

CPSC: 480 Software Engineering Exercise 5

10/18/22

Background

Chess is a popular board game that is among the first to be adapted to run on a computer, and extensive work has been put in to chess AI programs over the past 70 years. Implementing a chess program (with or without an AI) is a common exercise to practice programming skills. With so many implementations, there are many comparable codebases that can be reviewed. In this exercise, you will take three implementations of a chess program and review the three codebases for quality and adherence to a style guideline, then report on which implementation you believe has better code quality and why you think so.

Style Guidelines

The following are some style guidelines that you will be asked to follow. Read <https://web.archive.org/web/20190412011533/http://blog.timoxley.com/post/47041269194/avoid-else-return-early> to better understand items 6-8:

1. Loops and conditional statements should always use curly-brace blocks.
2. Usage of whitespace and indentation should be consistent throughout.
3. Variable names should be descriptive, and consistent in casing.
4. Code that is not clear should be commented (in a consistent manner).
5. Functions and lines should all fit on one screen of a standard display.
6. The main code path should be done at the lowest level of indentation.
7. Return as soon as your function cannot do any more meaningful work.
8. Avoid assigning values to variables that will never be used or returned.
9. Code style should be consistent throughout and follow other best practices where possible. Label code that you think lacks quality even if it doesn't directly violate one of the standards above.

Exercise

You will complete this exercise individually but you may consult with your project teammates. Clone the exercise GitHub repository, which contains zipped copies of other GitHub users' repositories implementing chess in C++. Create a branch named with your UANET id, unzip the source code, commit all the unzipped files to your branch, and push your branch to GitHub. The below commands will accomplish this (replace jdk72 with your own id).

```
git clone https://github.com/kilgallin/SWEF22-Exercise-5
git checkout -b jdk72 #checkout -b both creates a branch and checks it out
unzip chessRepos.zip
git add *
git commit -a -m "jdk72 codebase"
git push --set-upstream origin jdk72 #Need to specifically push your branch
```

In GitHub, navigate to the pull requests tab and select "New pull request". Keep main as the base and choose your branch in the second dropdown, and create the pull request. Go to the "files changed" tab in your PR and begin examining the changes as described below. Comment on a line by hovering over it and clicking the blue "+" at the beginning of the line. After typing the first comment, select the option to "Start a review". When you are done, submit by clicking "Finish your Review->Submit Review". Although in this case you are creating *and* reviewing the pull request, this will simulate a scenario where you are reviewing a pull request from a teammate.

Identify lines of code in each implementation that violates the style guidelines above. For each violation, add a comment on the line explaining the violation and how it could be corrected, as if you were commenting to the author of the code. After you have finished identifying violations, add an overall comment on the pull request (see the "Conversation" tab to the left of "Files Changed"), reviewing the codebases as a whole and explaining which you think has the best code quality and which has the worst code quality, and why you rank them that way.

Submission

Publish the review by **Friday, Oct 22, 11:59 PM**. This will be considered your submission. Grading will be 25% per implementation and 25% for comparison.