My final project for my Intro to Game Programming class was a turn-based role-playing game demo which includes a battle sequence and an overworld map. The overworld map was a last minute addition and is represented by a simple black background with a blue sphere representing the player. The game starts in an overworld map and every movement triggers a random number of 1 to 500. If the encounter number is less than 20, a random battle ensues.

Once an enemy encounter is triggered, the game segues into a battle scene. A random enemy with a different set of stats takes center stage (inspired by games like *Dragon Warrior* or *Earthbound*). There is a one in three chance of fighting a snake, a bat or a mouse. The enemyControl scripts stores the variables for these enemies. The snake has the highest amount of HP but poor defense, the bat has the strongest attack but has the lowest HP, and the mouse has the best defensive stat but also has the weakest attack. The player has a stat build as well including attack, defense, hit points and magic points. The player is given four options: Fight, Magic, Items and Run. “Fight” presents a standard attack that deals damage. “Magic” deals more damage at a price of some of your magic points. If the player exhausts all their magic points, the option becomes inaccessible. The player also has the option to run away, should they choose. If you or your enemy reaches zero health points, the fight will be over. Originally, I had two classes for the character “Hero” or “Mage”, with the latter having lower attack but higher magic points.



An important script for controlling the flow of the battle is battleFlow. This lets the game know what happens when the game begins, when it’s the players turn, when it’s the CPU’s turn and when it ends. This coincides with the damageControl script which tells the TextHUD on the bottom of the screen. For example, if it’s the player’s turn and they choose to attack, the Text HUD will read “You did 15 damage”.

  
  
One option I wasn’t fully able to implement were the items. I had three items slots for “Potion”, “Ether” and Elixir. After the player won a battle, they would receive one of these items at random. Potion would have restored Health, Ether would’ve restored Magic Points and Elixir would’ve restored both. Clicking the item button would’ve brought up the Item HUD and clicking it again would return to the fight. The text is still in the game but is not active. Although I was unable to program enemy animations, I did include animations for the attack damage that appear above the enemies head after every attack. I also had one animation available for the sword swipe which I wasn’t able to fine tune. It instantiates the swipe upon clicking “Fight”, but I wasn’t able to destroy the object once it served its purpose.



While not a traditional 3D game, a turn-based RPG was a game I’ve always wanted to make. I was able to take what I learned in the user interface (thanks in part to my Game Development Practicum) and create that sort of game. All of the core gameplay mechanics are there and most of the gameplay does work, while some could be fine-tuned if I had more time.