// gameData.js

|  |  |  |  |
| --- | --- | --- | --- |
| xp | 0 |  |  |
| health | 100 |  |  |
| gold | 50 |  |  |
| currentWeapon | 0 |  |  |
| fighting |  |  |  |
| monsterHealth |  |  |  |
| inventory | array | “stick” |  |
| weapons | array | Name:  Power: | Stick  5 |
|  |  | Name:  Power: | Dagger  30 |
|  |  | Name:  Power: | Claw hammer  50 |
|  |  | Name:  Power: | Sword  100 |

{ name: 'stick', power: 5 },

|  |  |  |  |
| --- | --- | --- | --- |
| monsters | array | Name:  Level:  Health: | Slime  2  15 |
|  |  | Name:  Level:  Health: | fanged beast  8  60 |
|  |  | Name:  Level:  Health: | dragon  20  300 |
| locations | array | Name:  "button text"  "button functions"  text: | "town square"  ["Go to store", "Go to cave", "Fight dragon"]  [],  // Leave empty or use placeholder values  You are in the town square. You see a sign that says \"Store\"." |
|  |  | Name:  "button text"  "button functions"  text: | "store",  ["Buy 10 health (10 gold)", "Buy weapon (30 gold)", "Go to town square"],  [],  // Leave empty or use placeholder values  "You enter the store." |
|  |  | Name:  "button text"  "button functions"  text: | "cave",  "Fight slime", "Fight fanged beast", "Go to town square"],  [],  // Leave empty or use placeholder values  "You enter the cave. You see some monsters." |
|  |  | Name:  "button text"  "button functions"  text: | "fight"  ["Attack", "Dodge", "Run"]  [],  // Leave empty or use placeholder values  "You are fighting a monster." |
|  |  | Name:  "button text"  "button functions"  text: | "kill monster",  ["Go to town square", "Go to town square", "Go to town square"],  [],  // Leave empty or use placeholder values  'The monster screams "Arg!" as it dies. You gain experience points and find gold.' |
|  |  | Name:  "button text"  "button functions"  text: | "lose",  ["REPLAY?", "REPLAY?", "REPLAY?"]  [],  // Leave empty or use placeholder values  "You die. &#x2620;" |
|  |  | Name:  "button text"  "button functions"  text: | "win",  ["REPLAY?", "REPLAY?", "REPLAY?"]  [],  // Leave empty or use placeholder values  "You defeat the dragon! YOU WIN THE GAME! &#x1F389;" |
|  |  | Name:  "button text"  "button functions"  text: | "easter egg",  ["2", "8", "Go to town square?"],  [],  // Leave empty or use placeholder values  "You find a secret game. Pick a number above. Ten numbers will be randomly chosen between 0 and 10. If the number you choose matches one of the random numbers, you win!" |

// gameController.js

DOM (Document Object Model)

The **Document Object Model (DOM)** is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content. The DOM represents the document as a tree structure, where each node is an object representing a part of the document.

**What is the DOM?**

1. **Representation of HTML Documents**:
   * The DOM provides a structured, object-oriented representation of a webpage, which is usually written in HTML or XML. This representation is a **tree of nodes**, where each node represents an element (like a paragraph, a button, an image, etc.), an attribute (like class, id, or style), or a piece of text in the document.
2. **Interactivity and Dynamic Changes**:
   * The DOM allows JavaScript and other scripting languages to interact with the HTML or XML content. Through the DOM, scripts can dynamically change the content, structure, and style of a webpage, making it interactive and responsive to user actions.
   * For example, using the DOM, you can:
     + Change the text inside an HTML element.
     + Modify the attributes of an element (like changing the source of an image or the value of a form input).
     + Add or remove HTML elements and attributes.
     + React to user events like clicks, mouse movements, or keyboard input.
3. **Tree Structure of the DOM**:
   * The DOM is structured like a tree, where the entire document is the root, and all elements are its children. Each element can have child elements of its own, forming a tree-like structure.
   * Here is a simplified example:

A screenshot of a computer program

Description automatically generated

1. **Node Types in the DOM**:
   * **Element Nodes**: Represent HTML tags (like <body>, <div>, <p>, <button>, etc.).
   * **Attribute Nodes**: Represent HTML attributes (like id, class, href, etc.).
   * **Text Nodes**: Represent the text content inside HTML elements.
   * **Comment Nodes**: Represent HTML comments (<!-- This is a comment -->).
2. **DOM Manipulation**:
   * Using JavaScript, you can manipulate the DOM to change the webpage dynamically. Some common DOM manipulation methods include:
     + **document.getElementById('myButton')**: Finds an element by its ID.
     + **document.querySelector('.myClass')**: Finds the first element that matches a CSS selector.
     + **element.innerHTML**: Gets or sets the HTML content inside an element.
     + **element.style**: Gets or sets the inline CSS style of an element.
     + **element.addEventListener('click', function)**: Adds an event listener to an element to listen for user interactions.

A screenshot of a medical test

Description automatically generated

|  |  |
| --- | --- |
| const button1 | const button1 = document.querySelector('#button1'); |
| const button2 | const button2 = document.querySelector("#button2"); |
| const button3 | const button3 = document.querySelector("#button3"); |
| const text | const text = document.querySelector("#text"); |
| const xpText | const xpText = document.querySelector("#xpText"); |
| const healthText | const healthText = document.querySelector("#healthText"); |
| const goldText | const goldText = document.querySelector("#goldText"); |
| const monsterStats | const monsterStats = document.querySelector("#monsterStats"); |
| const monsterName | const monsterName = document.querySelector("#monsterName"); |
| const monsterHealthText | const monsterHealthText = document.querySelector("#monsterHealth"); |

 monsterStats.style.display = "none"; //is used to hide an HTML element from the webpage.

text.innerHTML = location.text; is used to set the HTML content of an HTML element (text) to a specific string value stored in the location object.

/ Function Definitions – give each button a text and a function

function update(location) {

  monsterStats.style.display = "none";

  button1.innerText = location["button text"][0];

  button2.innerText = location["button text"][1];

  button3.innerText = location["button text"][2];

  button1.onclick = location["button functions"][0];

  button2.onclick = location["button functions"][1];

  button3.onclick = location["button functions"][2];

  text.innerHTML = location.text;