

# Skander Moalla

*Project co-leader / Git Leader / GameEngine team leader*

---

- As a team co-leader:
  - Divided the team into subteams: We made 3 subteams: The *GUI team*, the *GameEngine team* and the *ObjectDesign Team*. Given each one's preferences and skills we assigned each team member to a subteam.
  - Did routine check ups to make sure everybody has something to work on.
- As the **GameEngine** subteam leader:
  - Code and Design:
    - Designed the **GameEngine** class: I designed and coded the signature of the methods and attributes of the **GameEngine**.
    - Designed the interaction between the **public** and **private** methods of the engine.
    - Designed and implemented the interaction of the GameEngine with the different Objects and GUI elements **User**, **Team**, **Player**, **League**, and the **NextGame** window.
    - Coded **GameEngine::applytactic(Team& playingTeam, Team& initTeam)**.
    - Coded **GameEngine::getBacktoDefaultTactic(Team& playingTeam, Team& initTeam)**.
  - Management:
    - Designed the independent methods of the GameEngine to be able to share their coding among the team.
    - Supervised the coding of the shared methods.
    - Managed the communication with the GUI and the ObjectDesign team to make everything interact well together.
- As the Git Leader:
  - Created the GitHub repo.
  - Helped all team members familiarize with git mechanics and theory:
    - Git theory: local repo, remote repo, and push-pull requests.
    - Familiarize with GitHub desktop.
    - How to commit and solve conflicts easily on visual studio code and GitHub Desktop.
    - gitignore: add their build and user preferences in the gitignore.
  - Maintained a healthy sustainable and a runnable HEAD so everyone can test their new features.
    - Implemented different branches for the subteams to try new features.
    - Reverted commits when needed, resetted the HEAD sometimes and debugged everytime.
- Other contributions:
  - Fixed bugs in **Team** and **League** objects.
  - Designed the mechanics of a Game: implemented all **GameEngine** methods to ensure a smooth transition during the game.
  - Tried to do the make file of the project with CMake, but we eventually built the project with Qt thanks to the GUI team.