

Story Creator Final Report

Brandon Roberts, Nate Olderman, Billy Rathje, DJ
Maguddayao, Kyle Dybdal

December 13, 2013

1 Project Summary

The Story Creator module combines the Sharing Framework, Object Creator, Visual Editor and Animation modules in order to create a holistic view of Edith in a dynamic webpage.

The Story Creator will be directed to a login page that will prompt them for their username and password. If they do not have an account they may create one in the link that says "Create Account." In the case that they have already have an account they will be directed to the main Edith page where they will be able to create a "story." To do this they will click on objects(this makes them glow and puts them into an array for Visual Editor), give the objects animations by manipulating the selected object in the Visual Editor canvas, and create ways to interact with their story by manipulating other objects. By doing this, students will be able to learn how relationships among objects work in programming.

The Story Creator Module designed the Edith main page, login page, and sign up page. A large portion of time spent by the Story Creator module went into the creation of the three major canvas' on the page that were for the Animation, Object, and Visual Editor Modules. These were created and formatted correctly for what each team needed. The Visual Editor and Object Creator module implement a Kinetic canvas while the Animation team utilize an OCanvas. The canvas for Objects was directly implemented in order for objects saved to the Sharing Framework to be displayed in the canvas. The Story Creator module also added the functionality of clicking on the objects to make them glow and put them into an array for the Visual Editor module to implement.

2 Development Procedures

In order to complete Edith the Story Creator module followed the iterative, extreme programming, and the Boehm spiral model. We initially used the Boehm spiral by determining our objectives, identifying risks, testing, and planning our next cycle. We found quickly that this was not the most applicable because of the amount of changing requirements which forced us to throw away old code. It transformed into an iterative model because we

discovered that we had to repeatedly redo our entire module do to changing requirements and specifications from other groups and the project as a whole. As deadlines drew near we found ourselves switching to the extreme programming model in order to solve issues/bugs and to complete large functions of the project as a whole.

We began with the functionality of receiving JSON objects, adding a time to them, and passing them on to other groups. We also added the functionality to set a time to wait for a function (this was later transferred to Animation). Later our requirements changed completely and we began creating the base webpage view for the entire project. Then, we worked on incorporating the Visual Editor canvas into the main page, setup a main canvas, creating dummy buttons for different functionality, and the canvas for viewing objects. From there we worked on design for the login page, main page, and the registration page. We then worked on getting the Ajax calls up and working with the Sharing Framework. We assisted the Object Creator and Sharing Framework modules with saving and loading from the database as well as setting up the save and load for the entire project. We worked on connecting the various elements from the main page with the groups that they corresponded to. This included loading data from the database and displaying it on the object canvas.

Our process was effective at developing functioning software but very ineffective because we created software that was never used or even thrown away. We mostly implemented console log testing and used JSHint. Our program works because the main page displays all canvases as well as allowing the other modules to function correctly on them. The object panel at the bottom of the screen also displays the objects correctly. Console log testing was effective for showing us how the page interprets the information that we were displaying. JSHint was effective at showing us syntax errors in our code. At times console log testing was ineffective at pointing out why our code was not functioning.

The following responsibilities are based on the components of the final product:

Nate was responsible for how the various canvases were drawn on the Edith main page as well as their relative sizes to the window. He imple-

mented the canvas' resizing capability whenever the webpage window is resized. He also worked on various bug fixes, clean-up of antiquated code, and implementation of many small functionalities (ie. button method calls). He helped implement the functionality of getting objects from the object creation page into the object canvas below the main canvas. He then wrote and implemented the tutorial on the webpage.

Brandon was responsible for helping the Object Creator module and Sharing Framework gain access to the database with javascript. This included writing load and save functions for the main page that sent and retrieved data. Brandon was also in charge of porting the Visual Editor code and the Object Creator code so that it would function on the main page. Finally Brandon was in charge of loading data from the database in order to draw objects on the object canvas that were clickable.

3 Requirements Evaluation

4 System Design and Architecture

5 Individual Reflections

5.1 Brandon Roberts

Edith was a challenge from start to finish. I believe that the most voluminous issue was directly organizational. When we began this project we believed that we had an idea of what each team was doing, but in fact we had no clue what "Story Creator" actually meant. We had a description of our module that included things like "allows the user to specify animations and other events" and "utilize the Visual Editor to specify actions performed by objects provided by the Object Creator." This was completely fine to begin with and made sense but unfortunately the project did not end up functioning in this way. This meant that time and time again the Story Creator module had to modify/completely scrap large portions of code simply because it would not function correctly with other groups. One large issue we had tied back to the fact that "This module is the least able to function as a stand-alone system." There were times when we learned that the Object Creator module

was changing how they save objects, Animation was changing how they interacted with the sprites, Visual Editor changed how they displayed and moved functions, and the Sharing Framework changed how the tables were set up in the database and what we needed to do in order to retrieve information. This all comes together to show that it is difficult to be the final piece of the puzzle.

I also encountered many issues with retrieving objects(sprites) from the database and displaying them inside the object panel. I will take some very useful database debugging and a much better knowledge of Ajax with me. I have also learned a great deal about databases and php as well as javascript. These three things together will be very useful in the future because of how prevalent they are in the field of web development and computer science.

In order to complete the project I originally attempted to implement the use cases in my requirements document. That quickly became outdated so I started to use the "adapter" design patten to try and piece together the other groups software. If restarting on this project I would attempt to have a long session with other groups to go through what they planned on doing and how we could put it together. A large problem was that many pieces could not be put together in the way that individual groups had originally designed. If I were to continue on this project I would add functionality to actually work.

5.2 Nate Olderman

The biggest challenge I encountered during the development of this project is the dependency we had on the other groups. Because of this it was difficult at first to define clearly what we were responsible for. We tried to overcome this by talking to other groups as well as the professor in an attempt to define, as clearly as possible, everything that we needed to do. But in the end what really helped us to overcome this issue was the restructuring of our group (story creator) and the visual editor group. That helped to clearly define our groups' responsibilities and functions. In the future I think I would use this experience when creating the initial requirements specification which would help to solve this difficulty earlier in the process.

Another difficulty we had was that we also found ourselves repeatedly starting from scratch because of changing functionalities and requirements. This is something that we were not able to completely overcome because of the

dependency we had on other groups throughout the entire project, but we got better at communicating with other groups to spot possible future changes and try to account for them. In the future I believe I will use this experience in communication to start the process differently so we can just alter code as opposed to starting from scratch.

Regarding technical challenges, the biggest challenge was actually many small challenges. Because many of us had not worked with html, css and javascript before, some of the biggest challenges had very simple solutions. For example, early on the biggest issue we encountered was how to use variables from other javascript files, in our javascript file. Now that we have practice with these languages, issues like that are obvious. But they did cause big problems early on. Learning from these problems are definitely skills that I will bring to future projects.

Another large technical difficulty we encountered was obtaining the created objects from object group through the sharing group. We needed to obtain these objects so we could display them on the object canvas below the main canvas, and then pass the selected objects to the visual editor group. We overcame this through testing and trial and error. In the end I learned a lot about JSON and Ajax that I can use in future projects.

Personally I mostly applied the iterative software development technique. I tried to repeat previous cycles of development, this technique really helped when we were forced to redo parts or all of our code. This strategy failed when we were short on time. I felt that we did not have time to repeat the same steps. It was at this point that I implemented more of an extreme programming software development style. This strategy really helped me focus and finish large chunks of our program's functionalities but fell short when we had to redo parts of our code, because I hadn't documented my development as well. The end result of this case was that more code than what might have been necessary was rewritten.

If I were to start this project over again the very first thing I would make sure to do differently would be to communicate more with other groups during the requirements specification stage in order to catch the similarities in our requirements. I would also do a lot more research and teach myself more about javascript, JSON, and Ajax earlier on in the semester.

If I were to continue on working on this project I would work with other groups in order to get rid of repetitive code and hopefully make it so the code runs more efficiently. The next thing I would want to add to the project would be to have the object names above each object on the object canvas.

This isn't something that is necessary but it seems like it would be a nice touch. I would also want to implement a better tutorial system that would point to various elements of the main page and explain how to use them.

6 Glossary and References