

Turbo Enigma 3

Samuel Bateman, Kylee Bennett,
Sheena Hann, Tanay Choudhary

REST IN PEACE:

Enigma Emulator

Enigma Chess

We wanted to make a easily moddable roguelike

Let's be honest, games are fun to play, but hard to modify.

The few games that have made modification work?

- Minecraft
- Garry's Mod
- etc.

Our game holds as much data as possible, in text and image files

Rather than just holding assets in our data folder like most games, we planned to store everything but the core gameplay elements in the data folder, from enemy interactions to item data

- A modder could potentially add a new text file and a new sprite and add a new enemy to the game
- While we didn't get quite this far, we did make a pretty awesome procedural dungeon system :D

Technologies Used

Visual Studio 2017

- Purpose:
 - Simplify setting up a c++ environment for development
 - Give us a consistent environment across machines

Rating 4/5

- For those with windows laptops, it was awesome and productive
- For those with macs, it was hindered by our next technology...



Parallels Desktop

Purpose:

- Allow all team members to use a Windows environment

Rating 1/5

- Generally just doesn't provide a quick enough environment for visual studio

Github

Purpose:

- Project tracker, VCS Repo, Docs, Release Deployment

Rating 4/5

- Tighter integration with the project tracker and who you assign them to would help alot



SDL 2.0 (Simple Directmedia Layer)

Purpose:

- Allow for eventual cross platform deployment of game
- GPU texture rendering library

Rating 5/5

- Stays out of your way and lets you talk to the OS and remain performant



But wait where is Unity/Unreal
Engine/Source/Cryengine?!?!?

We didn't use one

- Game engines aren't the end all answer, especially for us
- Would have limited our ability to add on more moddability later on because of engine limitations
- Not very much time to implement our own :D

SDL is just a helper to hide individual OS's APIs

Methodology: Waterfall with Task Cards

- Frankly, we didn't have enough time to utilize agile because we just started this like 3ish weeks ago
- One of our team members has some experience with this exact tech stack, so it we weren't idealizing time frames
- Only regret is how much time other classes took which slowed down the project

Challenges

- Windows virtualization sucks :(
- Projects with poor testing quickly get overrun by complexity
- Not enough time to truly give the project the time it deserves

Demo Time!