10/23 - Sprint 3 Reflection

Overall, the division of the sprint requirements was well managed. The addition of the burndown chart in Trello worked wonderfully. One thing to improve on moving forward is having a more concrete expectation of time to spend on each task; the “points yet to complete” part of the burndown chart increased significantly over the course of the sprint while it should have only gone down. Bug detection systems worked well; the combination of pull request reviews and manual testing as each member tries to use previously implemented items quickly identified logical errors and allowed us to nip several problems in the bud. Documentation improved as a whole, building on last sprint’s setup.

As expected, the primary sticking point of the sprint was merge conflicts. A lack of allocated time for resolving conflicts caused us to effectively miss the functionality check-in and delayed progress on the final submission. In the future, the team should have better communication about major refactors to ensure as limited a scope as possible.

A screenshot of a graph

Description automatically generated

Looking at code analysis, the cyclomatic complexity of the program doubled from the previous sprint (largely due to collision checking and object spawning switches) and class coupling increased by 50%. However, each of these metrics decreased for most individual classes, indicating that overall file/class structure has improved even as the project has gotten larger. A few notable weaknesses were Game1’s Initialize, as it still contains Command registration; the square collision checker, as it is hyperspecific to certain edge cases; and several HandleCollisions methods, as they rely heavily on switching. These should each be addressed and optimized in future sprints.

Looking ahead: HUD implementation will likely require a refactor of our current object position logic, care should be taken to allocate enough time for that.