Project 3 Team Contract
Adriano Hernandez and Nancy Vargas

Meeting and Communication Norms
We will meet as necessary, but ideally at least twice a week for at least one to two hours. We hope most coding will be pair programming, but we understand that differences in our schedules may push us to collaborate asynchronously at times. We will use messenger as the primary mode for remote communications. On deadline days we expect high responsibility.

If someone is even considering dropping the class, they will inform their partner 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 roughly 10-15 hours of work in 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 member reserves 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. Please squash all commits that belong to a single meaningful feature change. 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.

Decision making
Decisions around bigger architectural components must be consulted with the other member and must be a unanimous decision. 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 faster code when deciding. In certain cases we might care about portability or other factors.

x Adriano Hernandez
x Nancy Vargas