**フォークしたリモートリポジトリをローカルにクローン**する ←要注意
7. ローカルに機能ブランチを作り，担当機能を実装する+READMEを追記する
 - ブランチ名：C0A22XXX/機能名
8. 完成した機能ブランチを自身のリモートリポジトリにプッシュする ←提出物

作業の流れ（イメージ図）

- その他メンバーの視点



README.mdの例

最初に全メンバーで相談して簡単なものを作成し、
本日の個人実装時に各自で追記する

丸焼き豪華豚 ← ゲームのタイトル（仮）

実行環境の必要条件

- * python >= 3.10
- * pygame >= 2.1

ゲームの概要

主人公キャラクター豪華豚（ゴウカトン）をマウス操作により丸焼きにするゲームで、...

ゲームの実装

共通基本機能

- * 背景画像と主人公キャラクターの描画

担当追加機能

- * 丸焼きエフェクト（担当：ふしみ）：バーナーにより豪華豚を丸焼きにするエフェクトに関するクラス
- * キッチンタイマー機能（担当：ぷしみ）：制限時間以内に調理が完了しなかった場合に、豪華豚が脱走する機能
- * 調理機能（担当：ふしみ）：調理器具をキー押下により選択し、豪華豚を調理する機能

ToDo ← 実装しようと思ったけど時間が足りずできなかったことなど

- [] ほげほげ機能
- [] ふがふが関数内の変数名の統一

メモ ← グループメンバーへの連絡など

- * クラス内の変数は、すべて、「get_変数名」という名前のメソッドを介してアクセスするように設計してある
- * すべてのクラスに関係する関数は、クラスの外で定義してある

マークダウン記法

書き方	範例
# 見出し	見出し
- 順序なしリスト	• 順序なしリスト
1. 順序付きリスト	1. 順序付きリスト
- [] チェックリスト	<input type="checkbox"/> チェックリスト
> 引用	引用
太字	太字
斜体	斜体
~~取消線~~	取消線

[link text](https:// "title")	リンク
![image alt](https:// "title")	画像
`コードブロック`	コードブロック
````javascript var i = 0; ````	<pre>1   var i = 0;</pre>
:smile:	

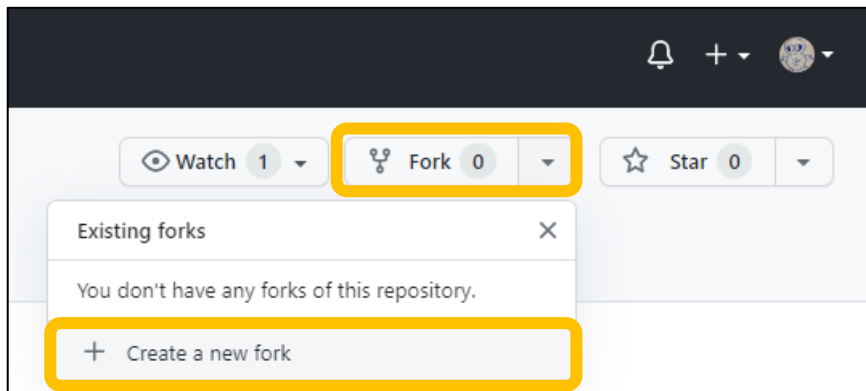
参考URL : <https://qiita.com/tbpgr/items/989c6badefff69377da7>

参考URL : <https://qiita.com/jkjoluvjlajelljicvjzobjieoaid/items/01cd7ff819bc2e69b652>

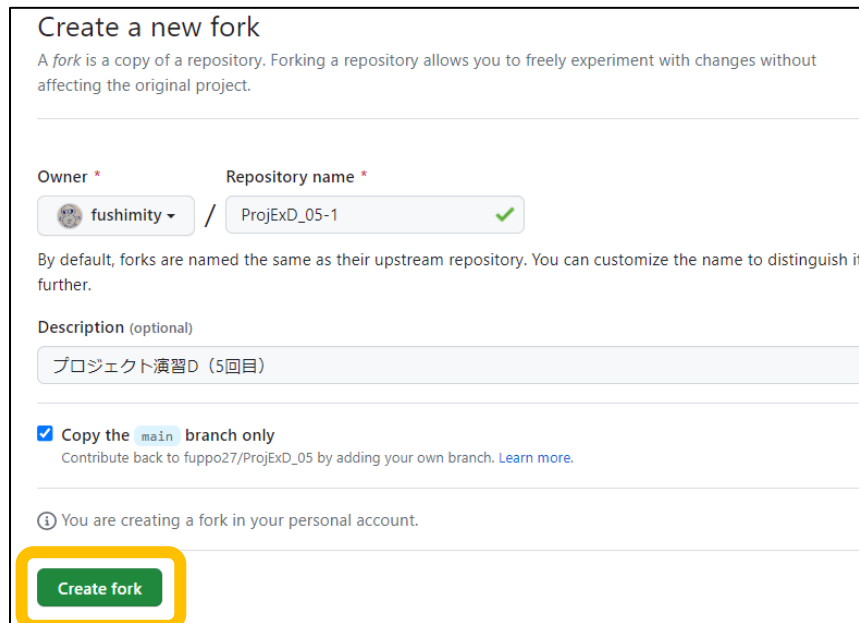
# フォークのやりかた

(全) : 全員  
(代) : 代表者  
(他) : 代表者以外

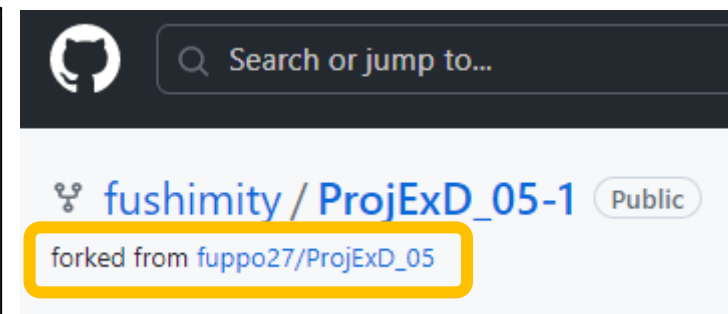
- (他) 代表者のリポジトリを自身のリモートリポジトリとしてフォークする
  - 代表者のリポジトリで, 「Fork」 → 「+ Create a new fork」 → 「Create fork」



↑最終的に, Forkの数がメンバー数-1  
になればメンバー全員に  
フォークされたことになる



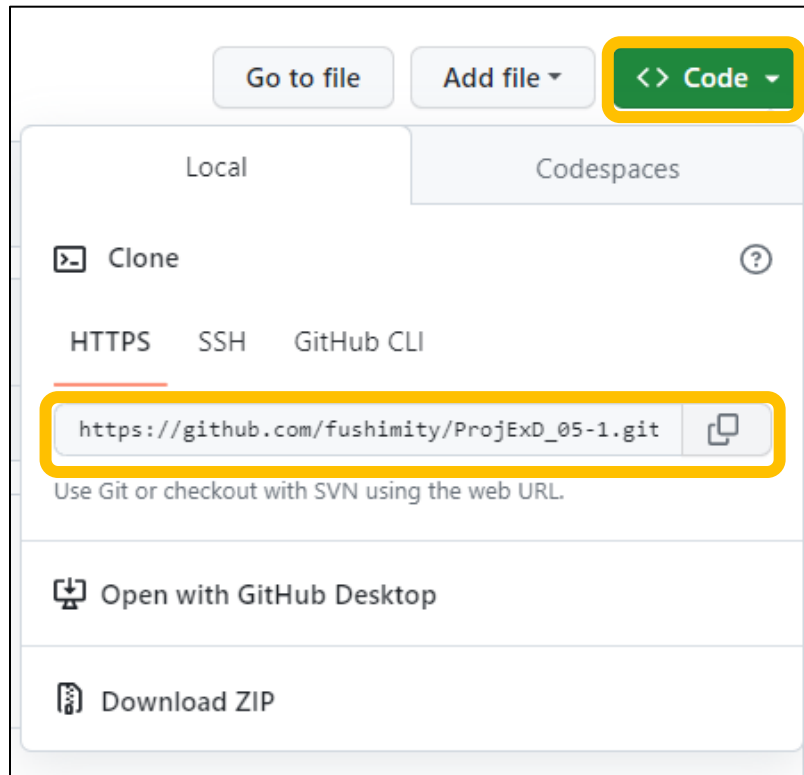
↑既に同名のリポジトリがある場合  
「-1」がつく (or つける)



↑自身のリポジトリ名の下に  
「fused from 代表者リポ名」  
が表示される

# クローンのやりかた

- (他) ProjExD2023にて, フォークしたリポジトリをクローンする :  
`git clone フォークしたリポジトリのURL ex05`



※フォークしたリポジトリ  
= 自分のアカウントにあるリポジトリ

# 提出物

学籍番号は, 半角・大文字で

- ファイル名 : C0A22XXX_kadai05.pdf
- 内容 : 以下の順番でPDFを結合して提出すること
  - README.md  
[https://github.com/fushimity/ProjExD_05/blob/branch/README.md](https://github.com/fushimity/ProjExD_05/blob/branch/README.md)
  - コードの最終版  
[https://github.com/fushimity/ProjExD_05/blob/branch/?????.py](https://github.com/fushimity/ProjExD_05/blob/branch/?????.py)

自分のアカウント名

自分の機能ブランチ名

# チェック項目：個人成果物(15点満点)

1. READMEの説明が適切か [0 -- 2]

- ゲームの概要, 操作方法, クリア条件, 参考サイトのURLなど
- 共通基本機能と自分が担当した追加機能がわかるようにすること

2. 十分な量のコードが書かれているか [0 -- 2]

3. コード内コメントが必要十分か [0 -- 2]

4. 型ヒント, 関数アノテーション, docstringなどは十分か [0 -- 2]

5. コード規約が(ある程度)守られているか [0 -- 2]

6. 関数, クラスなどを用いて, 可読性の高いコードか [0 -- 2]

7. どこかのサイトのパクりでないか [0 or 1]

8. コードは実行できるか [0 -- 2]

9. 提出物不備(PDFの作り方, ファイル名, マージ順)は1点ずつ減点

追加機能部分

コード全体

# 提出物の作り方

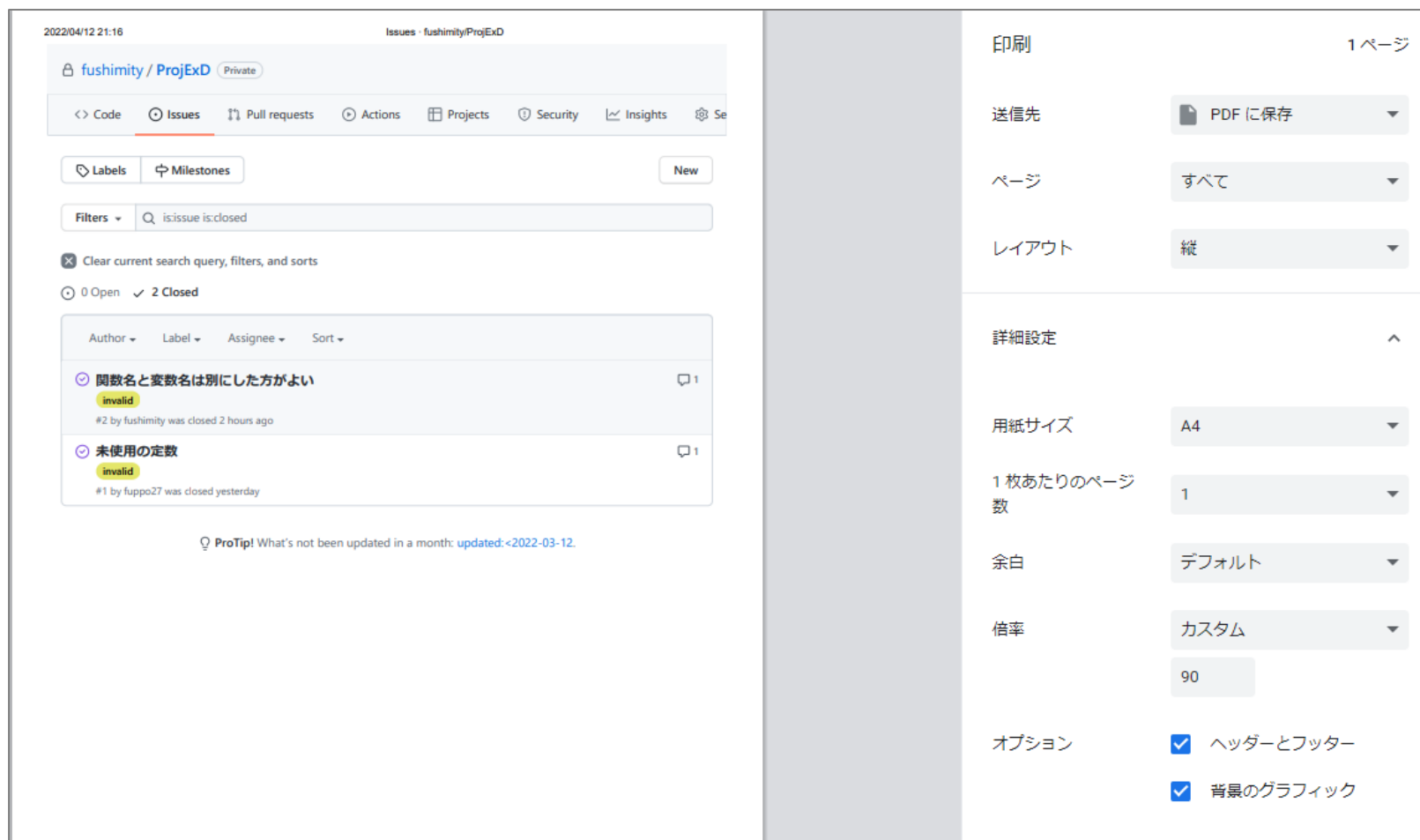
※スクショは認めません。

以下の手順に従ってPDFを作成し、提出すること

1. ChromeでPDFとして保存する（次ページを参照）
2. 以下のURLから各PDFをマージする  
[https://www.ilovepdf.com/ja/merge_pdf](https://www.ilovepdf.com/ja/merge_pdf)
3. ファイル名を「C0A22XXX_kadai05.pdf」として保存する

# ChromeでPDFとして保存する方法

1. 該当ページを表示させた状態で「Ctrl+P」
2. 以下のように設定し、「保存」をクリックする



←送信先：PDFに保存

←ページ：すべて

←レイアウト：ポートレート

←用紙サイズ：A4

←余白：デフォルト

←倍率：90

←両方チェック



# チェックの手順

※基本的に再提出できません。どうしてもの場合は要相談。

1. 受講生：提出物（pdf）を作成し，Moodleに提出する
  2. 受講生：担当TASAに成果物（ゲーム）を見せに行く
  3. TASA：提出物とゲームのデモを確認し，点数を確定する
  4. 受講生：帰る
  5. FSM：近日中に課題と点数を確認し，Moodleに登録する
- 時間内にチェックが終わらなそうな場合は，提出物をMoodleに提出し帰る  
（次回までor次回の3限にチェックされる）

← 時間外提出扱いになり  
割引いて採点するので  
できるだけチェックを  
受けること