

# 知能プログラミング演習 II 課題 1

グループ 8

29114116 増田大輝

2019 年 10 月 7 日

## ■提出物 rep1

## ■グループ グループ 8

## ■メンバー

学生番号	氏名	貢献度比率
29114003	青山周平	NoData
29114060	後藤拓也	NoData
29114116	増田大輝	NoData
29114142	湯浅範子	NoData
29119016	小中祐希	NoData

## 1 課題の説明

課題 1-1 Search.java の状態空間におけるパラメータ（コストや評価値）を様々に変化させて実行し、各探索手法の違いを説明せよ。

具体的には、変化させたパラメータと探索結果（最短パス探索の成否、解を返すまでのステップ数、etc.）の関係を、探索手法毎に表やグラフ等にまとめよ。それらの結果を参照して考察を行い、各探索手法の違いを説明せよ。

課題 1-2 グループでの進捗管理や成果物共有などについて、工夫した点や使ったツールについて考察せよ。

課題 1-3 Search.java の探索過程や最終的に得られた順路をユーザに視覚的に示す GUI を作成せよ。

## 2 課題 1-2

グループでの進捗管理や成果物共有などについて、工夫した点や使ったツールについて考察せよ。

### 2.1 工夫した点

今回のチーム開発を行うに当たって、チーム 8 では主に GitHub を活用することによって、進捗管理と成果物共有を行った。まず、はじめに本講義用の Organization を作成した。Organization 名は NITechProgrammingTeam8 とした。

Organization にチームメンバー全員を招待し、第一回課題用のリポジトリ Work1 を作成した。

(Organization の URL : <https://github.com/NITechProgrammingTeam8>)

(Repository の URL : <https://github.com/NITechProgrammingTeam8/Work1>)

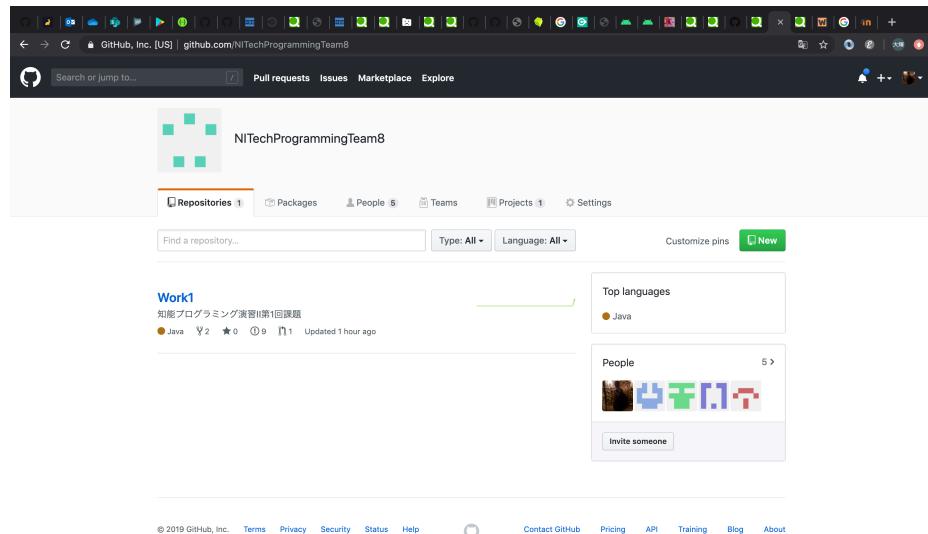


図 1 作成した Organization

次に, 主に利用した機能を以下に列挙する.

**Project 機能** Work1 用のプロジェクトを作成し, 進捗管理を行うためのカンバンとする.

**Issues 機能** 課題をいくつかのタスクに分解することにより, それぞれに担当者を割り当てる.

**MileStones 機能** 個々の Issues を完了するまでの期間を定める.

**PullRequest 機能** 各 Issues と結びつけることによって, タスクと実際の作業を結びつける.

## 2.2 Project 機能の運用と考察

今回は Work1 というリポジトリに対応した Project を作成した. この Project はカンバンと呼ばれ, プロジェクト全体の進行度を視覚的に把握するのに最適である.

今回はタスクの進行度を以下の 3 種類に分類し, Column に登録した.

さらに, PR への割当や Merge といったイベントを完了することによって, タスクの進行度が変化するように自動化を行った.

**To do** これから着手する予定のタスク. PR 発行時に Developing に移動する.

**Developing** 開発中のタスク. Merge 後に Done に移動する.

**Done** 完了後のタスク.

発行した Issues をタスクとしてカンバンにおけるカードとした. 以下に, Project の画像を添付する.

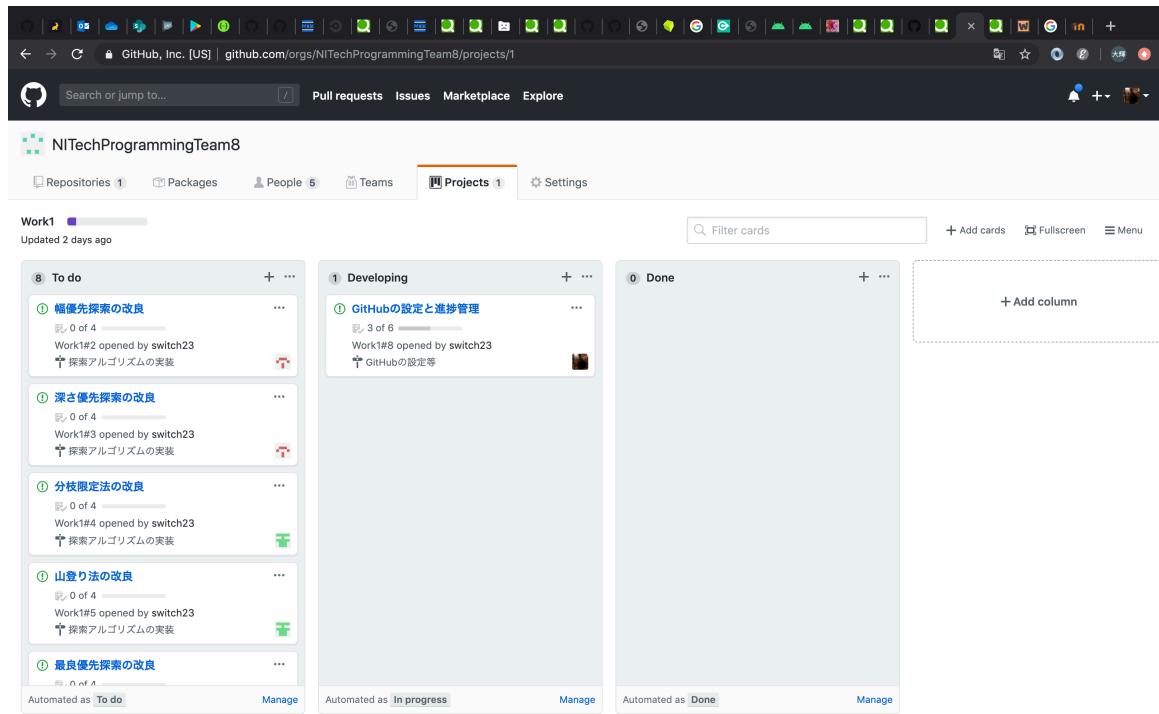


図 2 Project 機能を用いたタスク管理

### 2.3 Issues 機能の運用と考察

今回はタスクに対応した Issues を発行した。それぞれの Issues には最低 1 人の担当者を割り当て,MileStones 機能により期日を設けた。  
次のようにタスクを分解し,Issues を発行した。

幅優先探索

深さ優先探索

最良優先探索

A\*アルゴリズム

分枝限定法

山登り法

GitHub 管理

GUI 設計

また、個々のタスクはさらに粒度の細かい作業に分解され、それぞれに対してチェック欄

を設け、メンバー全員が各自の進捗状況を細かく確認できる仕様にした。以下にIssuesの画像を添付する。

The screenshot shows the GitHub Issues page for the repository 'NItechProgrammingTeam8 / Work1'. The URL is <https://github.com/NItechProgrammingTeam8/Work1/issues>. The page displays 9 open issues, each with a title, author, and a link to view more details. The issues are categorized under various labels such as 'レポートの作成', 'GUIの設計', 'GitHubの設定と進捗管理', 'A\*アルゴリズムの改良', '最良優先探索の改良', '山登り法の改良', '分枝限定法の改良', '深さ優先探索の改良', and '幅優先探索の改良'. The interface includes filters at the top, a 'New issue' button, and sorting options.

図3 Issues一覧

The screenshot shows the GitHub Issue details page for issue #8, titled 'GitHubの設定と進捗管理'. The URL is <https://github.com/NItechProgrammingTeam8/Work1/issues/8>. The page shows the issue was opened by 'switch23' 2 days ago. The issue description is: 'GitHubの各種設定を行い、進捗管理を行えるように整備する'. The 'Assignees' section lists 'switch23'. The 'Labels' section shows 'None yet'. The 'Projects' section shows 'Work1' with 'Developing' status. The 'Milestone' section shows 'GitHubの設定等'. The 'Notifications' section indicates notifications are being received because you were assigned. The sidebar on the left shows a list of tasks: 'GitHubの使い方をレクチャーする', 'タスクを割り当てる', 'Projectを作成する', 'Issuesを発行する', '他サービスとの比較等を行う', and '考案を行う'. The bottom of the page shows activity logs from 'switch23': 'switch23 commented 2 days ago', 'switch23 self-assigned this 2 days ago', and 'switch23 added this to To do in Work1 via automation 2 days ago'.

図4 Issues機能を用いたタスク割り当てとタスク分解

## 2.4 MileStones 機能の運用と考察

今回は個々のタスクを終了すべき期日を MileStones 機能により設定した。設定した MileStones は以下の通りである。

探索アルゴリズムの実装 10/10(木)

GitHub の設定等 10/10(木)

GUI の実装 10/13(日)

レポートの作成 10/13(日)

The screenshot shows the GitHub interface for the repository 'NItechProgrammingTeam8 / Work1'. The 'Milestones' tab is selected. There are four milestones listed:

- レポートの作成**: Due by October 13, 2019. Last updated 1 day ago. 0% complete, 1 open, 0 closed. Actions: Edit, Close, Delete.
- GUIの実装**: Due by October 13, 2019. Last updated 1 day ago. 0% complete, 1 open, 0 closed. Actions: Edit, Close, Delete.
- GitHubの設定等**: Due by October 10, 2019. Last updated 1 day ago. 0% complete, 1 open, 0 closed. Actions: Edit, Close, Delete.
- 探索アルゴリズムの実装**: Due by October 10, 2019. Last updated 1 day ago. 0% complete, 6 open, 0 closed. Actions: Edit, Close, Delete.

図 5 MileStones 一覧

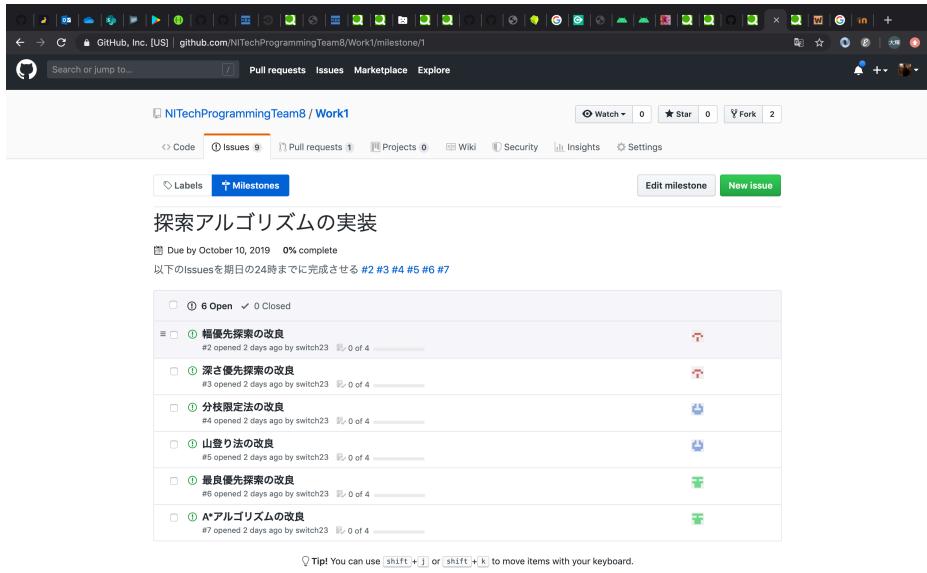


図 6 MileStones 機能を用いた Issues の期日管理

## 2.5 PullRequest 機能の運用と考察

今回は Issues 単位で PR を発行し, 紐づけるものとする.

PR を行うことによって, 変更を master ブランチに Merge する前にレビューを行い, 事前に問題点などを吟味することができる.

また, master との差分によって変更点を視覚的に示すことが可能なため, メンバーの詳細な作業内容を把握しやすく, ソースコードの共有が容易となる.

さらに, 何か議論がある場合には, PR 上で当該コードを指定してコメントを残すことができるため, 具体性が増すメリットがある.

以上のように PR 機能を用いることによって, ソースコードの品質を維持するだけでなく, 円滑なコミュニティを形成することが可能となる.

The screenshot shows a GitHub pull request (PR) diff view. The file being reviewed is `report/29114116.tex`. The changes are color-coded: green for additions and red for deletions. The code itself is a LaTeX document containing a figure and several mathematical definitions and comments. A specific comment in line 99 is highlighted in red: "次回のフィボナッチ数を計算するcalcFiboメソッドの実装をソースコード\ref{src:calcFibo}に示す。". The commit ID shown is 819793a.

図 7 PR の差分詳細

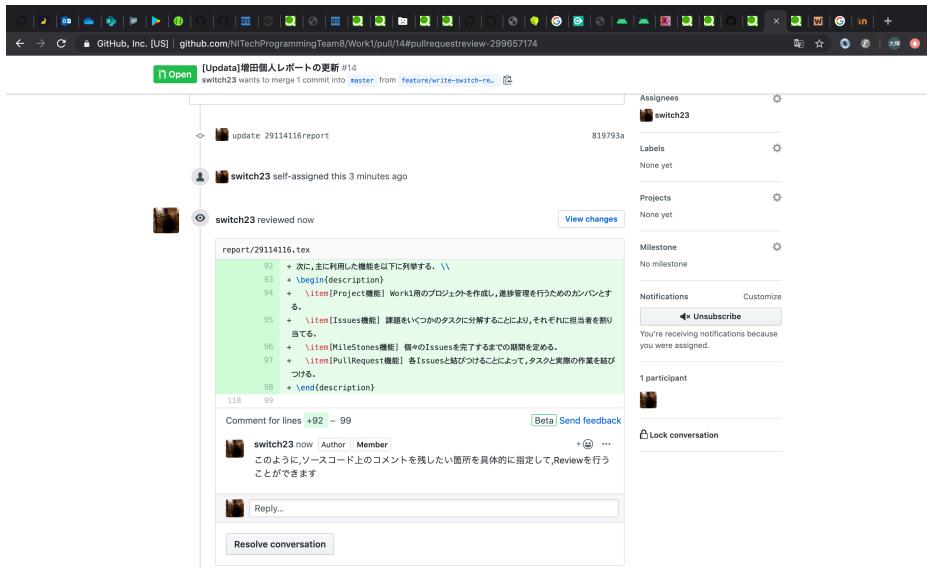


図 8 PR のレビュー機能

## 2.6 チームメンバーへのレクチャー

チーム開発においては必須と言える GitHub だが、概念的な理解が必要かつ多機能であるため、学習コストも非常に大きいというデメリットも存在する。しかし、今後の課題を想

定すると、早い段階で GitHub を導入することにより、メンバー全員に Git 管理の習慣を定着させることができ最善であると判断した。したがって、学習コストというデメリットを克服する必要があると考えられる。私は、そのために基本的な Git コマンドの解説や手順を概念的な理解と共に学習できるように README.md を作成した。以下に、実際に作成したものを見せて示す。



図 9 README.md その 1

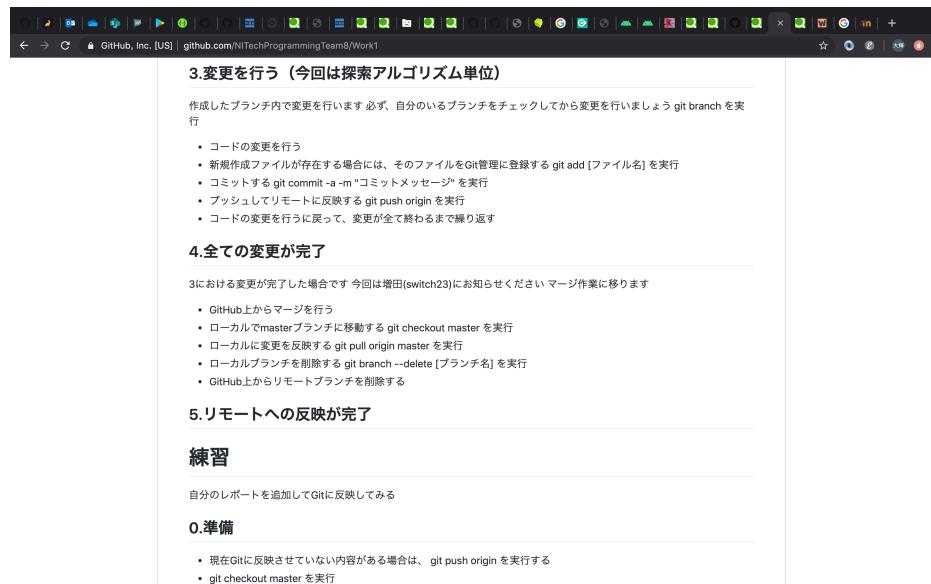


図 10 README.md その 2



図 11 README.md その 3

## 参考文献

- [1] 新入社員におくる GitHub でのプロジェクト管理の初歩 –著：hayato ki  
<https://qiita.com/gumimin/items/63dcb36d4730213bd63a> (2019 年 10 月 7 日アクセス) .