

CS307 Sprint 2 Retrospective

Team 2 - Anant Goel, Don Phan, Jason Rahman, Jon Egeland, Josh Selbo, Levi Linville

Group Reflection

As a team, overall we are satisfied with the progress made during sprint 2. Going into the sprint we aimed high and set the goal of accomplishing the core of the game's functionality, including gameplay, level designs, authentication, sandboxing executed code, and finally abstracting the game logic into a C++ library, in addition to the remaining sprint 1 tasks. The fact that we completed these goals and successfully linked the layers of communication together speaks volumes to the progress made this sprint. We did not accomplish 100% of the tasks, but we anticipated this by leaving sprint 3 as a time to clean up the remaining tasks, refine the design, and wrap up the project.

Our standup meetings continued to be essential to our team's operation, and the integration meetings introduced during this sprint proved to be very productive. These meetings were useful for ironing out details "at the margins" of our individual components. In practice these integration meetings exposed some instances where there was a mismatch of expectations. However, increased proactive communication between developers working on separate components would be useful to find differences in expectations sooner. More internal documentation regarding a) level formats, b) message formats and c) functional behavior of components would be helpful to reduce the time spent combining different components. This documentation does not need to be elaborate; simple examples are sufficient to avoid wasting time over simple mistakes such as misnamed fields, etc. Integration meetings will be less frequent during sprint 3, and when they are necessary, will likely be limited to only an hour long.

While improving user experience was a task that we concentrated on towards the latter half of sprint 2, it will be our primary focus for sprint 3.

As a team we continue to be productive and encouraging to each other which helps keep everyone motivated.

Story Reflection

Successful Examples

As a developer, I want game assets to be stored in a database for consistent access

The resource server was seen as a means to decouple resource access from resource storage. The benefit we receive is significant future flexibility in how resource storage is architected and formatted. Initial prototyping was rapid - only a few hours - and subsequent overhauls have dramatically improved the functionality of the server.

Currently, the only resources being served are the images used in the game. However, the server is capable of serving any file asset through a static URL, providing a persistent naming scheme for applications which do not have the necessary storage space.

As a developer, I want to have the core game engine implemented in C++

The motivation for implementing the game engine as a C++ library was to make it easy to support more programming languages in the future beyond Python. Modern C++ idioms such as smart pointers were used, easing development effort and leading to a better product that could have been achieved otherwise. The game library can be called by any language with C++ bindings. Overall, we were able to successfully build an object-oriented game engine with an actor-observer pattern which provides inversion of control between core game infrastructure and the game actors. The library tracks interactions between the game character and the world. These interactions are serialized as a JSON event log which is passed back to the execution server, eventually being parsed by the client and translated into animation events. The library has a unit test suite, which dramatically increases the ease of development by providing immediate feedback regarding new code.

As a developer, I want to design a story to go along with the curriculum.

Trying to think like a middle schooler was slightly challenging. Thankfully, one of our group members has been able to interact with middle schoolers during his time as a mentor in Reaching Out for Computer Science program here at Purdue. Overall, we thought the target audience allowed us to be flexible in creating the story since most kids at that age group are easily entertained. Therefore, this allowed us to take the story wherever we saw fit with our programming curriculum.

All Unsuccessful

As a student, I want game logic to run inside a contained environment

We were unsuccessful in fully completing this story because of an unexpected externality. We had completed ~90% of the functionality, but discovered that the third party library we were using had a bug that was introduced in a recent version. We were able to rapidly deploy an improvised solution for the short term, but we still need to complete this feature in the long term. The prime lesson learned here is two-fold. First, it was a good case study to drive home the point that it is dangerous to rely upon immature APIs and technology. The second lesson was to always leave yourself an out by designing the system in such a way that you have flexibility to choose an alternative implementation option.

As a developer, I want students' code to communicate with the game library

We were mostly successful finishing this task, but ultimately, we ran out of time at the very end to complete every required component. We achieved full functionality required, but failed to satisfy the constraint that we needed to use Restricted Python to prevent the student from writing code with undesirable results. Again, this was an issue with third party components that interacted in an unexpected and negative manner.

Individual Reflection

Anant Goel

I personally feel that this sprint went very well. A lot of time was spent breaking down the user story into small achievable tasks. This approach worked out very well as it allowed me to think of a user story as a number of tasks and then approach each task individually making sure it's completed product fits well with everything else. The sprint allowed me to learn about implementing logic in Ruby on Rails and improving the user experience by modifying views of certain pages. This Sprint definitely went off smoother than Sprint 1 for me as I was more comfortable in the position I am in for the team along with the fact that our regular meetings allowed us to collaborate with each other and work on the common product with people's specific strengths in mind. The time I allocated for the tasks was what I needed and hopefully the same will continue for the future sprint.

For this sprint the few things that did not go well initially were related to the hint button and tutorials page. The hint button was posing some issues as it required some manual tinkering to be done on a specific page. This task was confusing at first but after discussing with some members of the team I was able to implement it. The tutorials

page was a tough task to complete as it required a lot of writing to be done. I was not too excited when I started implementing the tutorials page but after a while I gained interest in it and was able to implement a decent tutorials page for now.

Don Phan

I feel like this sprint went very well for me. Things that were a struggle for me were not too big to overcome. The main issue was trying to develop questions that were geared towards middle school students and not college level interview questions. I was able to finish questions from the arithmetic, conditional, and logical programming curriculum. But, when I got to arrays and strings, the questions I was coming up with felt like they were interview questions. Obviously, it was too difficult for those students to handle. So, I decided to save those problems for next sprint and will work with Phil Sands to come up with age appropriate questions.

Levi Linville

This sprint saw many improvements to the existing execution server. The main task I needed to accomplish was to act as an intermediary between the routing server and the game library. It was relatively simple to connect to these two, but I encountered problems with the message formats being passed through the execution server. Jason was a great help on the C++ game library's python wrapper and its use. I am now able to take code submissions from the routing server and run them with the game library's functionality. These were partially leftover tasks from sprint 1, but I was able to accomplish them and the new sprint 2 tasks. I had to rush the development a little towards the end of the sprint, so I have some technical debt to solve in cleaning up my code. This shouldn't be terribly difficult, taking only a short period of time from the upcoming sprint. Overall, this sprint was both productive and a time of learning for me.

Jason Rahman

I spent this Sprint playing catch-up, simultaneously developing the majority of the game library, and integrating the game library, the execution server, and the routing server together into a functioning system. I faced three major setbacks during the Sprint, one related to a third party API, and a two related to internal components. In both cases I feel like I was reasonably successful at being able adapt to both challenges. Unfortunately my development took significantly longer than expected, and in several cases I spent more hours per week than I had initially allocated. In the future, more proactive communication with team members will help to avoid discrepancies in interface specifications, and provide transparency regarding the state of individual components.

Joshua Selbo

While in sprint 1 I focused on finding and integrating libraries and assets and creating a visible game canvas, in sprint 2 the majority of work involved implementing a game state for the client and responding to event logs generated by the game library. I was able to implement responses to all events we planned for with the exception of level change triggers. I worked closely with Don while he created level tilemaps and with Jason while he worked on the game library in order to ensure congruence with level formatting and event parsing across all layers, although admittedly we can do a better job next sprint in communicating formatting expectations and contracts. This can be accomplished by sharing documents with these pieces of information.

Jon Egeland

For me, Sprint 2 seemed to go more smoothly than Sprint 1. I felt more comfortable working as a team and communicating both issues and successes throughout the Sprint. In terms of work, I focused primarily on polishing the application server, whose initial development was the primary focus of Sprint 1. This involved tracing out a few bugs that came up near the end of the previous Sprint, as well as some new developments - including managing the handoff of students' code to the routing server and back again. There were a few obstacles in terms of ensuring that the libraries we were using were fully compatible with our system, and especially getting the two servers to communicate clearly and consistently, but I felt that I was being productive and contributing meaningfully to the project. For the next Sprint, I hope to be able to focus more on gameplay development and improving the smoothness of the user's experience while using the application.