CMPS 5153
Ethics Game Team
12/5/2019

Ethics Game Final Report

1. Introduction

The main aim of this game design is to create an environment in which a student can learn Software Engineering Ethics in a recreational way. The game has three levels of difficulty (easy, medium, hard) in which each level starts with a mini-game followed by multiple-choice Ethics Questions. To progress through the levels, the user must have little knowledge about ethics. We have used the agile method for developing this game.

1.1 Project Overview

The game has multiple screens. The very first screen displayed to the user is the home screen, this is the main screen of the game that has various options for the user to choose from. They can decide whether they want to begin the game, go to the help menu, look at the credits, or exit the game from this screen.

As for the actual game itself, there are 3 levels of difficulty that are associated with the inherent difficulty of the 3 minigames that the user will play. When the user selects a difficulty, they will play the minigame associated with said difficulty, then after reaching the win condition of said game, will go to a question screen where they are asked an ethics question and given 4 possible answers.

If the user answers the question correctly, they beat the difficulty. If they answer the question wrong then they return to the home screen and must try again.

2. Project Deviations and Issues

Across the project, the team ran into a number of issues, both technical and schedule related, that forced us to make difficult decisions in order to get the game ready in time for the final presentation.

2.1 Configuration Management Issues

The team encountered a number of issues involving configuration management where different parts of the project, the entire project in one case, got wiped from the repository due to unforeseen interactions with Unity. The first time, when the project got wiped, we took our local repository and pushed it to the primary branch manually to override the previous changes where it had been deleted. The second instance, where a number of changes to scenes had been made, had to be redone manually.

2.2 Deviations from Specifications

Original specifications stated that we would implement only one of the three difficulties, and randomly select which minigame to present upon play for a varied playthrough. We instead, due to time constraints, rated the difficulty of each minigame and tied them individually to each difficulty type on the level select screen. We were also unable to get a persistent score for each player implemented in time, so it has instead been added to the backlog.

2.3 Known Issues Remaining

Aside from the deviation from specifications, there are currently no known defects present within the project. There are still items that, as per the original scope, have not been implemented such as game saves and educator input, but the game is in full functioning capacity within its current scope.

3. After Action Analysis

The team learned a great deal across this semester in regards to Unity, the Agile process, and game design in general. There are a number of items, in general, that we would have implemented differently on a second time through, and lessons we learned that would lead us to not make the same mistakes twice.

3.1 Possible Future Implementations

3.1.1 Education Manual

Study guide for the players where the players can learn Software Engineering Ethics before playing the Ethics Game.

3.1.2 Score Count

The cumulative score could be generated at the end of each level, and stored somewhere so that it can be accessed and displayed by a menu. Highscores could also be implemented in this same fashion.

3.1.3 Save Game

An option to save the game by the users could be implemented.

3.1.4 Customized Questionnaire

Educators could change the questions according to the concerned subject of Software Engineering Ethics.

3.1.5 Educator Sign-in

The educator Sign-in feature could be implemented so that the educator can access the questionnaire database to make changes accordingly.

3.1.6 Users Sign-in/Register

Users' sign-in or register feature could be implemented so that users can resume their game any time in the future.

3.1.7 Effective Sound Effects

Sound Effects could be added to the Ethics Game for questionnaire screens as well as character movements in the mini-game.

3.1.8 AI implementation in Pong game

AI feature could be implemented to one of the controls in the Pong game.

3.1.9 More Minigames

More Minigames could be implemented in the future to increase the length of the game.

3.2 If We Were To Do It Again

The lessons we learned have provided some insight for tools we might have used differently if we were given a similar project a second time.

3.2.1 A Different CMT

The issues we encountered with using GitHub as our configuration management tool would likely lead us to try another CMT all together. Had we done a bit more research into Unity and GitHub specifically, we would have found the known issues and possibly gone with either manual file management, with something like Dropbox, or with a git competitor.

3.2.2 Better Scheduling

While the team did manage to meet every week on Thursday, then handle individual meetups with our pairs, our sprint scheduling did have to be revised a few times. We underestimated the length of some of the tasks, due to our inexperience, and on a second time through, we would be a bit more pessimistic in regards to estimating tasks we're unfamiliar with.

3.3 What We Learned

Each member of the team, before this project, had little to no experience with Unity. Aside from learning about that tool, and learning more about the inner workings of GitHub, many of us pulled different lessons though.

3.3.1 What Daniel Learned

Coming into this project, I had little experience with the agile process. Every group project I've done up to this point has really been ad-hoc, but in retrospect, there were elements that I had used then that made it really easy to transition into the agile process. In addition, I had never really done pair programming before, and that ended up being far more productive than I had initially thought it might. Having a second set of eyes dedicated to just analyzing what's being typed in, whether you switch off or not, really does lead to more thoughtful, efficient code.

3.3.2 What Sai Rama Krishna Tummala Learned

I don't have experience in developing games before. Eventually, I have learned how to develop the game in unity. I also learned to share ideas with the group and learned to do pair programming. It made it easier to make the game more efficient as we were chipping in the ideas together. I have learned to complete the tasks within the timeframe of each sprint.

3.3.3 What Neeharika Kasarla Learned

I have learned to appreciate and accept the work done by the other team members. I was able to recognize my abilities which could contribute towards the project. I have learned how the agile development process helps in developing a system in which regular meetups are much needed. I was able to improve my development skills and also learned new skills that helps in management.

3.3.4 What Ben Learned

I had worked with games before, so I knew the general lingo, but I learned a good amount about Unity. I basically learned Unity's take on game making (well with as much of the tool that we used).

I also learned that pair programming is pretty nice, it's really good to be able to bounce ideas off of another person rather than having to ruminate within your own mind.

Another thing I learned was that when someone offers to work on something, they usually end up doing it differently than you had expected. This isn't normally a bad thing, because it's good to see other people's perspectives and hopefully that will lead to you opening your mind, benefitting the creative factor and all the potentially different perspectives of looking at the project.

The final thing, which I am embarrassed to admit, is UML diagrams. I wasn't very knowledgeable about UML diagrams, and I had to create one for this project. Dr. Stringfellow was nice enough to let me borrow a book and I read up on them and made one.

4. Summary

In the beginning, our group set out to make an educational game that would teach it's player a bit about software engineering ethics, and we did accomplish that in the end. We had may bump along the road, all technical, that did inform how we'd approach this differently on a second time through, but nothing we experienced was overtly negative. Our team jived during our coding and work sessions, and the entire project was more fun than it was work, which ideally is how it should be; do what you love, you'll never work a day in your life.