Implementation of the Japanese board game Go

Project Roles

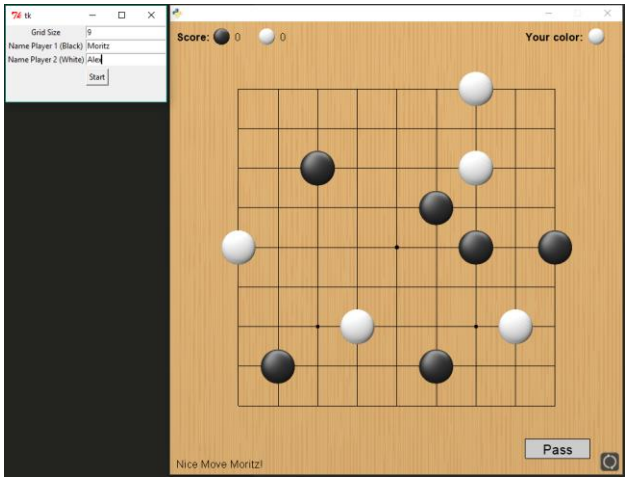

Figure 1: Go implementation

Alex Scheitlin

Responsible for: **`go.py`** & **`client.py`**

Contributions

Implementation of the first half of the client and the initial skeleton. This includes the loading and drawing of the window, grid, stones, and circles. Mouse clicks are also registered and trigger the placement of a stone on the grid. Furthermore, a simple GUI to start a game with a customized board size and player names is implemented with *Tkinter*.

Moritz Eck

Responsible for: **`game_model.py`**

Contributions

Implementation of the game logic, including the initialization of game data, the passing of a move, the validation and processing of the game's moves. Each player's action is passed to the model via the controller. Inside the model, the completed game logic to place stones, capture groups and mark territories and computing the score is implemented. The model directly performs the calculation and data operations.

Nik Zaugg

Responsible for: **`controller.py`** & **`client.py`**

Contributions

Implementation of the second half of the client. This includes the loading and drawing of labels on the field as well as displaying required buttons. Furthermore, the communication of the client with the model is enabled by the implementation of a controller. Controller functions include the passing of data between client and model and handle the initialization of a new game.

General Remarks

We were able to implement all required features of the game. It is playable with 2 players and follows the basic rules of Go. Further improvements that could be added would be a more sophisticated graphical user interface as well as a multiplayer-mode on separates screens (Web Application).