Postmortem for Hello There

This is the first time that I spent 3 months on a single project and totally completed by myself (with a lot of kind and precious people's help) No matter how assertive my resolve was back to the beginning of this project, after three months of struggling many things become dumb to me. I don't know it's a bad thing or a good thing. Anyway, I didn’t die, jk, survived pretty well in fact. It's not that difficult the project itself, at least, on the one hand, the story was written in advance, on the other hand, I didn't challenge myself on coding part too much. (didn't dare to take this risk.)

In this postmortem, I'm going to talk about the development process of both coding part and writing part. In the last, I will bring up several ideas or say prospects that I'd do to enhance my project.

Coding, coding, and coding...

If anyone wants to be a game designer or gets involved in the video game industry, they must be either artist (music/art) or technician (programmer/ designer). It's something inevitable, but I started facing this fact until my Intermediate Programming class in the last winter. It's tough to realize and admit that sometimes you are not as good as your peers. But, yeah, I guess that's the first step of learning. Get to know what you don't know and go to study them.

This project "Hello There" has three version updates on the source code aspect. In the beginning, the story was built in Twine 2.0. I didn't use too many advanced techniques of Twine. The outcome is just a single compressed HTML page with an inline style sheet and JavaScript script. The first version was completed by the end of 2017.

The second version is also only a blueprint or say a prototype. But I wrote all codes by hand. It's still a very simple and plain HTML page. I created an interface and grouped all the elements I need. Also added the footer information to the page. I spent much time because I was tripping by my clumsy skill of website design. Thankfully, it's a long way to fetch the 'perfect' level but works. I show this demo to my director and got the permission going to the next level.

The third version AKA the last version. This time I re-started the program based on the inkdML. Since I didn't make big changes in the function part, I carefully knitted my story into multiple nodes and animation box. It's extremely exhausting. The outcome ends up with more than 3.5k lines of code, and that's just the story part, the extra stylesheets aren't counted in.

As my director said, InkdML does help me save a lot of time. It's just for someone as a rookie like me, it's the least risky to do hard coding rather than building a code LEGO castle.

I've learned much new knowledge through this development process, even though I didn't implement all of them in my project. (By developing the project, by learning new techniques, it's a little dangerous to spend time on testing new features. for me at least.) Back to the last winter, I wasn't even awakened, didn't know where I shall go. But starting from this project, the route under my feet is so clear and goes to a distance far, far away.

People always say programming is about problem-solving. I think I also dig into this part, not only on the knowledge level. Breaking down a big problem/chunk into small steps; Inspect every detail when running into bugs; If the first approach doesn't work, coming up with the second and the third one; Ask Google, StackOverflow...

I'm always eager to achieve authorities' compliment. I know this is very childish. But programming is different from all the things I've done and all class I've taken before. It's more like a game playing with myself, competing against myself. It's cool, challenging, and fascinating.

About the story

[Spoiler]

During this part, I will write down some thoughts about the truth of the story compared before and after.

Messanger style text game is a genre that currently becomes popular. I was attracted by Lifeline series, also have heard of Mystic Messenger. That's where my inspiration came from, but not from the famous Zork series.

It's funny. I think these two different kinds of games reflect people's different needs at different times. Back to Zork's times, there's no Amazon Alexa but ELIZA. A computer is a rare thing for people. Game players wanted a virtual world with imaginative-darkness-space. Zork's briefty is very impressed. Compare with today's text games, they are more wordy, delicate. It feels like players' loneliness becomes something more important than their curiosity.

Hello, there is a messenger game I simply wrote for myself. Though many things have been changed during this year, I can still recognize the emotion and the expectation behind the story. I used to have that kind of thought: if I can be spared from the pain by forgetting about a certain person/eliminating someone's existence from the memory, would that be better for me? Not only me, I've even heard about a similar film. I bet many people think about this of the pain that they don't want to face and remember. To escape.

Back to I just started writing the story, I think that would be better if the actress could get a chance to time-travel, to warn herself in the past to avoid this pain. But after a long half year past, I switched my standing. I thought this me won the duel against the past myself. There's no need to forget. But the truth is, I guess, I didn't beat the coward myself but the pain forgets and departs me. So, the story also changed behind the curtain.

It doesn't matter with the part that already implemented into the program. They are just something I may write in the future. In the original story, Cecile is doing a real time-travel-conversation with herself from four years later's future. She pretends to be Snowy, one of her old net friend, to get close to herself and try to warn her from meeting with Mr.Valentaine.

Now I think it's better to make it a fake-time-travel. Thinking about changing the past is purely in vain. But if we see every human is a function (such as in JavaScript), and everything we meet is the different parameters, human's memory, and experience sound like something computable. If we find the key variable and delete it, will we forget the entire code block?

It's a little quirky to explain. This time, Cecile knows that the experiment won't change the fact that Mr.Valentaine left her and died. She just wants to re-write her memory by talking with the past herself. (Well, of course, she will realize how ridiculous and coward she is and affected by that bold, fearless herself in the past.)

I used to write a lot when I'm in the college. However, it's obviously different than writing in English. Many times, I know that I should have spent more time on learning programming. I just can't give up telling a story. I'm not talented in language studying. So, sometimes you can see many weird and strange, and nobody-knows-where-it-comes-from English expressions gather together to make a scene. I enjoy the air of freedom after I came to the US. Because I feel I'm standing on a special intersection, far away from my old family/friends circle, also a little far away from American society. I feel the same when I write in English. It's like being dropping into the void.

But still, the lack of proficiency in English is something troubles me at the last stage. Because I wanted to code all nodes firstly and do the proofreading later. But the timeline ran not as I planned. It's impractical to do an entire proofreading of a creative writing project with serveral people. Everyone would have different perspectives. It's also vital if I underestimate the amount of the work.

What happens next...[prospect]

1. Enhance the presentation part:

Build a new independent website to introduce the project and write some other information. Rewrite and reorganize this postmortem. Sort the development log with suitable markdown and update the Github repo.

2. Get to know the techniques:

Search and learn how to implement the styles that I proposed during the development process. Because it's not that ideal to write connecting functions in InkdML, so the styles part is the point. Write suitable mobile size style using media query, use font-awesome, and pseudo-class/elements to enhance the UI.

3. Learn more about Front-End:

I want to know how far I can push myself.

4. Write the sequel after the skill is satisfied:

Read more books, learn more about storytelling.