

EPICS: Engineering Projects in Community Service
Two-and-a-Half-Ryans

Albert Badalov, Kyle Angelbeck, Lauren Marx, Max Ryan, Ryan Ooley, Ryan
Tsai

2019 Spring

Thycco Mode - The Elemental Table

Table of Contents:

Chapter 1: Introduction	Page 2
Chapter 2: Requirements and Specifications	Page 2
Chapter 3: Architecture	Page 2
Chapter 4: Design	Page 3
Chapter 5: Implementation	Page 3
Chapter 6: Project Organization and Management	Page 4
Chapter 7: Future Work	Page 5
Chapter 8: Bibliography	Page 6

Chapter 1: Introduction

This project was brought to us by Sycamore Services with the hope that we can develop a game with mechanics focalizing on social skills. The idea for this game stemmed from a current attendee of Sycamore Services who has autism and was pushed for by the current employees. The objective of the game is to provide a way for the attendees of Sycamore Services to enhance their social skills by playing through our game which specifically focuses on characters facial expressions. Every action in our game has its consequences, and if you aren't making the right choices in regards to what would qualify as a proper response, then your score of "proper interaction" goes down. A bad score would result in a bad outcome or a loss in the game and a good score will result in better outcomes in the game other than winning the game. In this report, you will read about our journey through this learning process and where we got to in the end.

Chapter 2: Requirements/Specifications

The functional requirements given to us are very open and non-restraining: make a video game, make it designed to improve social skills, and do this while providing something that is actually entertaining. The person who inspired the project, Charlie, also suggested that the game be reminiscent of Nintendo's *The Legend of Zelda* games, which are high-fantasy adventure RPGs. Ideally, technology would be included to the game that can detect tone and eye contact, but the client had no expectation of this being executed by this semester's end.

No real challenges came from fulfilling the requirements, as loose and undefined as they are, but several came to be concerning the team's expectations of the difficulty of the work. The largest constraint was time and the portion of work we had assigned ourselves. We wanted an hour's worth of gameplay, then when that became clearly unattainable, the hour desired dissolved to ten, then to a tutorial, and finally rested at simply a strong start with a clear direction to go.

Chapter 3: Architecture

The Architecture of how our whole development process went is very complicated. It'll be explained in a process type linear fashion.

Game Hardware: The game hardware is all reliant upon the client. How this works is that once the game is downloaded, it can be run. It's able to run on most systems other than very outdated hardware.

We used Unity as a our software and C# as our language.

Chapter 4: Design

The majority of the design decisions that we had to make were concerning the art styles our game would consist of. Only one of our team members had a background in art design and we are unsure of what the art skills will be of future members working on this project. Thus we decided to choose a art style that will be simple to create different sprites from scratch and from modifying existing sprites. We decided on using pixel art, using 32x32 sprites for characters. Since the basic structure of the characters are the same, it is very simple to adjust pre existing sprites to make new ones. All of the other sprites and backgrounds are created in accordance with the size of our characters.

Chapter 5: Implementation

Selection of implementation language(s):

C#

Coding standards and comments used:

Proper code based on implementation from what is relevant and updated from the current year.

Comments were the main comments needed in the code.

Implementation process and distribution of work:

Work was distributed by what everyone is capable of.

Albert Badalov: Focused on leading the team and keeping everyone focused. Main goal was to maintain proper file system and run the main Github and make sure files were properly updated. Other than that, maybe 15% work was put into Unity. He helped a lot with graphics and implementation.

Kyle Angelbeck: Did a lot of work with the graphics and also implementing code

Lauren Marx: Did almost all the graphics work and creation.

Max Ryan: Focused mostly on the story and everything that has to do with writing.

Ryan Tsai: Mostly was working on the main unity development of interaction systems.

Ryan Ooley: Did all the other work on Unity. Helped create the inventory and interaction menu.

Organization of the code base (e.g. directories, files, packages, classes etc.):

All of our code was maintained and updated on GitHub. All of our code was created on different and eventually sent to merge on GitHub.

Chapter 6: Project Organization & Management

Describe your team's organizational structure

We had a Team Leader, 2 Main Unity Developers, 1 Main Graphics Artist, 1 Main Unity Graphics Developer, and 1 Main Story Creator.

Explain the role and detailed contribution of each member in the team

Albert Badalov: Team Leader, Client Liaison, File Manager, GitHub Manager, Secondary Unity Graphics Developer

Kyle Angelbeck: Main Unity Graphics Developer

Lauren Marx: Main Graphics Artist

Max Ryan: Main Story Creator

Ryan Ooley: Main Unity Developer

Ryan Tsai: Main Unity Developer and Secondary Client Liaison

Clearly describe how the overall work was divided and carried out by different members:

The work was all divided evenly. But for those who shared roles, it depended on what the roles were. Mostly all work was divided equally.

Describe your project's management process (e.g. conducting meetings, scheduling, communication, planning, reporting, conflict resolution, etc.):

We would meet twice a week during the designated hours. We communicated either in class or GroupMe or by direct text.

Describe any related technologies and tools selection and use:

Unity, Piskel, Visual Studio, Visual Studio Code—Insiders

Include all Weekly Status Reports (WSR):

Included in the Zip File.

Provide a detailed user's manual (e.g. instructions on how to operate your system)

Double Click the SycGame.exe, specify the graphics type, then move around using W/A/S/D keys.

Chapter 7: Future Work

The work completed by the team this year is a strong start, but leaves much work to be done before resemblance of a final product:

- Finalized Story
 - Before any more game design can be completed, a fuller story should be fleshed out first, in order to know what needs to be accomplished in the game. As such, it should be a major priority to complete finer plot details, and the direction in which the designers want the game to go.
- Items and Mechanics
 - After finalizing the story, the next step is to implement the desired items and mechanics in the game, as well as finish those left by us, the now previous group. Weapon design, and coding to be able to use and activate the various armours and weapons will need to be done as well. Things like inventory and teleportation are done but need visual counterparts. Save files, stats and other game related implementation also needs to be done.
- Style, World Craft
 - Lastly, after the coding and the world building, there is the task of crafting it. Skins, maps, animations, cut scenes, etc. could be crafted to make the game look attractive and thematic as well as properly communicating what the happenings are to the player.

Bibliography:

“Rinmaru Games - Avatar Creators and Anime Games.” *RinmaruGames*,
www.rinmarugames.com/.