Team-wide

Strong Points

Group dynamic. Having a good group dynamic made it much easier for us to support each other and not get caught up in arguments.

Effective division of labor. The roles we chose for the MVP implementation drew on each member's strengths and carried us well through the rest of the project.

Collaborative design. We had long discussions about design issues, evaluating the merits of different approaches. This led us to a successful design.

Weaknesses

Scheduling. We had a hard time coordinating for a variety of reasons. We keep different waking hours, have different weekly commitments, and have different styles of managing deadlines. More gravely, this project came along with the end-of-semester crunchfor all of us; our loads were heavy and highly unsynchronized. Due to tests, performances and projects in other classes, it was rare for more than two of us to be free at a time, making integration difficult. This resulted in a last-minute crunch and forced us to give up the replay view feature. To some extent scheduling conflicts are inevitable when outside commitments are heavy. The main way we could have lessened these problems is to plan out *exact* meeting times well in advance, with the time-scale of the whole project in mind, and *commit to them in advance.* This would avoid the multiple-hour overheads often incurred in same-day meetings.

Planning and specifications. We tended to push back issues of implementation. Sometimes this was okay, like with the game-state server-to-client protocol, but sometimes it led to delays and frustration. We also did an okay at best job of modular testing, which gave us lots of last-minute issues to work through.

Bethany’s Reflection:

Damien: Took charge of the sockets very well, good communication with the creator of the sockets Ruby gem plus a willingness to reach out in order to get something done. There were a few times you were unavailable when the rest of the team was going to meet, but you made up for it by working by yourself on the tasks at hand which worked out just fine.

Leon: It was a bit inconvenient that your computer didn’t work for the project, and that Athena didn’t work well for it either. I know you tried dual booting, vagrant, and a few other techniques for it that didn’t work out, so there wasn’t really much that you could do on your own. However when working on my computer/someone elses you were very productive and worked pretty well at doing so. You also attended meetings regularly and worked well at them, since you couldn’t do anything in your own time.

Ben: You commented your code as you went along which made it really easy in the end, a good programming tactic (as most people choose to comment their code after the fact, making it a bit hard for team collaboration in a SCM environment). You arrived at every meeting that we planned when you were available and was really good at team communication, allocating what needed to be done, and taking charge of the organization for the most part.

Ben’s Reflection

Comments to Team Members

Bethany

You did a great job getting along with the team -- I appreciate how difficult it is to work with a group where the other members already know each other. It’s also worth noting that you were often working later than your normal hours to accommodate the rest of us, which takes devotion and is not easy. On the technical side, it was nice to be able to count on you to create views for my models; I could focus better by taking my mind off of model issues completely until the integration phase. You did a great job on the views (and also on the wireframes). On the other hand, the integration phase was valuable practice for me in reading others’ code; I hope you got some more backend experience during this integration phase too. Another area where I think you could improve is in how you reuse code. Packages and modules can obviously be treated as black boxes, but for sections of code transplanted from previous projects, you should make sure you understand how they work (and perhaps adapt and clean them up a bit) before using them in the current project.

Leon

I was worried when you ran into problems with your computer; I thought that group management and pair-programming had only a small benefit and we would effectively be one member short. I was quite wrong. Your management is what kept us all on-board and focused, especially during the MVP phase. You helped keep our attitudes positive and realistic. Your pair-programming was also valuable, as there was always at least one group member who needed a fresh pair of eyes on their code. As a technical next step, I would recommend getting comfortable with Linux, as that will provide you much more power and transparency in almost any development setting.

Damien

You were quite a workhorse this project. We missed you on Saturday before the MVP deadline, but we would have been sunk without your all-nighter that night. You shielded us from the trouble of integrating WebSocketRails and Trueskill, and also were a great help to both the frontend and backend teams. The amount and quality of code that you wrote was tremendous. I think the main area you could improve in is staying focused and realistic -- sometimes “dream features” would distract you a bit from what absolutely needed to get done. Also, I know the end-of-semester chaos was largely out of your control, but it would have been nice to have more face-time with you, especially since integrating the components together was such a hurdle for us. If you could figure out how to make the daytime hours of your schedule more flexible, it could help a lot in future projects.

Leon’s reflections

Ben, you were good at staying on top of things and keeping us organized. I appreciated that you were willing to get your hands dirty and do the annoying stuff (various design items, meeting minutes, getting me to install Ubuntu). Your productivity had a tendency to fluctuate a lot, though, and when you weren’t in a good mood it kind of dragged on the whole group.

Bethany, our pair programming setup seemed to work pretty well. You provided good feedback and called me out when I wanted to do something dumb. You could have been a little more outspoken at times, though; it was sometimes hard to coordinate meeting up and to get your opinion on design decisions.

Damien, you definitely did work when push came to shove, and were generally positive and productive during meetings. There were definitely some issues with time management, though; some of your best work was done at around 5am on Sundays, which made it difficult for the rest of the team to keep track of your progress.

Damien's Reflections

Ben: You did a really good job organizing and driving the group, especially when motivation lagged. You also kept us on track with lots of the minor deliverables. You contributed the bulk of the backend design, and implemented that design with clean and well-commented code (though some of your console outputs weren't very helpful.) I appreciate how much work you put into this project despite having so many other high-work classes to juggle simultaneously.

Bethany: Despite the fact that you didn't know the rest of us beforehand, you integrated into the team very well. Thanks for staying up so late all the time for our (often) last-minute meetings. I appreciate how you helped keep us on track when we got distracted. Your front-end design and code were sensible and worked well, though I had some minor quibbles with how the final views turned out.

Leon: Though it was unfortunate that Windows didn't work with Websockets, I think your pair-programming worked out pretty well in the end. You contributed a lot to both the view and controller code, and it seemed like you worked very well with Bethany and Ben (and helpfully hovered over my shoulder a bit as well.) Thanks especially for the idea behind the project and for your input, based on past experience, on a lot of the design decisions.