**Group 20 Chess game Local two-player**

**Tools and Technologies**

**-** Python 3.10

- Tkinter – Graphical User Interface

- Pytest – Unit testing

- Pipenv – Dependency management

- GitHub + GitHub Actions – Version control and continuous integration

**Team Roles**

**Group lead:**

* Neala Daniel Mingyi

**Backend Team: Piece logic & rules**

* Neala Daniel Mingyi
* Faruk Sani Aima
* Joshua Chukwuma Chime
* Abdulahad Kasim Sulaiman

**UI Team: Chessboard interface**

* Chidubem Godsfavour Ogbonna
* Caleb Olugbemi
* Pytest tests
* Ezeani Chidiogo Scholastica

**Documentation: README, report, demo video**

* Umeakunne Silvia Adaugo
* Joshua Chukwuma Chime

**Abstract**

This project developed a local two-player chess game in Python, combining object-oriented design, a Tkinter GUI, and a console mode. Automated testing with pytest and GitHub Actions ensured stability. The game runs successfully, and the team gained valuable experience in software development and collaboration.

**Introduction**

Chess was chosen as it challenges programming skills in algorithms and UI design. The objectives were to build a working game, provide GUI and console modes, integrate testing, and practice teamwork using GitHub.

**System Design and Implementation**

The backend (game.py) manages board state, move validation, and turn switching. The GUI (board.py) uses Tkinter for piece interaction, while main.py supports console play. Testing was automated with pytest, and CI was set up via GitHub Actions.

**Challenges Faced**

Difficulties included import errors, Git/GitHub conflicts, limited time for advanced rules, and gaps in experience with pytest and CI workflows.

**Results and Evaluation**

The game enforces legal moves, supports two-player local play, and runs in GUI and console modes. Automated testing enhanced reliability. The design proved stable and modular.

**Future Work**

Planned improvements include timers, AI opponent, online multiplayer, and broader test coverage.

**Conclusion**

The project achieved a functional chess game and provided strong lessons in programming, debugging, collaboration, and automation. These skills will support future software development endeavors.