Lab 04 Report - Gipson

# Introduction

UGC has become a large part of the industry, and having the proper tools to implement it into my own games would be a boon. By using file io and parsing ugc can be created outside of the engine and read in later, to allow players to create their own content.

# Methods

The methods section of the report is where you create a technical outline of your code. This should include how the program works, why it works, and what it does. A knowledgeable programmer should be able to replicate your lab with the details provided in the methods section of the report. You are allowed to include code, however this should be mostly essay based.

# Conclusion

UGC is easy to implement when you are doing basic things like reading text files, but will get incrementally more difficult as you add more options and possibilities. Keepings things limited is best practice but not always possible depending on your production goals.

# Post-Lab

1. They are more dynamic than arrays
2. You could use a streamreader and readline to add each line to an array or list. Avoiding the use of a fileinfo or textasset
3. One as a default behavior, the other as a modding style system for UGC
4. Not in the current context. If the file doesn’t exist the fileinfo will just hold an empty fileinfo. But if you never assign it yes. Except its unity so itll hold the default value.
5. By giving a gui with a string field. Player enters their string, that gets passed through to fileinfo. If it exist itll load, else itll give a message or load default.

# Code

Finally, you should include a hard copy of your code.