## 5. Github Flowにもとづくどうぶつしょうぎ開発

どうぶつしょうぎの開発について，チームの行動を下記にしたがって振り返ること．なお，本シートはそのまま大レポートの一部となる．1チームに複数人の大レポート担当者がいた場合，個々の項目のデータについては共有して構わない．ただし，ブランチ名に関する考察やコミットグラフに関する考察等，考察内容については各自で記述すること．そこまでコピーしていた場合は両方共再提出を指示し，評価を減点する．

## 開発者情報

主開発者及び共同開発者の情報を下記に記述すること．また，自分の学生番号，氏名，Githubアカウントについてはアンダーライン(“\_\_\_”)を引いて分かるようにしておくこと．

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 主開発者 | 共同開発者1 | 共同開発者2 | 共同開発者3 |
| 学生番号 |  |  |  |  |
| 氏名 |  |  |  |  |
| Github |  |  |  |  |

## リポジトリURL

どうぶつしょうぎのGitリポジトリURLを下記URLに続けて記入せよ．

https://github.com/

## 開発状況

どうぶつしょうぎの動作画面をキャプチャし，以下に貼り付けよ．また，全9Stepのどこまでできたかを記述せよ． Step内の特定のTaskまでの場合，どのTaskまでできたかもあわせて記述すること．

[どうぶつしょうぎ動作画面]

[開発進捗状況]

## 開発担当Task数とレビュー担当Task数の確認

チームメンバ毎の開発及びレビュー担当Taskを以下に記述せよ．さらに，それぞれの担当Task数を記述せよ．なお，開発・レビューのどちらか一つでも担当Task数が5に満たないメンバがいた場合は，開発のやり直しが発生することがあるので，チームメンバ及び教員に相談すること．最後に，開発・レビューTaskについて，下記の観点に基づいて考察せよ．

1.特定のTaskについて，同じメンバが開発とレビュー両方を担当していないかどうか

2.開発・レビューTask数がメンバ間で均等になっているか

開発担当，レビュー担当には各Task担当者のGithubアカウントを記述すること．

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 主開発者 | 共同開発者1 | 共同開発者2 | 共同開発者3 |
| 開発Task |  |  |  |  |
| 開発 Task数 |  |  |  |  |
| レビューTask |  |  |  |  |
| レビュー Task数 |  |  |  |  |

[開発・レビューに関する考察]

## ブランチ名の確認

チームで作成したすべてのブランチを下記に記述せよ．Githubのブランチリストをコピペするだけでも構わない． なお，ブランチ名は 2.で記述したGitリポジトリURL末尾に/branches/all をつけることで表示できる．

[チームで作成したブランチ一覧]

次に，ブランチ名を見るだけで，そのブランチでの開発内容が第3者でも理解できるものになっているかどうか，各ブランチ名について考察し，記述せよ．わかりにくい名前になっていると思ったものについては，良いブランチ名とあわせて記述すると良い．実際のブランチでの開発内容についてはコミットを見るか，チームメンバに聞いて確認すること．

[ブランチ名に関する考察]

## コミットグラフ

どうぶつしょうぎリポジトリにおいて， Githubサイトで表示されるコミットグラフをキャプチャし，以下に貼り付けよ．コミットグラフは，2. で記述したGitリポジトリURL末尾に"/network"をつけることで表示できる．その後，下記1~3それぞれについて対応できているかを記述せよ．

1.すべてのブランチがmasterブランチから作成されているか  
2.すべてのブランチがMergeされているか  
3.初期コミット以外でmasterブランチに直接コミットされていないか（Pull RequestによるMergeならOK）

[コミットグラフ]

[ブランチ及びコミットに関する考察]

## Pull Request

すべてのPull Requestについて，下記に記述し，以下の1~3を守れているか考察すること．特に，3のPull Request作成・登録数が2に満たない場合は，開発のやり直しが発生する場合があるため，チームメンバ及び教員に連絡すること．Pull Requestは2. で記述したGitリポジトリURL末尾に"/pulls"をつけることで表示できる．Mergeが終了したPull RequestはClosedというリンクで，MergeされていないPull RequestはOpenというリンクをクリックすることで表示できる．

1.すべてのPRがMergeされているか  
2.PR登録者とレビュアーが別のメンバになっているか  
3.メンバごとのPull Request作成・登録数が2以上である

|  |  |  |  |
| --- | --- | --- | --- |
| PR URL(#) | タイトル | PR作成・登録者 | レビュアー |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

ここで，PR URLにはGithubで割り当てられたPull Requestの#で良い(#1,#2等)．PR作成・登録者及びレビュアーはGithub アカウントを記述すること．

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 主開発者 | 共同開発者1 | 共同開発者2 | 共同開発者3 | 共同開発者4 |
| PR作成数 |  |  |  |  |  |
| レビュー数 |  |  |  |  |  |

[Pull Requestに関する考察]

## 発生したトラブル及び得られた知見，開発における改善点