CSIS 3280 – Lab 02 - A little terminal game - Hack n Slash

Visual Studio Code

1. Using the provided template, create the PHP application according to the requirements section, please also pay close attention to the structure.
2. Zip up your solution according to the Submission Guidelines and submit to blackboard by the given date and time.

Purpose

The purpose of this lab is to further solidify some of the PHP basics and additionally the basic types of arrays and functions.

This Lab is a simplified version of the original, if you want to add additional functionality at player levels, monster levels gold and a shop where you can gain gold from monsters or searching and buy better weapons that inflict higher damage etc.

Solution

- The user can use the following commands:

  (f)ight (r)un (q)uit

- After every turn the user is presented with their own status and the status of the current monster they are engaged with:

```
Player Stats: Name: [Your Name] Health: 100
Monster Stats: Type: Dragon Health: 100
```

- The user can quit at any time.

- The game is based on a 20-sided dice.

Requirements

The following are the Lab Requirements, please pay attention to them, not following requirements will result in a zero mark for your Lab. For Sample output please see the appendix.

- The probability that a player wins against a monster is 50%

- Regardless of the outcome the damage either to the player or to the monster can be calculated as follows:

```
damage = (20 - [20 sided dice roll]) * 5
```

- The probability that a player can run away and get some rest is 70%.
- If the player runs, the outcome of the 20 sided dice roll result should be added to their health.
- If the player is unsuccessful at running away they are forced to fight the monster.
- The player can fight as long as they have health, once their health is at 0 they are dead and the game ends.
- Once one monster is killed a new monster is randomly selected.
- In order to win the player must defeat all three monsters.
- Player data must be specified in config.inc.php and must be stored in an associative array.
- Monster data must be stored in a single array that at the first level must be numerically indexed.
- The player must be notified when a monster is killed.
- The player must be notified what type of new monster is selected.
- You are not permitted to put any game logic into the main file, you may only collect a command and execute functions.
- You must stick to the structure outlined below.
- You must use the provided functions outlined below

Structure

File Name	Description
Lab02_SHi_56789	This is the main file I will run when I run your lab.
inc/config.inc.php	This is the configuration file, it contains all the configuration for the application including the Player information, monster information, hit and run probability and the number of sides on the dice
inc/monsters.inc.php	This file stores all the information about the monsters
inc/player.inc.php	This file stores all the functions related to the player
inc/logic.inc.php	This file stores some of the logic with regards to the game that is not directly related to either the monster or the player.

Functions

Function	Description	Entity
YouWin	Notifies the user they have won and exits	Player
YouLose	Notifies the user they have lost and exits	Player
run	Allows the user to run acts on the outcome	Player

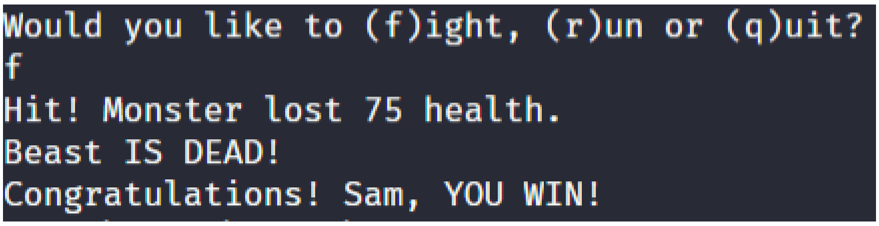
Function	Description	Entity
playerDamage	Removes the health points according to the calculated Damage for a player. Acts accordingly if the Player's health has run out.	Player
getCurrentMonster	Ensures that a an appropriate current monster is selected, if the monster is dead then a new monster is selected, if all the monsters are dead youWin()!	Monster
fight	This is the main function for the game, calculates the WIN ratio and applies the damage to either the monster or the player.	Misc
displayStatus	Prints the status of the player and the monster (see above)	Misc
rollDice	Rolls the dice given the configuration and returns a number.	Misc

Hints

- If you are struggling with your PHP environment, it might be best for you to use a lab computer.
- Figure out how to make a variable accessible to all parts of the program, its bad practice.
- You can calculate percentage of a dice roll by dividing by the number of sides and multiplying by 100.
- You can use the rand() function to generate a random number.
- Using the "global" keyword may help.

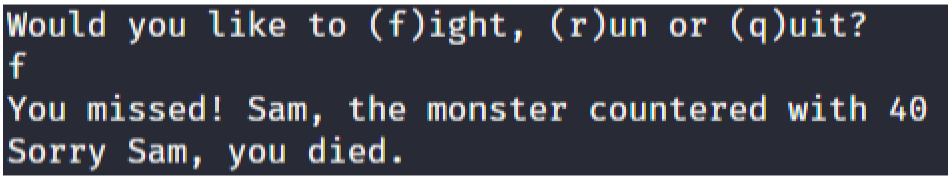
Appendix

Player won:



```
Would you like to (f)ight, (r)un or (q)uit?
f
Hit! Monster lost 75 health.
Beast IS DEAD!
Congratulations! Sam, YOU WIN!
```

Player lost:



```
Would you like to (f)ight, (r)un or (q)uit?
f
You missed! Sam, the monster countered with 40
Sorry Sam, you died.
```

STOP! - This is a pre-assignment submission checklist!

- Did you follow the naming convention for your files?!
- Did you follow the naming convention for your folder?!
- Does your submission work on a lab computer?!
- Double check **before** submitting
- Did you conform to the requirements outlined?]

References:

- Base model provided by David Campbell
- Monsters Reference - https://en.wikipedia.org/wiki/Creature_type_(Dungeons_%26_Dragons)
