

Sprint 3 Retrospective

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

Group Reflection

As a team we are pleased with the end product we created over the course of the semester. HERO.play was a learning experience in more ways than one for many of us, and it was a great opportunity for us to improve our communication and interaction skills while working as part of a team. We successfully put into practice many guidelines learned in CS 307, including source code management, feature branching, scrum agile software development, user stories, and various interaction diagrams. All of these enabled us to work effectively and efficiently as a team. Our final presentation was professional because we planned a specific presentation schedule and rehearsed several times to ensure consistency.

Together we worked more efficiently than in previous Sprints, likely due to greater familiarity with the frameworks in use and our experience working as a team. Everyone had a clear understanding of their responsibilities and team members actively followed up with other group members if work was not being completed on time. In future work, we can improve in a few different ways. We can improve by sharing internal API specification docs between client/server relationships. Additionally, we can prevent situations where team members have different visions for a feature by being proactive about communication and ensuring everyone has the same understanding.

We will discuss with our project coordinator, Phil Sands, what he envisions the future of this project to be. At the beginning he stated that if the project is successful he could arrange for it to be tested by middle schoolers and possibly be adapted into a curriculum for teaching programming skills.

Story Reflection

Example Successful User Stories

As a developer, I want unit tests for the game library to ensure it functions properly.

Unit tests were written during the development of the game library. The tests were written to match the specification of the game play mechanics, and evolved in time as those mechanics evolved.

Development policy was to run the unit tests on the game library prior to committing major changes. This ensured that we nearly always had a copy of the code for the game library which was entirely or almost entirely functional. Also, any newly discovered bugs were added as an additional test case for both the purpose of validating the initial fix for the bug, and to ensure that the given bug did not regress in the future as additional changes were made.

As a student, I want to read tutorials to help me with navigating and using the site.

This task was carried on from Sprint 2. More work was done on making the tutorials page user friendly and appealing to new users. An easy to use scanning bar was added to the page to allow users to quickly access specific instructions they might be looking for. The tutorials page improved a lot from Sprint 2 as there were a number of additions made to it which helped in making the page effective overall.

As developer, I want to refactor the tile maps that were created in Sprint 2.

Unit tests were created to help with the refactoring of levels 1, 2, and 3. The unit tests helped check for consistency to see if the same RPG-Map pack were used. Also, the tests helped check for the triggers to be in the proper format. This helped greatly because it caught a lot of missing or improperly formatted triggers. Finally, the last step of refactoring the tile maps was to rearrange the tile board to force the student to take a certain path to maintain the appropriate learning process.

All (Non-optional) Unsuccessful User Stories

As a developer, I want to have maintainable code that conforms to JavaScript best practices in the game implementation.

I assigned myself (Joshua) this task at the end of Sprint 2 after becoming frustrated with having to “hack” the client code together to get it to work. However, while working with the code in Sprint 3, I became more comfortable the layout and naming conventions, and another group member confirmed it was understandable. Due to these reassurances, I never made this task a priority to work on.

As a developer, I want unit tests for the app server to make development smoother.

While we had initially created a number of unit tests for the app server in Sprint 2, we found that changes to the system in this Sprint caused a number of them to break. Not only that, but some major changes caused some unavoidable errors that also caused problems in the server itself. To get around this, we accepted that the affected functionality would simply be ignored, as both fixing it and fixing the associated tests

proved to be a far greater task than we had originally planned, and there were a number of other tasks with a higher priority still to be taken care of.

As a developer, I want behavioral tests to simulate potential use cases.

Behavioral tests were always going to be a stretch. The idea was to simulate a user going about their normal workflow on the site. However, this workflow changed constantly throughout the Sprint as both design and development decisions were made that affected what different users were capable of doing, as well as the methods by which these actions were performed. As a result, and due to other tasks taking priority, we were unable to create any behavioral tests that could effectively test the system.

As a system administrator, I want sessions to run inside an isolated container

For this particular task, we had many challenges with getting a third party Scala library for Docker to function correctly. Initially this task carried over from Sprint 2, and we had intended to finish it in Sprint 3.

The vast majority of the work for this task had been completed: We successfully built a Docker image containing the execution server and game library, we had incorporated the API calls to create/destroy these Docker containers into the routing server, and the routing server had appropriate functionality for communicating with the execution server inside the Docker container.

The sole problem was related to an undiagnosable bug when we tried to actually run the routing server and request the creation of a Docker container. Countless hours were spent trying to debug this exception, in both Sprint 2 and Sprint 3, but to no avail. Eventually, other tasks became more pressing, and so this task slipped.

As a student, I want timeouts on my submission so that infinite loops aren't infinite.

While this seemed to be a straightforward issue of “giving up” on submissions which take longer than a set amount of time, it proved to be more difficult. All available solutions involved creating a separate thread in which to do the computation. Running the ‘exec’ command on the students’ code can be easily done in a separate thread. The problem comes in communicating the results back to the main thread that keeps time. I was unable to find a solution to this problem, as some of the data which needed to be transferred was not serializable, and thus couldn’t be sent between threads.

Individual Reflection

Anant Goel

I personally felt that Sprint 3 went smoothly for the most part and it was a great way to finish most of the tasks we as a team set out for ourselves. The beginning of Sprint 3 was centered around mainly cleaning up tasks that were finished in a non-elegant manner in Sprint 2. For me specifically, this mean cleaning up the tutorials page and adding sanity checks for certain items, such as a student withdrawing from a course. Those tasks went well as I had a clear idea of what to do and how to implement them.

Nothing in particular did not go well. I concentrated a lot on improving site usability and modifying any apparent faults with the website. A particular fault that was not solvable was fixing a SQL error that occurred when clicking on the 'Courses' page when a user is not signed in. In the end everything went smoothly and I feel that reflected in our final presentation.

Don Phan

For Sprint three, I believe that it was pretty good since it was mainly refactoring problems and fixing loose ends from game level 2. We decided that since we won't get to demo all six levels we decided that it was better to take a sample problem from the five later levels and put them into level 2 to show that this was our plan for those levels. Towards the end of the Sprint, I think I did a good job planning for the demo presentation because I was able to envision how the demo would go and squeeze all our progress within a 10 minute span.

Jason Rahman

For Sprint 3, my initial progress was slow because of the Thanksgiving break and other priorities. I experienced significant difficulties trying to implement support for Docker containers. I faced repeated failures while debugging the third party library we were using, and sought another third party library as a replacement. Eventually, other stories and tasks became a greater priority and that task fell to the wayside.

Work on the in-game objectives was slow at first because of a some initial lack of communication and misunderstanding regarding the exact vision for how that feature was to be implemented. Following some group discussion of that feature and the establishment of a consensus for how in-game objectives were intended to work, progress proceeded very quickly.

By the end of Sprint 3 I feel like I had accomplished the core functionality that was required. I do regret that I was unable to complete the Docker container functionality, but external forces conspired to make that task far more difficult than expected.

Joshua Selbo

Like last Sprint, I made average progress throughout the first few weeks and made the most progress the week of the Sprint ending. This worked out very well because it allowed the developers working on the servers to fix and refine their functionality throughout most of the Sprint. In the last week I made several changes to integrate different responses returned by different endpoints, and to ensure a consistent visual game state.

I was not able to implement several of the “optional” or nice-to-have features (e.g. allowing the user to select the hero’s gender) because of time constraints, but I am not disappointed because as a team we anticipated not finishing several of the optional tasks. I think it was appropriate to list many optional features knowing that realistically only a few would be implemented, because having more tasks motivated us and pushed us as a team.

Jon Egeland

Originally, this Sprint seemed to be fairly relaxed. Coming in to it, I felt that I had a solid grasp of what my tasks were, and was comfortable with the scheduling that I had come up with. Halfway through the Sprint, however, it became clear that there were a number of small issues and changes popping up that caused me to gradually fall behind. In spite of this, though, I was able to work through them and finish a majority of my tasks, as well as help out other members when they asked for help. Overall, I feel like my work during this Sprint and the project as a whole were both fairly productive, resulting in a fairly stable end product that I was proud of.

Levi Linville

For me, Sprint three consisted of a couple new features and some cleanup of existing code. I added support for objectives, along with attempting to add timeouts to code submissions. My attempts at adding timeouts failed, unfortunately, as it broke communication with the game engine. While this seems to be a straightforward issue of “giving up” on submissions which take longer than a set amount of time, it proved to be more difficult. With our current setup, this couldn’t be rectified by the end of the Sprint.