# ChooChoo the Checklist Tool

Lucy Whalley ORCID [1]

1 Department of Mathematics, Physics and Electrical Engineering, Northumbria University, Newcastle upon Tyne, UK

Summary

ChooChoo is a **ch**ecklist **too**l for educators and students. ChooChoo encourages learning through student self-assessment, class collaboration and diagnostic teaching. Students and instructors interact with ChooChoo by writing comments in a Github issue thread. This makes ChooChoo especially relevant to those who already teach using Github. Although not its primary purpose, ChooChoo can also be used to introduce your students to Github and basic concepts in version control. ChooChoo is free to use and adapt under a Creative Commons License.

Statement of Need

Online repositories such as Github are increasingly being used to host open-source educational content. This content is often designed to be used in both a formal education setting (e.g university), and more informally (e.g. at home with a student working through at their own pace). In addition, adult students are often approaching content with various levels of prior expertise (Jordan et al., 2018); this is especially relevant for more computation-focused courses, where the level of programming experience can vary widely (Dawson et al., 2018).

In such cases, a certain amount of self-directed, independent learning is required. Self-assessment, where students independently assess their current level of understanding, is a key aspect of this (Brown & Harris, 2014). Checklists in particular can be used by students to monitor and track their progress through a couse (Makram et al., 2022). However there are no checklist tools for education which can be easily integrated into a Github repository. The primary purpose of ChooChoo is to fill this gap.

Target Audience

ChooChoo is primarily aimed at people who already use Github for their teaching, and who would like to integrate a simple tool for student self-assessment. It can be integrated into an existing Github repository, or templated as a standalone project. As the instructor must write the checklist specific to their course (if not cloning an existing ChooChoo project), ChooChoo is a domain agnostic tool.

ChooChoo is based around Github and, in particular, Github Issues. In addition, instructors use the yaml markup language to specify the ChooChoo checklist items, links to learning resources and settings. The self-assessment question bank and tutorials (both optional features) are written using Markdown and/or Jupyter Notebooks. These tools are widely used within the research computing community, and familiarity with them will make adopting ChooChoo more straight-forward. Importantly, ChooChoo does not require any local installations; everything is installed and ran remotely via Github.

ChooChoo features

Checklists and student self-assessment

- Automatically generate student checklists as a Github Issue
- Link checklist items to tutorials (external or deployed using ChooChoo)
- Link checklist items to questions stored in a question bank (built and deployed using ChooChoo)
- Generate personalised question sets for e.g. incomplete checklist items

Diagnostic teaching

- Generate and progress plots to visualise class progress

Class collaboration

- Students can propose new questions for the question bank
- Class votes to decide which questions are accepted into the bank

Software design

At the core of ChooChoo is a Python 3 package (https://github.com/lucydot/ChooChoo). The package dependencies are the matplotlib Python library for plotting (Hunter, 2007), PyGithub to interact with Github repositories, python-dotenv for reading environment variables and pyyaml for reading and writing yaml files.

Students and instructors interact with ChooChoo by writing ChooChoo commands in a Github issue thread. Github actions are used to process and respond to each comment. This is done remotely on the Github servers and consists of several steps: (i) parse the issue comment; (ii) verify permissions; (iii) install ChooChoo; (iv) run the appropriate ChooChoo command; (v) update repository and/or issue thread; (vi) build and publish the website. The ChooChoo configuration files and Github action workflows necessary for this are available in the ChooChoo-template repository.

Automatic tests are ran in a dedicated testing repository. This is a fork of the ChooChoo-template repo, and is updated daily with any changes made to the upstream repo. Once updated, a suite of tests are ran using Github actions. The ChooChoo documentation is automatically built and deployed using mkdocs and Github actions.

Integration into teaching

The first step is for the instructor to setup a ChooChoo repository. The length of time this takes will depend on the number of checklist items, questions and tutorials you are setting, and familiarity with the tools used (Github, yaml, markdown). In many cases it will take one hour or less for a basic implementation; a diagram outlining the setup steps is given in Figure 1.
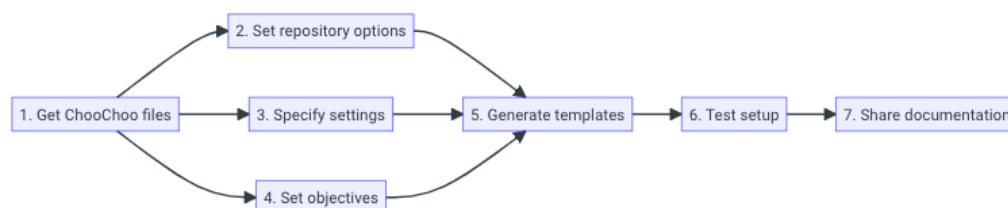

Figure 1: Diagram outlining the initial steps needed to setup a ChooChoo repository.

Once setup is complete, students can generate their own personalised checklists by raising a Github Issue. In addition, students and instructors can use ChooChoo commands in the issue comment thread to monitor class progress, propose new self-assessment questions and generate personalised question sets. For more detailed information about setup and usage please see the ChooChoo documentation.

ChooChoo has been used to teach first and second year undergraduate students on the physics degree programme at Northumbria University (UK). It had particular success as a tool to help second year students re-cap basic computing skills learnt during their first year. An example checklist from this course can be found in ChooChoo examples.

Those who adopt ChooChoo for their own teaching are encouraged to share their checklists, question banks and tutorials for others to use and build on. Over time, we hope this will form a growing collection of ChooChoo repositories.

Acknowledgements

References

Brown, G. T. L., & Harris, L. R. (2014). The future of self-assessment in classroom practice: Reframing self-assessment as a core competency. *Frontline Learning Research*, *2*(1). https://doi.org/10.14786/FLR.V2I1.24

Dawson, J. Q., Allen, M., Campbell, A., & Valair, A. (2018). *Designing an introductory programming course to improve non-majors' experiences* (pp. 26–31). Association for Computing Machinery. https://doi.org/10.1145/3159450.3159548

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Jordan, K., Michonneau, F., & Weaver, B. (2018). *Analysis of software and data carpentry's pre- and post-workshop surveys*. Zenodo. https://doi.org/10.1145/3159450.3159548

Makram, A. M., Wang, J., Vaghela, G., Gala, D., Quan, N. K., Duc, N. T. M., Imoto, A., Moji, K., & Huy, N. T. (2022). Is checklist an effective tool for teaching research students? A survey-based study. *BMC Medical Education*, *22*(1), 561. https://doi.org/10.1186/s12909-022-03632-z