

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>)

また、コミュニケーションやソースコードの共有を目的として、Slack を導入した。

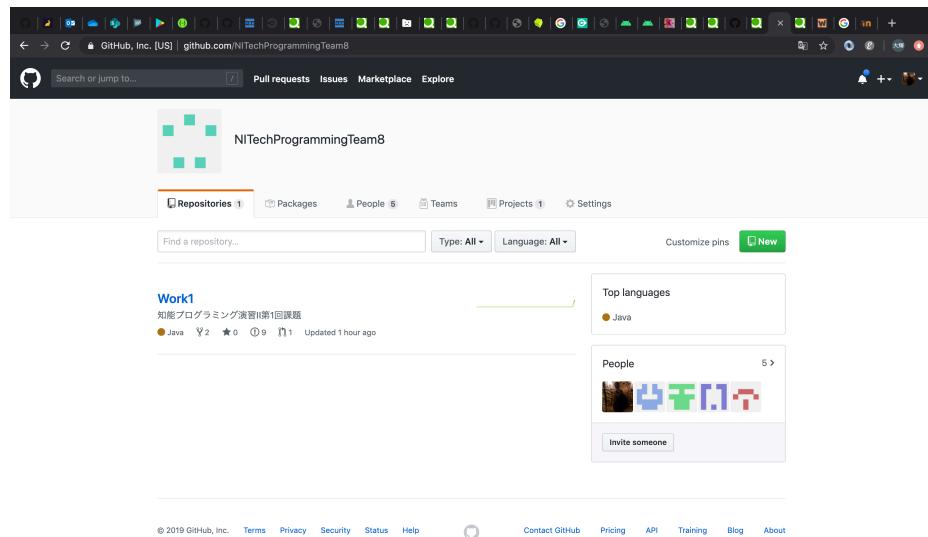


図 1 作成した Organization

2.2 GitHub の導入

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.

2.3 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.

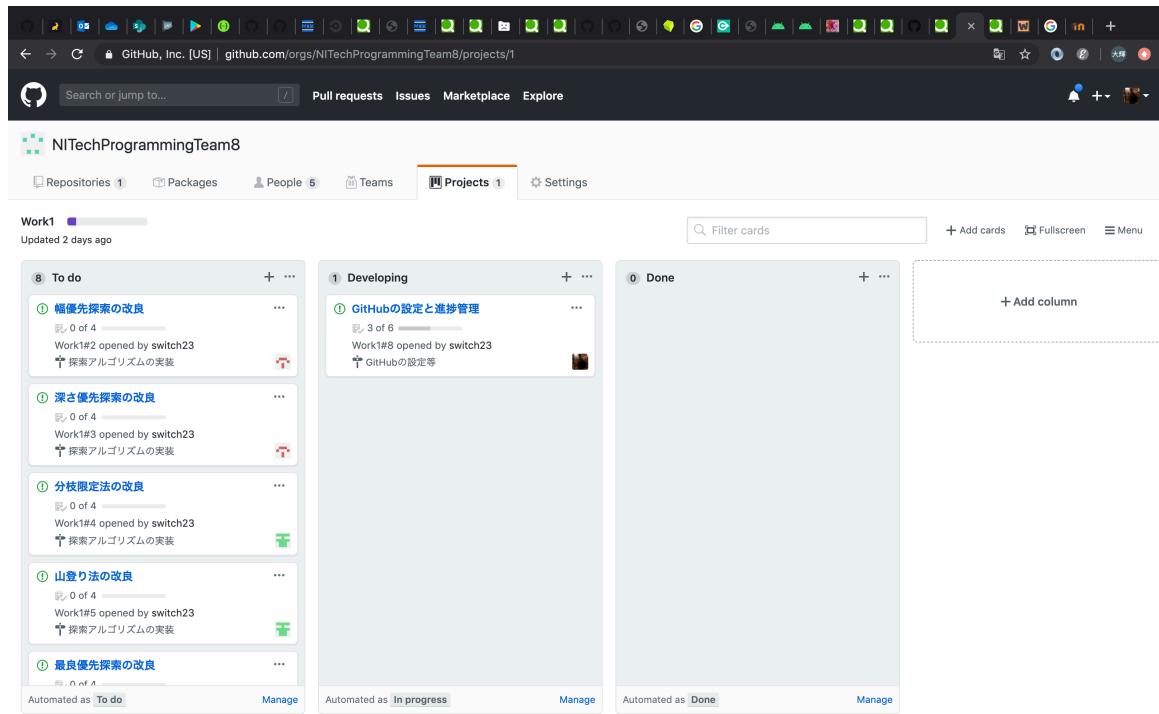


図 2 Project 機能を用いたタスク管理

2.4 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 search bar, and a 'New issue' button.

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 body contains a '要件' section with the text 'GitHubの各種設定を行い、進捗管理を行えるように整備する' and a 'タスク' section with a list of tasks. One task, 'GitHubの使い方をレクチャーする', is checked. The sidebar on the right shows the assignee 'switch23', labels 'None yet', projects 'Work1 (Developing)', and milestones 'GitHubの設定等'. Notifications indicate that the user is receiving notifications because they were assigned.

図4 Issues機能を用いたタスク割り当てとタスク分解

2.5 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 a repository named 'NItechProgrammingTeam8 / Work1'. The 'Milestones' tab is selected. There are four milestones listed:

- レポートの作成**: Due by October 13, 2019. Progress: 0% complete (1 open, 0 closed). Actions: Edit, Close, Delete.
- GUIの実装**: Due by October 13, 2019. Progress: 0% complete (1 open, 0 closed). Actions: Edit, Close, Delete.
- GitHubの設定等**: Due by October 10, 2019. Progress: 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. Progress: 0% complete (6 open, 0 closed). Actions: Edit, Close, Delete.

図 5 MileStones 一覧

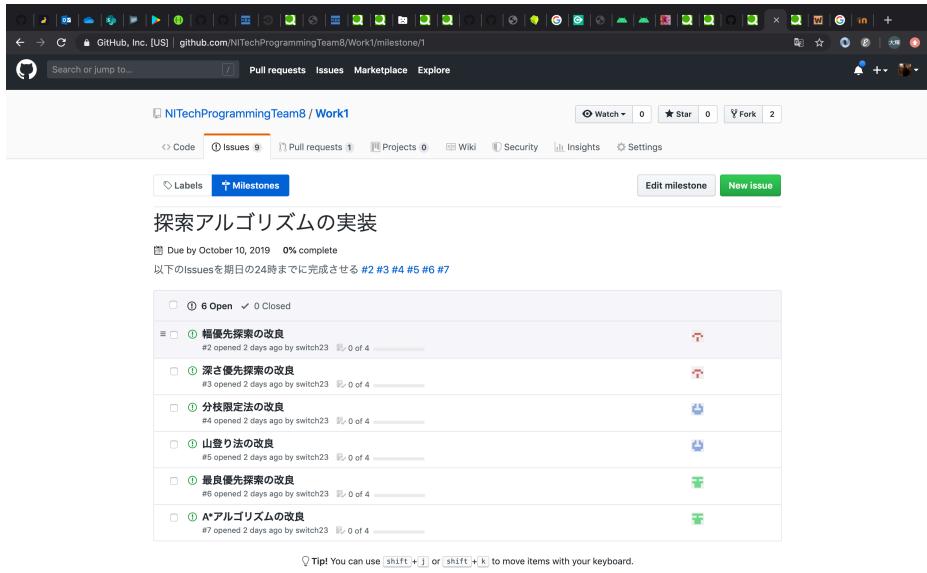


図 6 MileStones 機能を用いた Issues の期日管理

2.6 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.

→ C GitHub, Inc. [US] | github.com/NiTechProgrammingTeam8/Work/pull/14/commits/819793a7fc05bf5b009a80826bdcf9aaeae41098

[Open] 増田個人レポートの更新 #14
Changes from all commits ▾ File filter ▾ Jump to... ▾ Review changes ▾

▼ 178 report/2914116.tex

```
84 + \begin{figure}[!ht]
85 + \centering
86 + \includegraphics[scale=0.20]{git_image/organization_image.png}
87 + \caption{作成したOrganization}
88 + \end{figure}
89
90 - n項目のボタン数を計算するcalcFibメソッドの実装をソースコード[ref(src:calcFib)]に示す。
91 + \begin{lstlisting}[caption=calcFibメソッド,label=src:calcFib]
92 - // n項目のボタン数を計算して返す
93 - int calcFib(int n) {
94 -     if (n <= 1) {
95 -         return 1;
96 -     }
97 -     return calcFib(n - 1) + calcFib(n - 2);
98 - }
99 - \end{lstlisting}
100
101 - isFibonacciソースおよびcalcFibonacciの実装については、グループレポートを参考にされたい。
102
103 - calcFibメソッドは直接引けたプログラム実行時に定義できよろしくないため、mainメソッドおよびFibonacciクラスのコンストラクタを以下のように実装した。
104 - \begin{lstlisting}[caption=main()とFibonacciクラスのコンストラクタ,label=src:main]
105 - public static void main(String[] args) {
106 -     new Fibonacci(Integer.parseInt(args[0]));
107 - }
108
109 - \end{lstlisting}
110
111 - 這次、主に用いた機能を以下に列挙する。\\
112 + \begin{description}
113 + \item[Version(バージョン)] Word 2010 のプロジェクトを作成し、進捗管理を行うためのカバーページとする。
114 + \item[Issues(課題)] 課題をいくつかのリストに分離することにより、それぞれに担当者を割り当てできる。
115 + \item[Milestones(里程碑)] 際に Milestones を作ったまでの期間を定める。
116 + \item[PullRequest(埋怨)] 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づける。
117
118 + \end{description}
```

図 7 PR の差分詳細

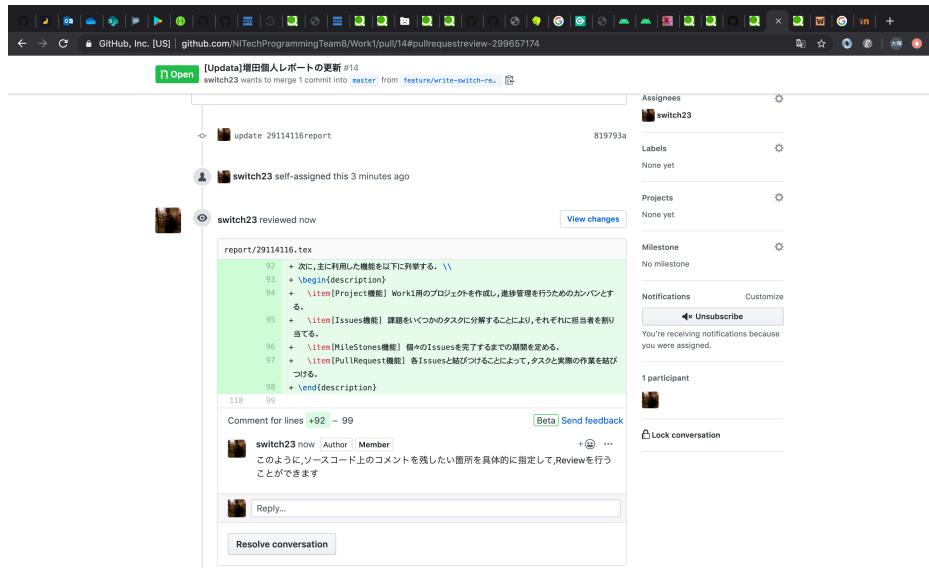


図8 PRのレビュー機能

2.7 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

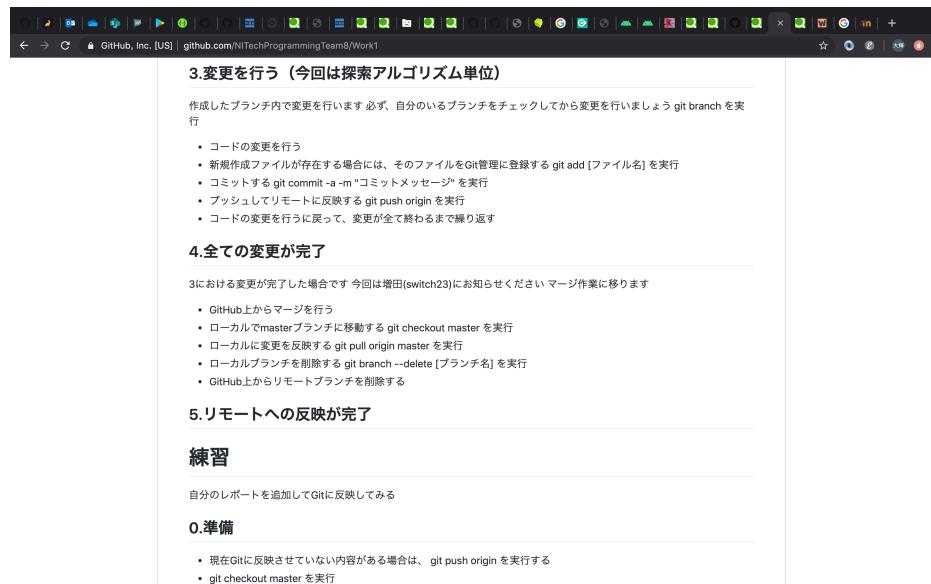


図 10 README.md その 2



図 11 README.md その 3

2.8 Slack の導入

Slack で専用のワークスペースを作成し、複数のチャンネルを作成した。具体的には、今回の課題用のチャンネル#work1 と、GitHub 運用についてのトピックスを扱うチャンネル#githhub を作成した。

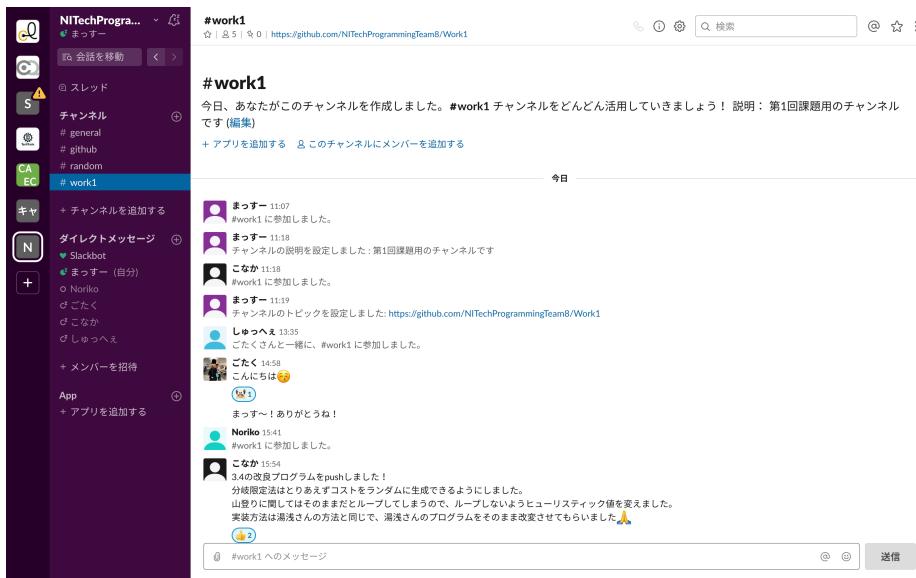


図 12 README.md その 3

2.9 他サービスとの比較検討

今回のチーム課題では、主に GitHub と Slack を導入することとした。この 2 サービスを導入することによって、メンバー間でのソースコード共有と議論の活発化を狙った。そこで、今回導入を検討した他のサービスについても比較検討する。

まず、進捗管理ツールとして代表的な Trello の導入を検討した。Trello の長所としてはガントチャートを作成することで、視覚的に工程を捉えることができる点が挙げられる。さらに、画像を保存や、カードを作成することでメモを残すことができるなど、無料の範囲であっても十分な機能を使用できる。加えて、Chrome の拡張機能によって、カスタマイズが可能である。しかし、今回の課題を行う上でタスクの数は多くはなく、ガントチャート等を活用するほどの規模では無いと考えられたため、導入を見送ることとした。対して、GitHub では、ソースコードの共有を行うと同時にタスク管理を行うこともできるのでこちらを採用することとした。

次に、Zoom の導入を試みた。Zoom を活用することによって、問題発生時やグループメンバー間に GitHub の使い方を説明する際にオンラインで議論を行おうと考えた。しかし、タスクなどが多くなかったことや Slack の導入によって、Zoom が利用される機会はなかった。

2.10 考察

各々のサービスについての考察は上述した通りである。ここでは、全体的な考察について述べることとする。

今回の課題では、全 6 回の課題を通して利用価値があると考えられるサービスを導入し、チーム内で普及させることを目標とした。特に、GitHub は初期の学習コストが高く、慣れが必要とされるため、早めに導入するのが効果的であると考えた。加えて、丁寧に解説することで、基本的な使い方をメンバー全員に周知してもらうことができた。

GitHub や Slack などのサービスを積極的に利用することによって、有意義なチーム開発のスタートを切れたのでは無いかと考えている。

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步 -著：hayato ki
<https://qiita.com/gumimin/items/63dcb36d4730213bd63a> (2019 年 10 月 7

日アクセス) .