Adhem Azzabi

Professor Neeley

CIS 250

16 April 2020

RPG Characters & Inventory

For this project, I plan on working on a game application for my professional portfolio, which will have a fully functioning database integrated within it. The project will be based around a digital role-playing game (RPG) of my own creation, made with the Unity3D game engine and C# code.

The database will connect the raw data such as the player’s item & equipment inventory, player and enemy stats (i.e. Level, Health, Damage, Experience, etc.), various enemies, items such as potions, and weapons/armor. I find this will be incredibly useful to have inside an actual database as the data can become cumbersome, and additions are often made, such as adding additional items or enemies. I worked on a smaller scale version of a similar system and found that having the data in a less organized, non-queryable form makes scaling a project much more difficult.  
 The database will have a minimum of 5 tables as required, but more may be added as necessary for an organized structure. Currently, the five tables are “Character Stats”, “Enemies”, “Equipment”, “Potions”, and “Inventory”. Each of these has 5 or more fields, using primary keys such as “Level” or “InvSlotNum” (Inventory Slot Number), and there will be quite a large number of rows in each table.

One issue I foresee is connecting a database properly with the Unity scripts. I originally planned to use MySQL as the DBMS, however, I read that setting up databases with C# code inside of Unity is simpler and more straight-forward without using an outside DBMS. I plan on studying up on this further, as I still would like to use MySQL if I can find a simple solution. My research so far has dug up a micro-ORM called Dapper which seems to be a simple way of connecting SQL with C# or the Microsoft Entity Framework which is an API that maps data from a database.  
 By the end of this project, I expect to have a flushed out and solid database which is free of errors, easy to query, and overall nicely organized. This will result in a smooth transition into gameplay, with a well-coded inventory system. Overall, past the grade for this course, I take pride in this project as it will be something I hope to show potential employers as I continue onto developer interviews.