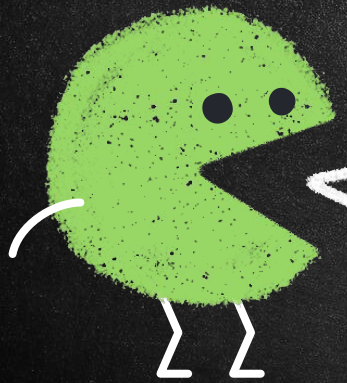
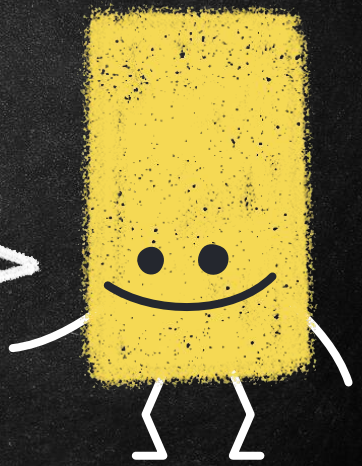
Team C4

s1280124 Ogata Toshiki
s1280147 Saotome Makoto
s1280153 Hagihara Shun





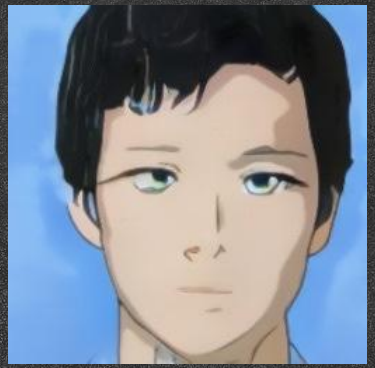
Outline

- Member Introduction
- Development Environment
- Team Mission
- Working Flow
- Exercise Report
- Result (Demo)
- Conclusion

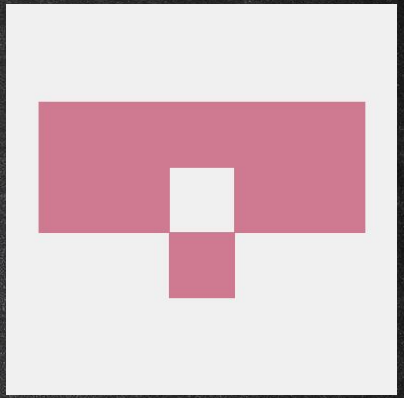
Member Introduction

Member Introduction

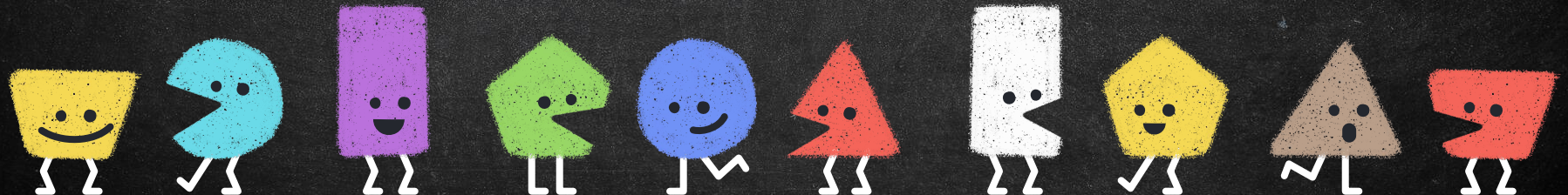
s1280124, Toshiki Ogata
Team Leader
Instruction



s1280147, Makoto Saotome
Bringing Ideas
moodmaker



s1280153, Shun Hagihara
GUI Design
moodmaker



Development Environment

Development Environment

- Machine

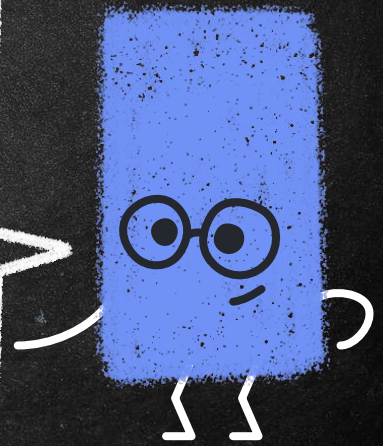
  Macbook(Intel), Macbook(M1)

- Editor

  Eclipse 2022 Full, VisualStudio Code

- Git

  Git or Apple Git



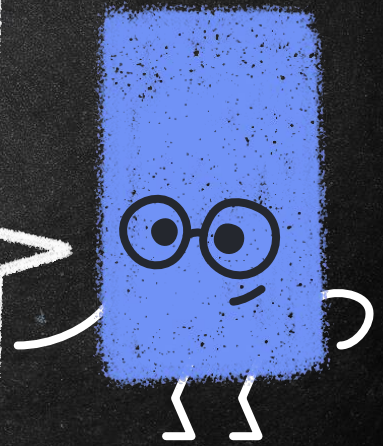
Development Environment

- Language

  Java 17 (Eclipse 2022 Bundled)

- Library

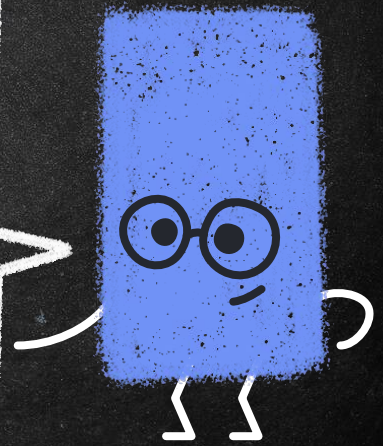
  Swing, JFreeChart



Development Environment

Swing is java language library for GUI application development.

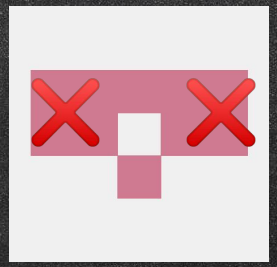
and, We also used JFreeChart because we also wanted to draw graphs.



Team Mission

Team Mission

Our team, with the exception of the team leader, did not have enough experience with Java development and Git.



Team Mission

Therefore, I thought that we must practice the team development methods discussed in the lecture and build a team that can gain solid experience in handling Git and collaborative development, which will be useful for future reference

Team Mission

The goals to be we achieved are

- To create a good team atmosphere
- To be able to complete the GUI application
- To be able to use Java with Eclipse
- To understand how to use Git and how it works

Working Flow

Working Flow

We've decided on a rule.

branch name:

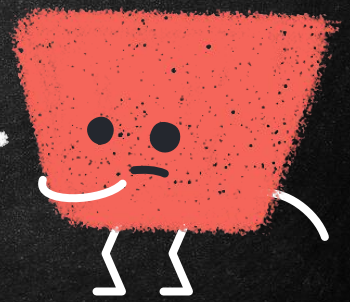
using for chain “-” (e.g. impl-bubble-sort)

commit message:

So that you know what you've done.

using prefix “add:, fix:, update:”

(e.g. fix: 戻るボタンが正しく機能するように修正しました)

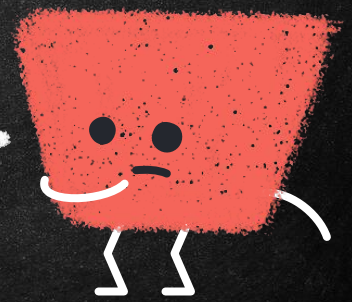


Working Flow

We've decided on a rule.

coding: do not copy and paste from web site
use only as a reference.

git: If you are unsure about a git operation,
don't operate it badly.



Working Flow

The rules we decided on were summarized in README.md so that they can be checked later.

📝 実装していく中での注意

`Javaの基本文法` + `ライブラリの扱い方` + `アルゴリズムの理解力` があれば多分乗り越えられる。

ネット上にあるコードをコピペしない

- コードの書き方がファイルごとに違うと可読性が下がるため
- ネット上にあるものはコードが冗長であったり間違っていたりするため
- そもそも行為としてよろしくない

コードを参考にする程度に留めて、**自分達の方法で**実装すること
わからないことがあればメンバーに聞く

適切な命名を行うこと

書いたコードは他の人が見て使うことがあるので気を配る
以下は良くない例

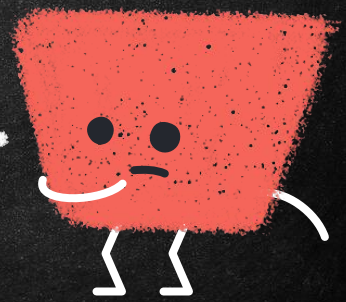
- `a` → 使用用途が一発でわからない
- `numberOfSortedList` → 無駄に長い
- `data` → 中身が推定できない

詳しくは これ を見るといいかもしれない(見た方がいい)
ちなみに、ブランチ切るときも同様。
`-` で区切って2から3節くらいで作るのがベスト

Working Flow

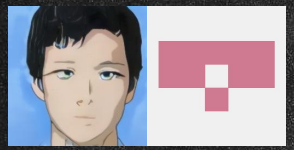
To ensure a smooth process, team members adjusted the amount of work to just finish during class.

Preparation before class and finishing touches at the end of the exercise were done by the team leader.



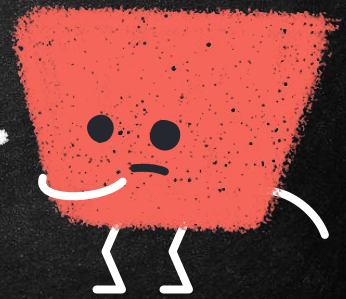
Working Flow

Preparation before class.



Programming...

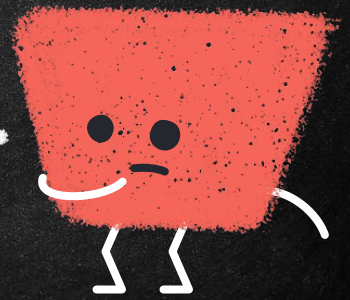
Code review & Merge



Exercise Report

Exercise 1

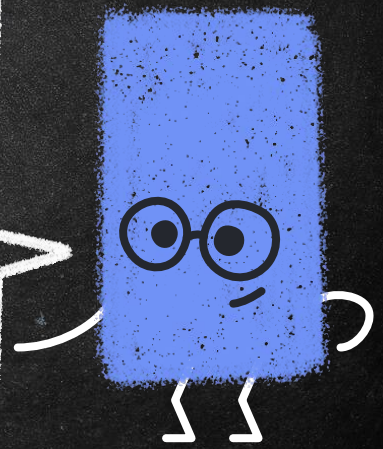
- Decided on a team name.
- Build a development environment.
- Learn how to use Git & GitHub.
  add, commit, push, pull, status



Exercise 2

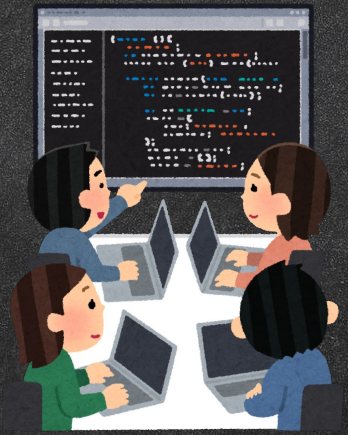
- Continued learning about Git.
  branch, switch and "merge & conflict"

- Learn how to use Eclipse and Java.
  Each implemented a sorting algorithm.



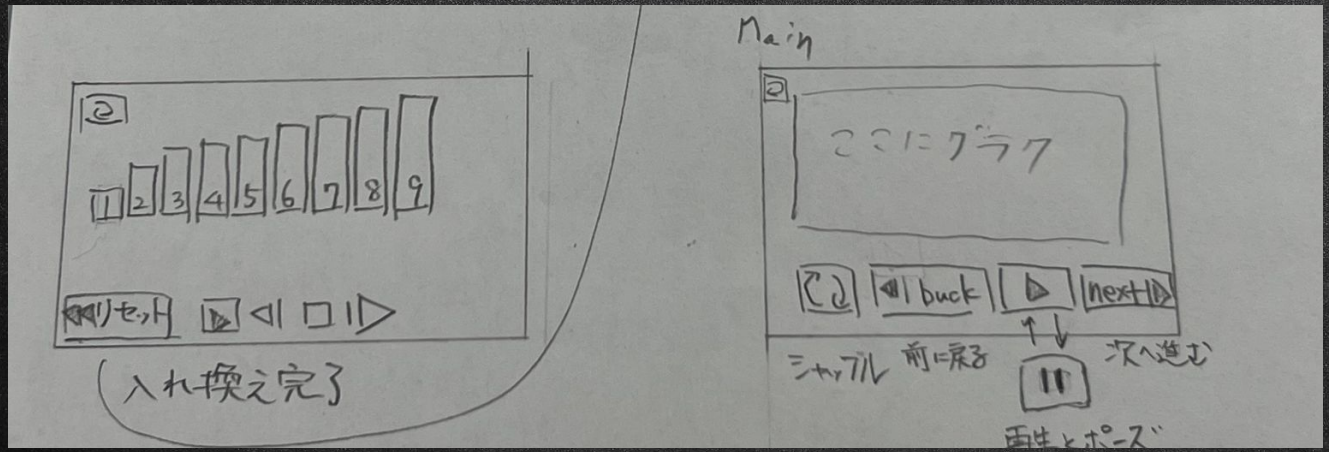
Exercise 3

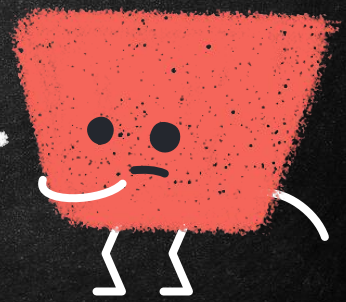
- Implementation of tests using JUnit.
  This was done for the sorting algorithm created in Exercise 2.



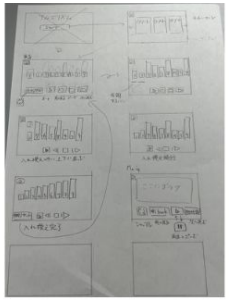
Exercise 4

- We decided through discussion what form of visualization we wanted and what form of GUI we wanted.





Exercise 4

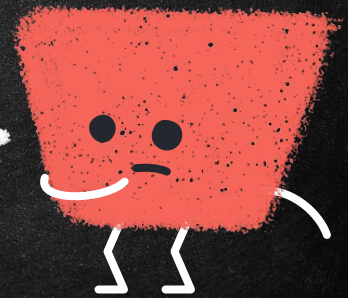


1

2

3

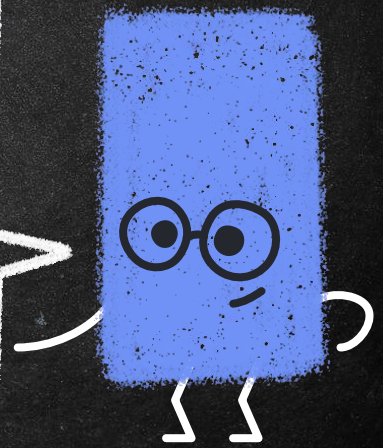
4



Exercise 5

- Creating slides
  Due to insufficient preparation and not being able to start programming from the beginning of the class.

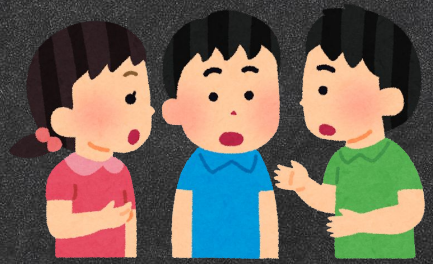
- GUI programming
  Created a button that transitions the screen when pressed.



Exercise 6

- Preparation of presentation slides.

- Look back ourselves.



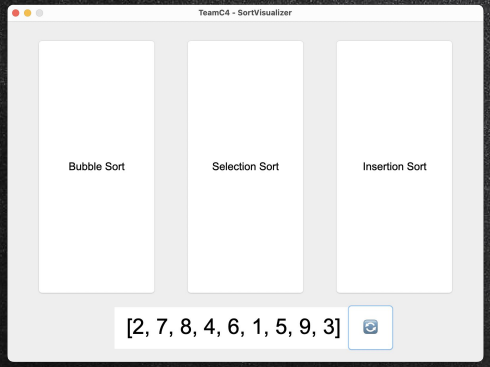
Result (Demo)

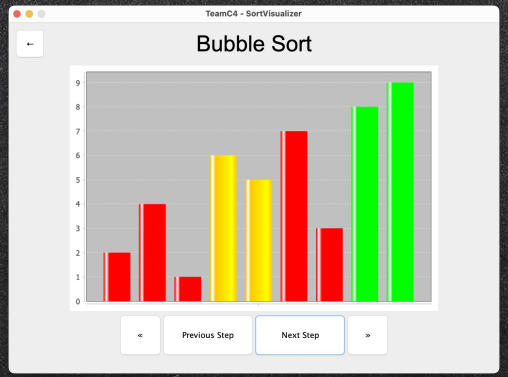
Result (Feature)

- The graphs are colored for each role, making them easy to understand visually.

- Can move forward and backward in a instantly

.





Conclusion

Needs improvement

- Failed to assign appropriate workloads.
  →Gave too little.
- Failed to implement object-oriented advantages.

Good point

- I was able to experience a team leader position.
- It was fun to make simply.

s1280124 O.Toshiki

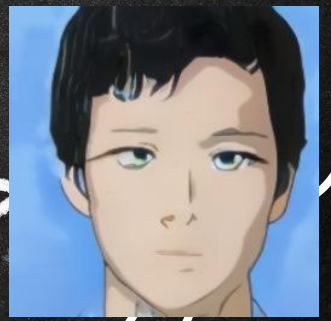
Needs improvement

- Coding level not yet up to working level
- There are many areas of knowledge that are lacking and are left to those with a higher level of technology.

Good point

- The design has been thought through to ensure that it is easy for everyone to understand.
- Absorbed design and development mechanisms and reached a level where they can run on their own.

s1280147 S.Makoto



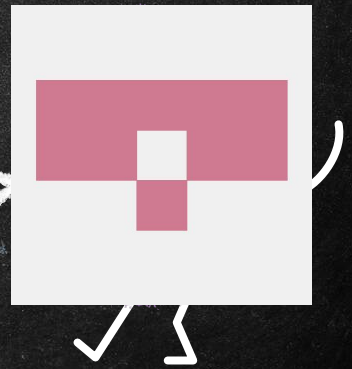
Needs improvement

- Did not fully understand how Git works.
- Programming skills are immature.
- Left too much to the leader

Good point

- Generate ideas and organize opinions.
- Make active discussions.
- learn basic Git operations and the fundamentals of team development

s1280153 H.Shun

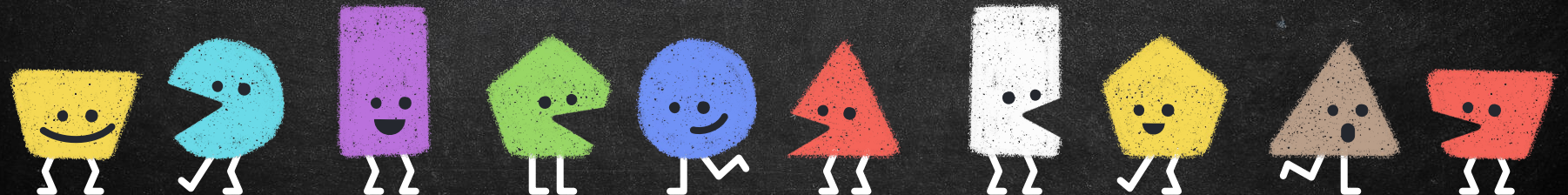


Needs improvement

- There was a difference in experience, which led to a difference in workload.

Good point

- We were able to create GUI application.
- The team got along well from start to finish.



Thank you for listening ! !