Assignment 6: 100 points

This assignment covers module 6 and module 7, and module 8 topics. To write the programs below you will need to demonstrate an understanding of the following topics:

- **Objects**
- **Functions**
- **Arrays**
- **Random Numbers**
- **variables**
- **simple data types**
- **expressions**
- **loops**

Please refer back to the module lectures if you find you have any gaps in your understanding for these topics.

Introduction

> Emboldened by your newly learned skills, you decide to face Scarlet Byte alone. Unfortunately, you are defeated swiftly by Scarlet Bytes notorious body guard, Input Error, known for his “Devils Kick”, a swift and deadly style which has never been defeated. Returning to Iron Professor you study even harder in a form of fighting known as the “The Eight Deadly Venoms”, named after the eight mythological JavaScript features that the fighting style emulates. Now that all his pupils have mastered the Eight Deadly Venoms and Transcended to the Golden Army of Software Way, Iron Professor now plots the final duel with Scarlet Byte. The Golden Army approaches Scarlet Bytes fortress at dawn. With armor gleaming in the sunlight, they sound the battle cry …. JS Rocks!!!! They descend upon Scarlet Bytes Army and the battle begins. To defeat Scarlet Byte, the Golden Army must combine all the Eight Deadly Venoms. Will you emerge victorious, or will the Seven Kingdoms be cast into years of darkness? Let the battle begin.

Create a script called battle.js. In this script code the following:

A. Class and Constructor Creation (25 Points)

Player Class

Create a constructor function or ES6 class for a Player object. The Player object should have the following properties:

Name: Holds the name of the player. Pass into the constructor.

Health: Defaults to 10, do not pass into the constructor, just define and set to 10 in the constructor

Strength: Defaults to 2

Weapons: An array of weapons objects. Should be passed into the constructor.

Weapon Class

Create a constructor function or ES6 class for a Weapon object. It should have the following properties:

Name: Holds the name of the weapon. Passed into the constructor.

Damage: Defaults to a random number between 1 and 5, do not pass into the constructor, just define in the constructor and set it.

Enemy Class

Create a constructor function or ES6 class for a Enemy object. The Enemy object should have the following properties:

Name: Default to “Enemy”

Health: Default to 5

Strength: Default to 2

BattleSimulation Class

Create a constructor function or ES6 class for a BattleSimulation object.

Give it the following properties:

Players: An Array to hold the players, Initialize it in the constructor to an empty array

Enemies: An Array to hold Enemy objects. Initialize it in the constructor to an empty array

B. Methods to add (50 Points)

Player Class

Add a function to the Player **prototype** called "applyDamage". The function takes as input an integer and subtracts that amount of points from the players health property.

Add a function to the Player prototype called “isAlive”. This function checks if the players health value is greater than 0 and returns true if it is and false if it isn’t.

Add a function to the Player prototype called “attackWith”. This function should use a random number between 7 and 0, to select a weapon from the weapons array property, at that index and returns that weapon.

Enemy Class

Add a function to the Enemy **prototype** called "applyDamage". The function takes as input an integer and subtracts that amount of points from the Enemys health property.

Add a function to the Enemy prototype called “isAlive”. This function checks if the Enemys health value is greater than 0 and returns true if it is and false if it isn’t.

Add a function to the Enemy prototype called “attack”. This function takes as input a player and calls the applyDamage of the player, using the Enemys strength as the input.

Weapon Class

Add a function to the Weapon prototype called "attack". The function takes as input a player instance, and an Enemy instance. The attack function should implement the following algorithm:

In a loop, while both the player and the Enemy are alive (use the isAlive methods), do the following:

1. Calculate the actual damage by multiplying the strength of the player times the damage value of the weapon.

2. Call the applyDamage function of the Enemy object and pass it the actual damage value you just calculated.

3. Call the isAlive function of the Enemy object. If the Enemy is dead, exit. If the Enemy is not dead, call the attack function of the Enemy and pass it the player object.

BattleSimulation Class

Add a function to the BattleSimulation class prototype called “createEnemies”. In this

function, use a loop to create 20 Enemy instances and populate the Enemies array property.

Add a function to the BattleSimulation class prototype called “createPlayers”. In this function, do the following:

Create eight Weapons objects. Create a variable called **“weaponsCache”** and add the 8 weapons you just created to it.

Create 5 player instances and add them to the players array property.

Add a function to the BattleSimulation class prototype called “run”. In this function, implement the following algorithm:

Display “Simulating Battle”

Call the “createEnemies” function to create the Enemies.

Call the “createPlayers” function to create the players.

Until all the players are dead or all the enemies are dead:

Select a random player from the player array
Select a random enemy from the enemies array
Call the attackWith Method on the player, to get a weapon to attach with.
Call the attack method on the weapon and pass it the current player and current enemy.

When the loop is completed, display a list of all players currently alive and if there are more players left than enemies, display “Congratulations, you have defeated Scarlet Byte” Or “Sorry, Scarlet Byte has defeated you and conquered the free world.”

C. Test Program 25 points

Create an instance of the Battle Simulation class and call the run function.

Save battle.js in a folder named Assignment_4. Zip that folder and submit in the Assignment 4 Dropbox.