RPG Game

Array Methods - Extra

1. **Group Characters by Level**
    - **Input:** An array of character objects
    - **Task:** Write a function that returns an object where keys are unique levels and values are arrays of characters at that level.
    - **Example:**

```
const characters = [
    { name: 'John', level: 2 },
    { name: 'Jane', level: 5 },
    { name: 'Doe', level: 2 }
];
groupCharactersByLevel(characters); // { '2': [{ name: 'John', level:
2 }, { name: 'Doe', level: 2 }], '5': [{ name: 'Jane', level: 5 }] }
```

2. **Find Highest Level Character**
    - **Input:** An array of character objects
    - **Task:** Write a function that returns the character object with the highest level.
    - **Example:**

```
const characters = [
    { name: 'John', level: 2 },
    { name: 'Jane', level: 5 }
];
findHighestLevelCharacter(characters); // { name: 'Jane', level: 5 }
```

3. **Calculate Average Character Level**
    - **Input:** An array of character objects
    - **Task:** Write a function that returns the average level of all characters, rounded to the nearest whole number.
    - **Example:**

```
const characters = [
    { name: 'John', level: 2 },
    { name: 'Jane', level: 5 }
];
calculateAverageCharacterLevel(characters); // 4
```

4. **Filter Characters by Level Range**

   - **Input:** An array of character objects, two numbers representing the lower and upper bounds of a level range
   - **Task:** Write a function that returns an array of character objects which have a level within the provided range (inclusive).
   - **Example:**

```
const characters = [
    { name: 'John', level: 2 },
    { name: 'Jane', level: 5 }
];
filterCharactersByLevelRange(characters, 1, 3); // [{ name: 'John',
level: 2 }]
```

5. **Count Number of Characters per Level**

   - **Input:** An array of character objects
   - **Task:** Write a function that returns an object where keys are levels and values are the number of characters at that level.
   - **Example:**

```
const characters = [
    { name: 'John', level: 2 },
    { name: 'Jane', level: 5 },
    { name: 'Doe', level: 2 }
];
countCharactersPerLevel(characters); // { '2': 2, '5': 1 }
```

6. **Find Character with Most Skills**

   - **Input:** An array of character objects, each with a skills property (array of strings)
   - **Task:** Write a function that returns the character object with the most skills.

- **Example:**

```
const characters = [
    { name: 'John', level: 2, skills: ['fire', 'ice'] },
    { name: 'Jane', level: 5, skills: ['fire', 'ice', 'wind'] }
];
findCharacterWithMostSkills(characters); // { name: 'Jane', level: 5,
skills: ['fire', 'ice', 'wind'] }
```

7. **Count Total Number of Skills**

- **Input:** An array of character objects, each with a skills property (array of strings)
- **Task:** Write a function that returns the total number of skills across all characters.
- **Example:**

```
const characters = [
    { name: 'John', level: 2, skills: ['fire', 'ice'] },
    { name: 'Jane', level: 5, skills: ['fire', 'ice', 'wind'] }
];
countTotalNumberOfSkills(characters); // 5
```

8. **Find Unique Skills**

- **Input:** An array of character objects, each with a skills property (array of strings)
- **Task:** Write a function that returns a new array with all unique skills from the characters.
- **Example:**

```
const characters = [
    { name: 'John', level: 2, skills: ['fire', 'ice'] },
    { name: 'Jane', level: 5, skills: ['fire', 'ice', 'wind'] }
];
findUniqueSkills(characters); // ['fire', 'ice', 'wind']
```

9. **Find Character with Specific Skill**

- **Input:** An array of character objects, each with a skills property (array of strings), and a skill string
- **Task:** Write a function that returns an array of character objects that possess the specified skill.

- **Example:**

```
const characters = [
    { name: 'John', level: 2, skills: ['fire', 'ice'] },
    { name: 'Jane', level: 5, skills: ['fire', 'ice', 'wind'] }
];
findCharactersWithSkill(characters, 'wind'); // [{ name: 'Jane',
level: 5, skills: ['fire', 'ice', 'wind'] }]
```