Zimmer, Michael J

ENG 2140-90

Process of

Object Creation

**Introduction**

Over the summer, I had been working on a personal project through Java to continue to learn and refine my understanding and skills in that programming language. This project is a simple text game, with a main screen, Player info card, NPC info card, and a JPanel that displays the current situation for the Player to react to. The Player interacts with the game with four buttons, though that is subject to change as I continue to develop the game. Through the different processes for the program, this technical description will focus on initial object creation. There are multiple custom object types created, which are Planets, Situations, Weapons, and NPCs.

A diagram of a process

Description automatically generated

*Figure 1 (By Michael Zimmer)*

**Discussion**

**ObjectCreation**

ObjectCreation is a custom class created for the name’s implication, creating objects. The four different objects are created in the same manner, though their purposes are all different. Inside this class, the different object generators are called to create Planet, Weapon, Situation, and NPC objects. The data flow of ObjectCreation can be seen in Figure 1.

**Generator Methods**

The methods called are planetGenerator, weaponGenerator, npcGenerator, and situationsGenerator. Each generator is responsible for creating the Planet, Weapon, NPC, and Situation objects, with their corresponding object requirements.

**Text File Storage and Array List**

Each object generator uses the text from specific text files for either the name for Planets, Weapons, and NPCs, or longer text for Situations. For all methods, it uses a try-catch to pull data from the text files, then add the text to a temporary String array list.

**Create New Objects**

planetGenerator creates Planet objects with two variables: name and difficulty. weaponGenerator creates Weapon objects with three variables: name, weight, and damage. npcGenerator creates NPCs with five variables: name, health, aggression, weapon, and alliance. situationsGenerator creates Situations with one variable: String. Each object is created via for-loop, which uses an integer tracker “for(int i = 0; i < File.size(); i++)”. When the “i” is no longer less than the file size variable, the for-loop ends.

**Object Array Lists**

Once the new object is created, the object is stored in an object array list created specifically to contain that certain object. The array lists are simply named: listOfPlanets, listOfWeapons, listOfNPCs, and listOfSituations. All four array lists are made public static, so they are able to be called on from other classes throughout the game. After each object is made, it is added to its corresponding array list, and the process begins again to create the next object until the for-loop is finished.

**Conclusion**

These different objects are all created prior to the rest of the game initializing. Their purposes vary, like mentioned before, but they all are equally important for the game to run correctly. The objects are necessary for the Player to interact with, travel to, and store in their inventory. This process is still being developed on as I refine the way it works and make it more efficient.