JAVASCRIPT LABS

Over the course of the next few labs, you will be creating a simple role-playing game. The game involves two players - the prompted user (you!) and the Almighty Grant. The game consists of simulated attacks that reduce the total health points of each player.

The game will be built in 4 stages:

1. Set up the basic functionality of the game
2. Complicate the game by adding **function**ality
3. Convert characters to objects
4. Add a front-end to your game

JAVASCRIPT LAB PART 1

Task: Prompt the user if they would like to play the game with two characters - the user and the Almighty Grant. If yes, prompt the user to name their character. Run a while loop that will iterate until either the character has beat Grant three times or the character has been defeated.

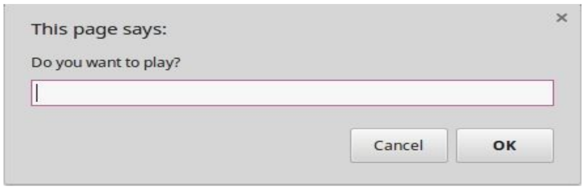
What does the application do?

1. The user is prompted to play a game. If the user chooses yes, the user is prompted to enter his or her name, and the battle begins. Otherwise nothing else happens.
2. The game will use a `while` loop to simulate a turn-based fight between the user and Grant.
3. The user starts with 40 “health points.” Grant starts with 10 “health points.”
4. Each iteration of the `while` loop will remove random numbers of health points (either 1 or 2) from both the user and Grant.
5. Each time Grant’s health points hit 0, he is “defeated” and the user gains 1 “win.” But Grant’s health points are reset to 10. Note: the user’s health points never reset.
6. The game ends either when either a) Grant has been defeated 3 times (i.e. user has 3 wins) or b) the user has been defeated (hit 0 health points).
7. When the game is over, the application logs the winner.

Build Specifications:

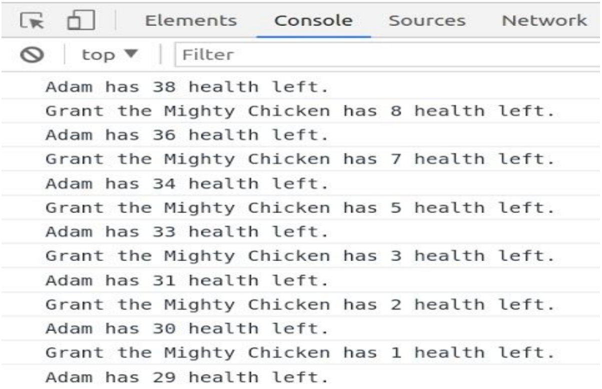
1. The application must prompt the user for his or her name and use it throughout the game.
2. The application logs the progress of the fight after each iteration of the loop.
3. **Hint:** Save asking the user if they want to play for the last part of the code you write. This way you won’t have to type “yes” every time you test your program.

Console Preview:



This page says:

Do you want to play?



```
Adam has 38 health left.
Grant the Mighty Chicken has 8 health left.
Adam has 36 health left.
Grant the Mighty Chicken has 7 health left.
Adam has 34 health left.
Grant the Mighty Chicken has 5 health left.
Adam has 33 health left.
Grant the Mighty Chicken has 3 health left.
Adam has 31 health left.
Grant the Mighty Chicken has 2 health left.
Adam has 30 health left.
Grant the Mighty Chicken has 1 health left.
Adam has 29 health left.
```

JAVASCRIPT LAB PART 2 - FUNCTIONS

Task: Expand on the game by adding functions that allow the user to start the game and get a number to use as damage.

Functions:

Rearrange your code to include these functions.

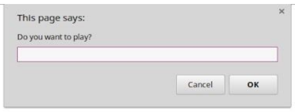
1. `startGame` function. This function will
    a. Prompt the user if they would like to play, and if they say yes…
        i. Allow the user to enter the character's name.
        ii. Call the `startCombat` function.
2. `startCombat` function. When executed, this will run the entire loop you created in Part One.
3. `getDamage` function. This will return a number between 1 and 5 that will be used to decide how much damage the user and Grant will deal each round.

Additional Changes:

1. Each iteration of the loop will include a prompt that will ask the user if they would like to "attack" or "quit".
2. If the user decides to attack, adjust the character's health points and Grant's health points based on the getDamage function.
3. If the user decides to quit, figure out a way to exit out of the loop and function.

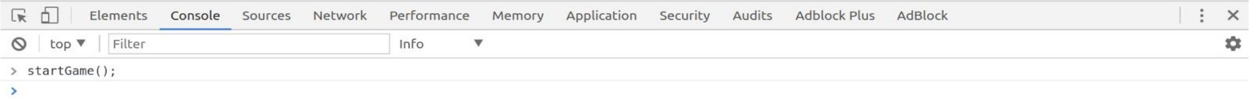
Continued on next page…

Console Preview:

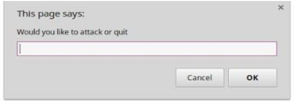


This page says:

Do you want to play?

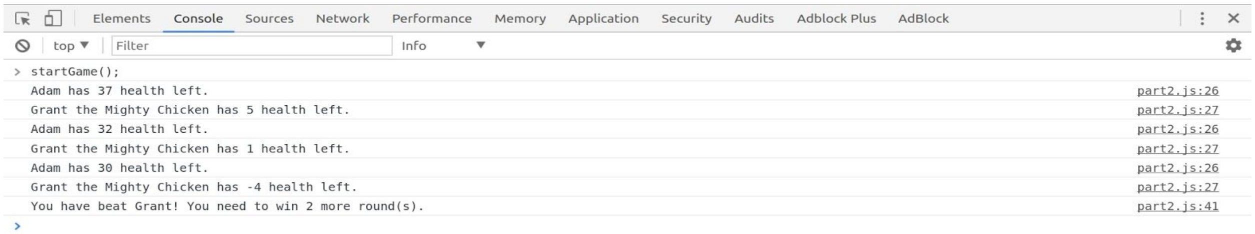


```
> startGame();
>
```



This page says:

Would you like to attack or quit



```
> startGame();
  Adam has 37 health left.                                  part2.js:26
  Grant the Mighty Chicken has 5 health left.               part2.js:27
  Adam has 32 health left.                                  part2.js:26
  Grant the Mighty Chicken has 1 health left.               part2.js:27
  Adam has 30 health left.                                  part2.js:26
  Grant the Mighty Chicken has -4 health left.              part2.js:27
  You have beat Grant! You need to win 2 more round(s).     part2.js:41
>
```

JAVASCRIPT LAB PART 3 - OBJECTS

Task: Convert the variables relating to Grant and the user's character into objects. Add an option to heal your character.

The `character` object:

Properties

- `name`: the name entered by the user via prompt at the beginning.
- `health`: how many health points remain. Starts at 40.
- `healsRemaining`: how many more times you can heal yourself. Starts at 2.
- `wins`: how many times you have taken Grant's health to 0. Starts at 0.

Methods

- `generateAttackDamage`: returns a random number between 1 and 3. This number will be used elsewhere in the program to remove health from Grant.
- `heal`: adds a random number between 1 and 10 to the character's **`health`**. Also subtracts one from **`healsRemaining`**. (hint: you should use `` `this` ``).

The `grant` object:

Properties

- `name`: the name you assign Grant's character.
- `health`: how many of Grant's health points remain. Starts at 10.

Methods

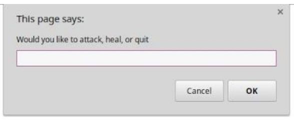
- `generateAttackDamage`: returns a random number between 1 and 3. This number will be used elsewhere in the program to remove health from the user's character.

Build Specifications:

1. Each time the user is prompted, they have the choice either to attack, heal or quit. Grant always attacks.
2. Do not allow heal if there are no heals remaining.
3. Grant still regenerates to 10 when his health hits 0. Every time this happens, add 1 to the user character's wins.
4. When the user character's wins reaches 5, Grant is defeated and the game is over.
5. When the user character's health reaches 0, the user is defeated and the game is over.
6. Continue to log health and wins at each iteration using the name entered by the user at the beginning, just as in previous parts of the lab.

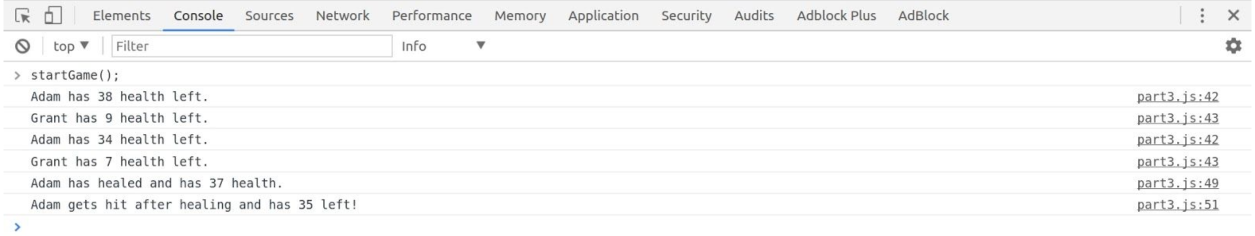
Continued on next page...

Console Preview:



This page says:

Would you like to attack, heal, or quit



```
> startGame();
  Adam has 38 health left.                                  part3.js:42
  Grant has 9 health left.                                  part3.js:43
  Adam has 34 health left.                                  part3.js:42
  Grant has 7 health left.                                  part3.js:43
  Adam has healed and has 37 health.                        part3.js:49
  Adam gets hit after healing and has 35 left!              part3.js:51
>
```

JAVASCRIPT LAB PART 4 - DOM

Task: Implement a front-end for your game based on the wireframes given to you.

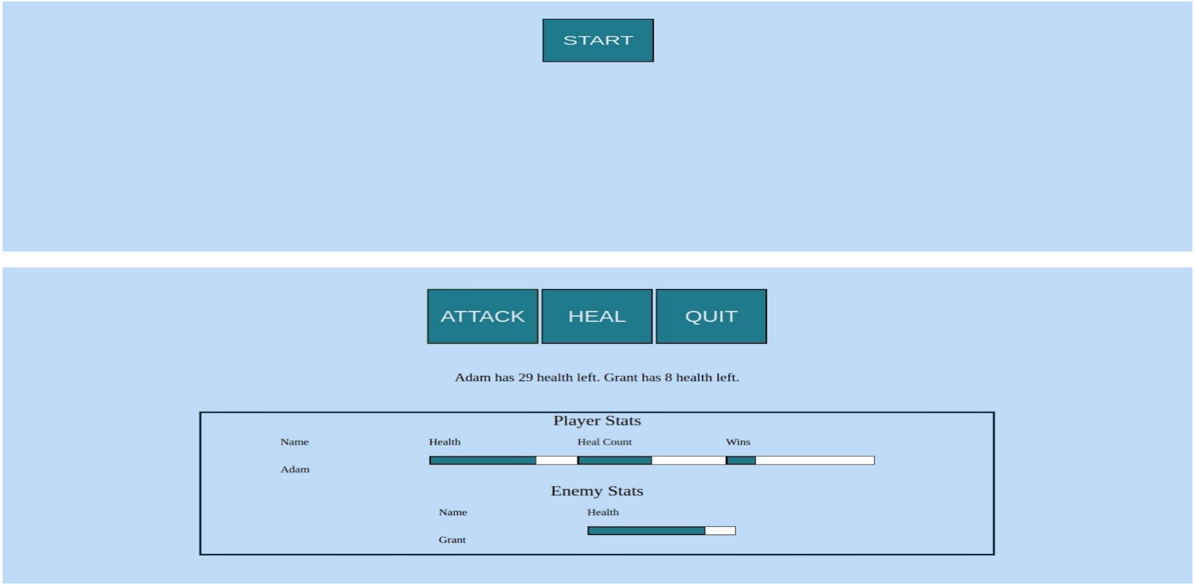
What does the application do?

1. Displays the character's name, health, heals remaining, and wins.
2. Displays Grant's name and health.
3. Allows the user to click a button to:
    a. Start the game
    b. Choose to attack
    c. Choose to heal
    d. Choose to quit
4. Each time the user selects an action, the app will display text to let the user know what has happened that round. (See line on wireframe below attack, heal, and quit buttons.)

Build Specifications:

1. The start button will execute the startGame function, which creates the character and Grant object.
2. The attack, heal, and quit buttons will execute the startCombat function with an argument describing what action they have chosen, which will no longer contain the while loop.
3. The startCombat function will execute two functions:
    a. One function will update the character and Grant's information within the DOM
    b. One function will update the text relating to what has happened during the round

Wireframe Preview:



Adam has 29 health left. Grant has 8 health left.