Project 4 Team Contract
Adriano Hernandez, Jay Lang, Chris Rinard, and Anthony Grebe

Meeting and Communication Norms

We will meet as necessary, but ideally at least three times a week for at least one hour. The group meetings will be primarily for discussion of large-scale strategy, and most programming will probably be done either in pairs or individually. We will use Signal as the primary mode for remote communications. On deadline days we expect high responsivity.

If someone is even considering dropping the class, they will inform the rest of the team ASAP. They will also finish completing the work they already promised to complete, thoroughly document their code, and write up a summary of their work to make the transition easier.

Work Norms

We anticipate putting in at least 20-35 hours of work per week to make this project successful. We will allocate time and tasks as we deem necessary after deciding after checkpoint meetings.

We might set personal/team deadlines. Each member is allowed to work in their own style and pace as long as their work is completed by the deadlines set by the team.

If a member of the team does not follow through with commitments, the other members reserve the right to inform the teaching staff of their lack of adherence to this contract.

Code that does not pass correctness tests CANNOT be pushed to master. Please lint before pushing. We will work on feature branches with descriptive names (akin to <user>/<feature>). Ideally, rebase push and pull but merge is OK if necessary during crunch time. Please squash all commits that belong to a single meaningful feature change if it is possible and necessary. It is necessary if you have, say, ten trash commits that are not semantically relevant. Each commit onto master should be semantically relevant to a feature. This keeps history easy to understand and makes debugging easier. Origin can have copies of feature branches. Every code block (for example, a function) should have at least a one-line comment explaining its purpose. Code ought to be modular. Git PRs will be used.

Decision making

Decisions around bigger architectural components must be consulted with the other members and opinions should be heard out, time permitting. Implementation decisions, however, of how to implement agreed upon architecture is discretionary. We can decide on whether a component requires unanimous consent to be pushed or not before embarking on its creation.

Assuming the code is correct, we should pick the code that performs better against reference implementations that we are competing against. If we are able to beat all the references, then we will optimize for highest ELO against other bots in the scrimmages. This will likely mean that each iteration will beat the previous one (especially in initial stages of testing), but if there is not a strict ordering of play styles (i.e. if the meta is rock-paper-scissors), then the scrimmages are a better comparison.

x Adriano Hernandez
x Jay Lang
x Chris Rinard
x Anthony Grebe