

1 Planning and Reflection

1.1 Complete Schedule

September 29

On Time

- Registration form that writes to the database
- Log in form that queries the database and writes session cookies
- Log out button that deletes session cookies
- Basic informational content (about us, contact us, privacy policy)

Delayed/Ongoing

- Research on server connection
- Basic informational content (code of conduct, terms of use)

Added

- Site mock-ups created with login spot
- Using SSL encryption

October 6

On Time

- Account settings page that edits a database entry

Delayed/Ongoing

- Player history page that reads from the database*
- Server connection done*

Added

- Javascript validation of forms
- Basic CSS stylesheet
- Dropdown menus in place
- Basic informational content (code of conduct, terms of use)

October 13

On Time

- Layout prototype finalized

Delayed/Ongoing

- Top scores page that reads from the database*

Added

- Layout fully coded in HTML/CSS/PHP
- Further Javascript validation of forms
- Account settings page with current user information preloaded
- Integration of pong code and communication code

October 20

Note: The week of October 20th was included in our initial plan as a buffer week with few goals, but due to fall break and midterms our boss waived progress for this week, thus we have little output.

Delayed/Ongoing

- Live streaming completed*

October 27

- Integration*

November 3

- Game registration
- Live streaming*
- Player history page that reads from database*
- Top scores page that reads from database*

November 10

- Private messaging forum
- News and special events
- Initial testing for cross-browser compatibility

November 17

- Online gameplay
- System requirements page* (including site-supported browsers)
- Rules page*

November 24

- Pong game looping in background of content box
- More detailed records and high scores*
- Further testing for cross-browser compatibility

December 1

- Testing and polishing of all existing features

Note: If time permits, we may consider implementing additional secondary features. However, at this point, we do not want to commit ourselves to any particular task, as we are not sure of what unexpected setbacks we may have encountered prior to this week.

* May be delayed due to reliance on information/collaboration with other departments

1.2 Challenges

1.2.1 Server/Client Communication

One of the main challenges that we faced in creating a method of viewing VirPong games was in the actual communication between the server and the client. In the past, there have only been a handful of ways that developers have tackled the issues of browser communication including the use of making a connection via a port opening by Flash, or through Reverse AJAX. However, both of these techniques had disadvantages that we wanted to avoid, so we elected to use a newer technology, WebSockets, specifically the Socket.io framework for Javascript. The challenge that came along with this route is the newness of Socket.io as well as Node.js, which we use in conjunction of Socket.io. Looking for online tutorials, we found many tutorials that used both of these technologies in different situations, as well as samples of code that used the two technologies in conjunction with other frameworks, making it difficult to get an understanding of how Node.js and Socket.io worked independently of these other frameworks. To overcome the issue of adopting these two new technologies, our team focused on trying to re-implement tutorial code as we found it, and upon finding working tutorial code, we would manipulate the code in attempts to break the code down into more simplistic forms. Going through this process, our team was able to achieve very simple recreations of tutorial code that we had found, providing us with an understanding of what is required in using Node.js and Socket.io.

While this technique worked for our team, and we were able to break tutorial code down into simplified forms, and from these simplified forms determine the best way to structure our own code, we believe that we may have saved some time by investigating our options further. During this process, we explored many different methods of communication between the client and the server, however, we were eager to begin coding and would often try to implement a piece of code before pursuing alternatives. We believe that if we had pursued other examples or pieces of tutorial code, we could have shortened the amount of time, and potentially frustration, that we used to develop a working implementation of server to client communication.

1.2.2 Large Workload

One challenge that we anticipated in the Initial Plan was the sheer number of tasks that lay before us in creating a functional, attractive web user interface for VirPong. Identifying this issue early on was a key to the way we have handled the workload. By always keeping in mind the magnitude of our task, we stay motivated to produce results each and every week.

In hindsight, this challenge may have been easier to deal with if we had planned our initial schedule more carefully. One of our proposed strategies for tackling the workload was to stick tightly to our schedule, but this proved unrealistic. Often our features relied on the existence of features from other departments – for example, we cannot code and test pages that read information from the database of past games until we have discussed with the server department who is responsible for writing this information to the database and what the schema of the database will be. Had we been more aware of the realities of this situation, we would have placed all such features that rely on other departments further down the schedule, and we would have communicated our needs to other departments before the submission of the initial plan so that they might consider them when designing their own schedules.

Because our schedule was at times unhelpful, rather than following it to the letter we have taken care to be proactive about getting things done. In particular, Kyle and Katie often found that the tasks assigned to them were unfeasible at the present time. Rather than shrugging it

off and making no progress for the entire week, their response was to seek out alternate tasks that could be completed. In this way, we implemented Javascript validation of forms and a complete site layout more quickly than anticipated in the original schedule.

Our other main strategy in conquering our heavy workload was splitting our team into pairs that worked together on shared tasks. In general, Aryn and Garrett have been configuring the database and working toward communication with the server while Katie and Kyle have been writing pages that access the database and focusing on the aesthetics and organization of the website. We have been happy with the buddy system, as it simultaneously holds us accountable for working on our assigned tasks and gives us another bright mind to bounce ideas off of. Working in pairs has also been quite practical, as it allows our team to tackle two problems at once. Since many of our meetings are with only the pair as opposed to the whole group, they are easier to schedule into our busy lives.

1.3 Milestones

November 3: Live streaming

Aryn & Garrett: Communication with the server, interpreting game information
Katie & Kyle: Aesthetics, integration with site layout

November 10: Testing for cross-browser compatibility

Aryn & Garrett: Testing the game display
Katie & Kyle: Testing the main layout & navigation

November 17: Online gameplay

Aryn & Garrett: Retrieving & interpreting game information
Katie & Kyle: Aesthetics, integration with site layout

December 7: Final release

Aryn & Garrett: Ensure that databases, game display, and chat are functioning properly
Katie & Kyle: Ensure that all graphics, hyperlinks, forms, database writing, and database reading are functioning properly