



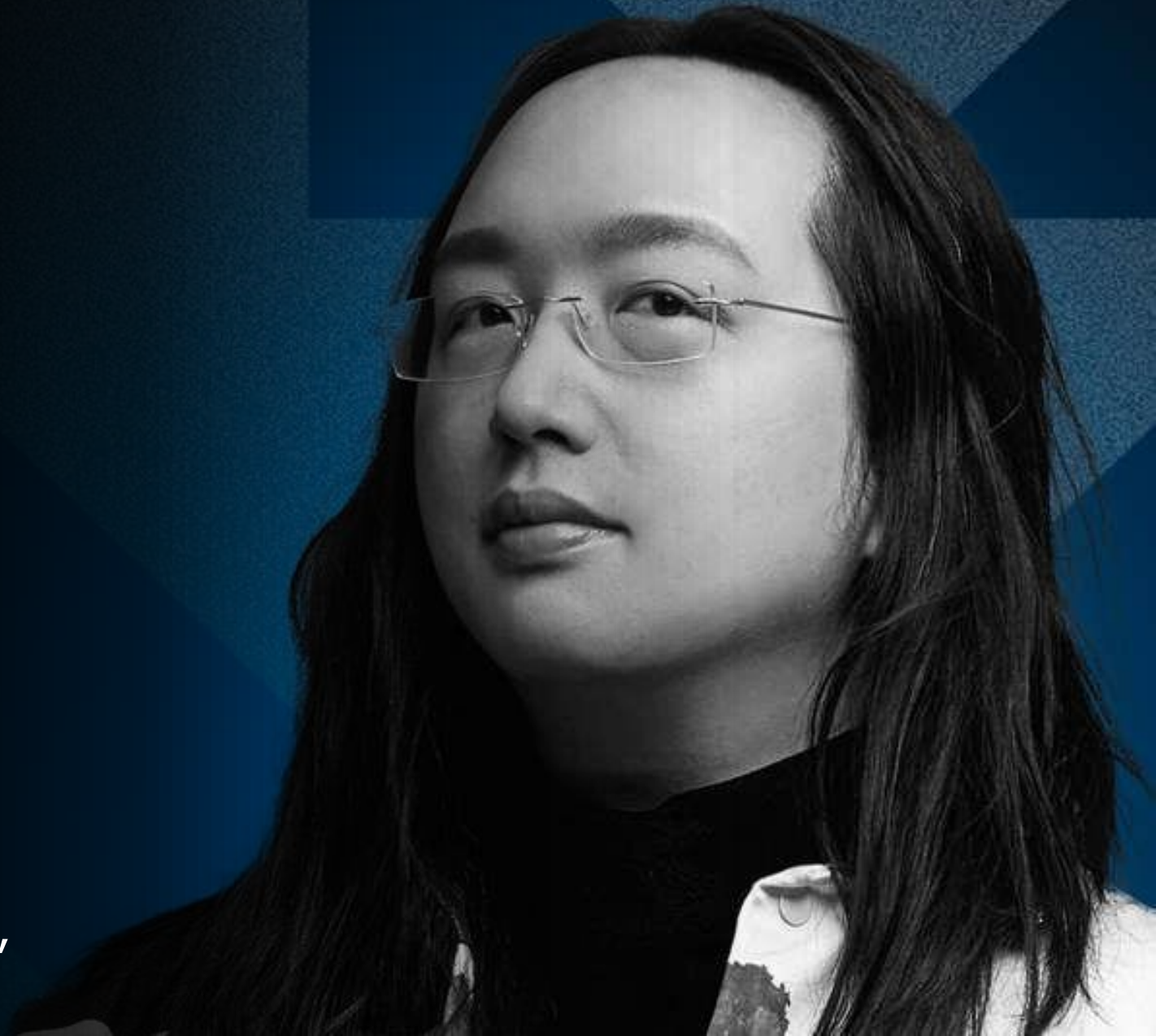
Code Review Exercise

JD Kilgallin

CPSC:480

10/19/22

Photo: Audrey Tang, Business Insider
Known for: Perl 6, Perl code signatures,
over 900 released Perl projects, free
software globalization, software
accessibility, radical transparency initiatives,
Taiwan's first Minister of Digital Affairs



Code Review Exercise

JD Kilgallin

CPSC:480

10/19/22

Notes

- Standups typically occur *every workday*. "I'm going to finish all my tasks for this sprint" is not realistic. Remember you might only get 5 hours/day to plan and execute your exact code modification, test, debug, and check in.
- Poll: Would it help to designate an official team lead to organize project work? What about a technical lead in addition to a team lead?
- Project 3 is posted; due Sunday, November 13, with the coding and code reviews to be complete by Sunday, Nov 6.

Code reviews exercise

- Clone github.com/kilgallin/SWEF22-Exercise-5; create individual branch.
- Unzip code with three implementations of chess in C++ (about 1200 total lines each) and commit/push **on your branch. Committing or pushing to main will result in a penalty.**
- Issue pull request to main with the unzipped code and start a review.
- Read [Avoid Else, Return Early. Tim Oxley. 2013. "Probably Wrong" blog.](#)
- Review code in each implementation as if it was someone else's PR.
- Compare the code quality of the three implementations and report on which you like best/worst, and enter as code review comment.
- Submit assignment by finishing your review.

Git Commands

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

Style Guidelines

- Loops and conditional statements should always use curly-brace blocks.
- Usage of whitespace and indentation should be consistent throughout.
- Variable names should be descriptive, and consistent in casing.
- Code that is not clear should be commented (in a consistent manner).
- Functions and lines should all fit on one screen of a standard display.
- The main code path should be done at the lowest level of indentation.
- Return as soon as your function cannot do any more meaningful work.
- Avoid assigning values to variables that will never be used or returned.
- 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.

References

- [Taiwan's new digital minister. Echo Huang. Aug 2016. Quartz.](#)
- [Avoid Else, Return Early. Tim Oxley. 2013. "Probably Wrong" blog.](#)
- [Google Style Guides. Google. 2008-2022.](#)
- [Coding Style. Charlie Wong et al. April 2019. Read the Docs.](#)
- *Reading for next lecture: Pressman Ch 14*