

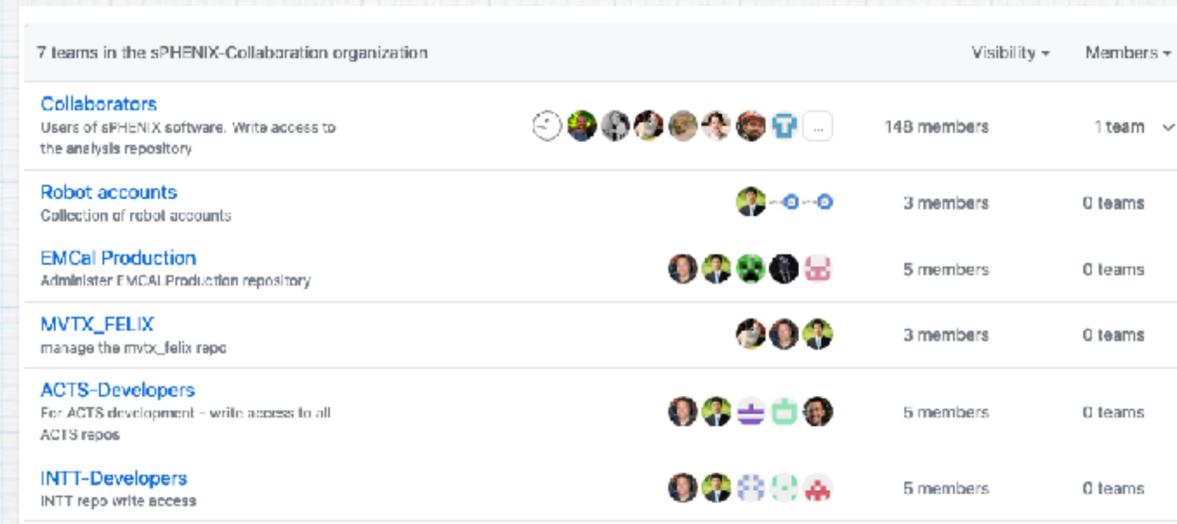
GitHub repositories

G. Nukazuka (RBRC)

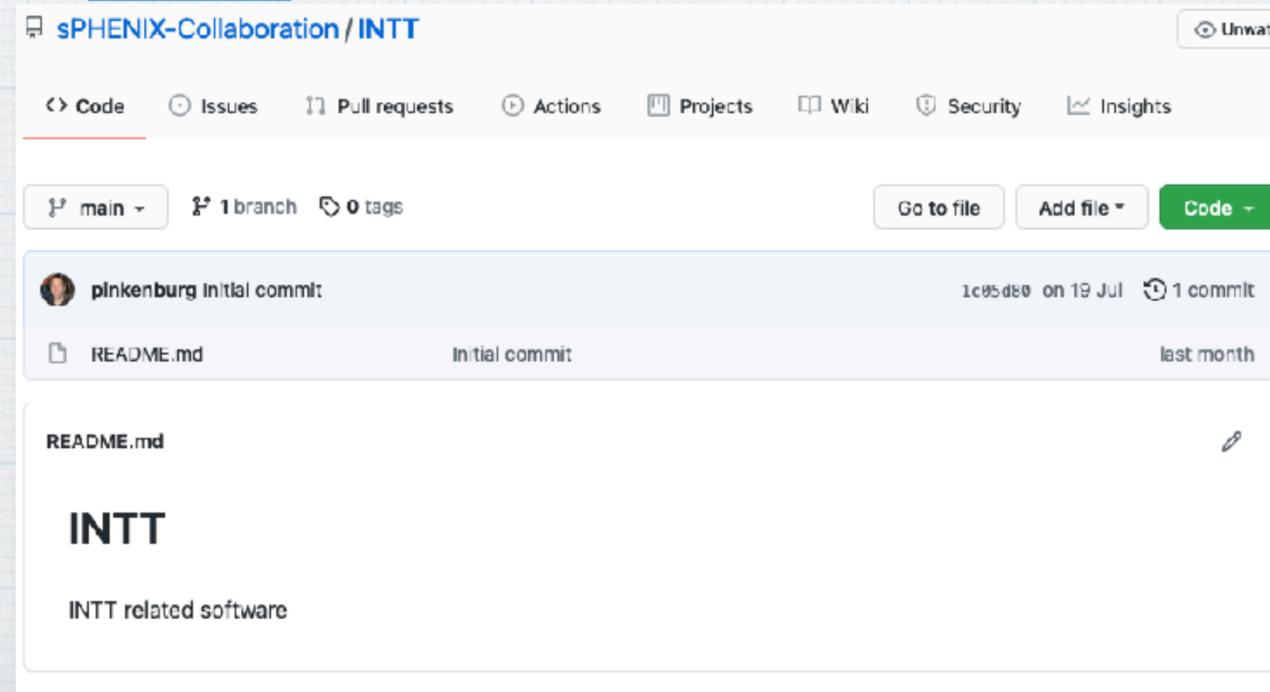


GitHub management

- Our team “[INTT-Developers](#)” was made in the sPHENIX-Collaboration



- Our repository “[INTT](#)” was made at the same time



Basically we are ready to use GitHub.

GitHub management

- Projects, codes to be managed (as far as I remember):
 - General macros for data from the testbench
 - Ladder classification
 - Nevis GUI
 - FPGA codes for FEM and FEM-IB
 - FPGA codes for ROC
 - standalone Geant4 codes (Chen-Wei's? Yumika's? anything else?)

GitHub management

- Projects, codes to be managed (as far as I remember):
 - General macros for data from the testbench
 - Ladder classification
 - Nevis GUI
 - FPGA codes for FEM and FEM-IB
 - FPGA codes for ROC
 - standalone Geant4 codes (Chen-Wei's? Yumika's? anything else?)
- Ideas to manage them:
 - A. Putting everything in the repository “INTT”
 - 😊 Simple for users.
 - 😓 Everything comes together even (some projects have large files) if you don't need some.
 - B. Making a repository for each projects
 - 😊 You can choose what you get.
 - 😓 Requests need to make repositories.
 - C. Intermediate way of A and B

GitHub management

- Projects, codes to be managed (as far as I remember):

- General macros for data from the testbench

repo: INTT

- Ladder classification

- Nevis GUI

INTT_Nevis_GUI

- FPGA codes for FEM and FEM-IB

INTT_FPGA

- FPGA codes for ROC

- standalone Geant4 codes (Chen-Wei's? Yumika's? anything else?)

INTT_simulation

- Ideas to manage them:

A. Putting everything in the repository “INTT”

- 😊 Simple for users.

- 😓 Everything comes together even (some projects have large files) if you don't need some.

B. Making a repository for each projects

- 😊 You can choose what you get.

- 😓 Requests need to make repositories.

C. Intermediate way of A and B