```cpp
/*
Daniel Avila    Section 19      February 5th, 2020
Lab2: RPG
Description: This lab will simulate a turn based RPG that will receive user input for the turn of hit or run.
*/
#include <iostream>
#include <time.h>
using namespace std;
// These functions allow for the code to be condensed into a simpler flow of code for understanding
char menu();
int makeAttack(int enemyHP, int playerAttack);
void takeHit(int &playerHP);
//These variables are set for the Player's and Enemy's HP and for the Player's decision
int playerHP = 20, enemyHP = 10;
char response;
//The main function processes the background layout of the game in simple codes
int main()
{
    srand(time(0));//This is used to generate a random number

    cout << "Starting the RPG!" << endl << endl;
    cout << "You spawned with " << playerHP << " health" << endl << endl;//Player starts wiht 20 HP
    //This do-while is used to iterate the choice menu, 'a' or the 'b' option as long as the enemy or player is alive
    do
    {
        cout << "An enemy lies before you, what will you do?" << endl;
        cout << "\t'a' Attack or 'b' Run" << endl;
        cin >> response;//Receives the user input to decide which route to go through in the if-else
        cout << endl;
        //This if-else loop is used to go down the route of the user's input and tests if it's valid as well
        if (response == 'a' || response == 'A')
        {
            menu();//If the user chooses 'a' then it will go to the menu function and proceeds its code
        }
        else if (response == 'b' || response == 'B')
        {//If the user chooses 'b' then it proceeds with the surrender of the player and the RPG is over
            cout << "You ran away safely" << endl;
            cout << "Game Over" << endl;
            break;//Break is used to stop the iteration of the do-while loop after the option of 'b'
        }
        else
        {//The data validation part is here and makes sure it's just either 'a' or
```

```
            'b'
43              cout << "That is not a valid choice." << endl;
44          }
45      } while (playerHP > 0 && enemyHP > 0);//Once the Player and Enemy HP goes below 0, it'll stop iterating
46                                                  //But as long as it's above 0, it'll keep iterating over and over
47      system("pause>nul");
48      return 0;
49  }//This char function is in response to the user input because it's the "game" part of the RPG
50  char menu()
51  {
52      int playerAttack = rand() % 3 + 1;//Using the random number from main, it attacks from 1-3
53                                          //And the +1 makes it so it isn't 0 when doing the damage
54      takeHit(playerHP);//Calls the function of the enemy doing the damage and it passes it by reference
55      enemyHP = makeAttack(enemyHP, playerAttack);//Since the player function is passed by value more code is needed
56      if (enemyHP <= 0)
57      {//Once the enemy has been slained or downed to 0, the ending will be displayed that he has been slained
58          cout << "\nYour foe has 0 hp..." << endl;
59          cout << "Your foe has been slain!" << endl;
60          cout << "Game Over" << endl;
61      }
62      else if(enemyHP > 0)
63      {//As long as the enemy is still alive it will keep iterating the damage report and HP left after the attack
64          cout << "You strike your foe for " << playerAttack << " damage!" <<
              endl;//Attack damage
65          cout << enemyHP << " HP remaining..." << endl << endl;//Hp left
66      }//It displays the amount of damage done to the foe and how much Hp is left
67      return response;//Since it's not a void, it needs to return a value in which in the function is: char returns char
68  }
69  void takeHit(int &playerHP)//This function's parameter is being passed by reference
70  {//This function is a void so it won't return anything but it's purpose is to reference the Player's HP
71      int enemyAttack = rand() % 5 + 1;
72      //Along with the random number, it generates an attack damage for the Enemy from 1-5
73      cout << "The enemy strikes you for " << enemyAttack << " damage!" << endl;// Attack damage from the enemy
74      playerHP = playerHP - enemyAttack;//Calculates HP going down from attack
75      cout << playerHP << " HP remaining..." << endl << endl;//HP left for the player
76  }
77  int makeAttack(int enemyHP, int playerAttack)//Needs to return a value of int, and
```

```
      it has two parameters so that it
78  {                                                   // allows for the attack to be
    overwritten since it's passed by value
79      enemyHP = enemyHP - playerAttack;//Calculates the attack from the overal HP of
          10 of the enemy
80      return enemyHP;//Returns the value of the Enemy's HP since it needs a new
          value after every iteration
81  }
```