Project: Zombie Ruby Run
Authors: Kito Clark & Nick Pasquariello

Plan:

We wanted to address the problem of plain old boredom. But we also wanted to create an educational experience for users of our app. We figured everybody likes games so decided to create one. Our first step was to come up with a general theme for our game. We decided it would be useful to create an app that provides an educational experience for our players on how to survive a zombie apocalypse.

Once we had decided that we wanted to make a zombie game we had to plan how the character creation screen would look and what options we would provide the player. We also had to plan and design a “map” in which the character would have to explore to escape the building.

Initially the game was going to be entirely text based. But with the discover of ASCII art we decided to implement some basic graphics into our game.

After we a had a basic plan set out, we split the work up and decided which tasks each of us would do.

We had to be careful to include within our plan that we were able to integrate our separate codes efficiently.

Problem:

The problem we seeked to solve was boredom and lack of educational games about the zombie apocalypse.

Purpose:

The purpose of our app is to entertain and educate. The app is a game that not only provides us with some old school throwbacks and fun, but also can help the user prepare for the zombie apocalypse. The other purpose of making the app was to teach how to practically use Ruby to solve a real world problem.

Solution:

Our solution was to create a zombie escape mystery game where the user must find clues and fight zombies to escape from a maze.

Ethical Issues:

The game has some cut sequences and flashing color that could be dangerous to anyone who is susceptible to seizures. Standards state that you should avoid designing something with more that 3 cuts a second. The game has some sections that might startle or scare the user. If the user has a heart condition, they should be warned before using the app.

Technical issues:

Working in a team, we had to learn how the person was coding their side so that they would fit in together and run. If one person set up a section in one manner and we had not communicated with them, someone would have to rewrite their code, costing us more time. In future, we will use Github to better manage our app.

Another technical issue was to get the background music and sound effects to play and stop playing on both Apple and Windows operating systems.

Conclusion:

Both team members learned about how to practically use arrays, hashes, gems, and loops in the Ruby programmable language through the terminal interface. Team members presented the application to group and had a volunteer successfully work their way through the game.