ESMAD | TSIW | POO
Exercise Sheet nº 8
Classes

Open Visual Studio Code and solve the following exercises:

1. **App using classes (Escape Rooms)**

   An escape room, or riddle room, is a game in which a player discovers clues, solves puzzles and accomplishes tasks in one or more rooms to accomplish a specific goal.

   a. Create an application that can have several escape rooms with various puzzles each. For simplicity, puzzles will be questions with only a correct answer from four choices. You must create two classes named "EscapeRoom" and "Puzzle". Preferably, create each class in its own JavaScript file and code the main application in a third one. You should export each class and import both in the main file; the main file must be imported with the type of module in the HTML so this can work. For more info on this approach, check the following:
      - Developer Mozilla - JavaScript: import
      - Developer Mozilla - JavaScript: export
      - Developer Mozilla - JavaScript Modules

   b. An **escape room class** has the following properties (add the convenient getters and setters):
      a. A **name**.
      b. An **image** that is supplied as an url.
      c. A collection of **puzzles**.
      d. A counter for the puzzles completed.

   c. The escape room should also have two methods. One to solve a puzzle (mark a puzzle as solved), and another to check if all puzzles in the escape room have been saved.

   d. A **puzzle class** has the following properties (add the convenient getters and setters):
      a. A **name**.
      b. The **difficulty** level: easy, medium, or hard.
      c. A **solved** property, indicating if it has been solved.
      d. The **text** of the question.

   e. A collection of **answers**.
   f. The number of the **solution** (value related to the index of the collection)

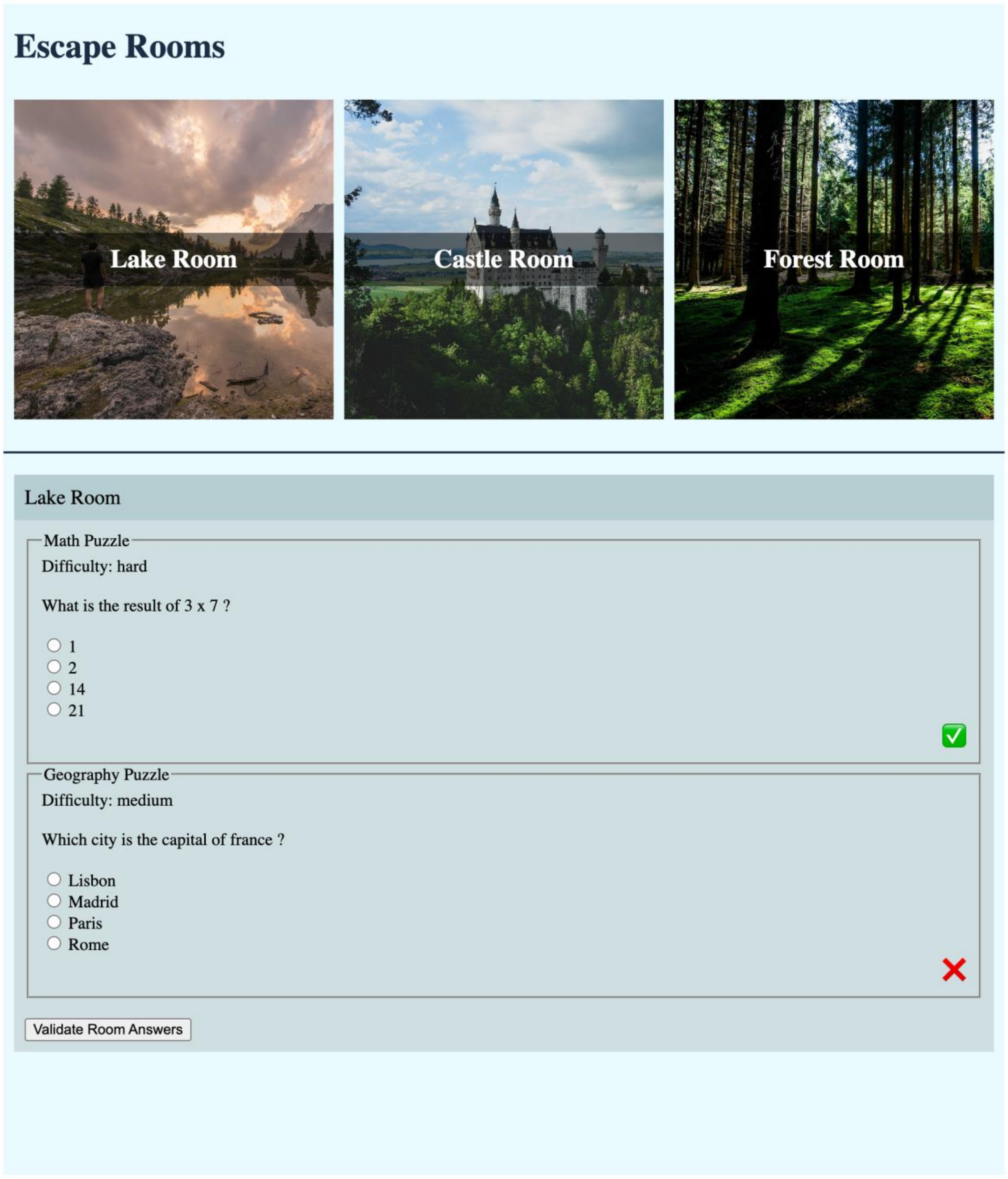
e. Define a method to solve the puzzle, this should also log the info that you have solved the puzzle.
f. For the graphic interface you should use the available HTML na CSS code on the GitHub repository: ES08/base-material.
g. As an example, create 3 rooms and 4 questions:

```
const puzzle1 = new Puzzle('Math Puzzle', 'hard', 'What is the result of 3 x 7', [1, 2, 14, 21], 4);
const puzzle2 = new Puzzle('Geography Puzzle', 'medium', 'Which city is the capital of france', ['Lisbon', 'Madrid', 'Paris', 'Rome'], 3);
const puzzle3 = new Puzzle('Quiz Puzzle', 'easy', 'Which animal is the Mickey Mouse', ['Cat', 'Dog', 'Mouse', 'Rabbit'], 3);
const puzzle4 = new Puzzle('One Letter Puzzle', 'hard', 'Which letter denotes the roman numeral for 100', ['C', 'X', 'V', 'I'], 1);

const escapeRoom1 = new EscapeRoom('Lake Room', 'https://picsum.photos/id/980/5000/3509');
escapeRoom1.addPuzzle(puzzle1);
escapeRoom1.addPuzzle(puzzle2);
const escapeRoom2 = new EscapeRoom('Castle Room', 'https://picsum.photos/id/1040/4496/3000');
escapeRoom2.addPuzzle(puzzle3);
const escapeRoom3 = new EscapeRoom('Forest Room', 'https://picsum.photos/id/502/1920/1280');
escapeRoom3.addPuzzle(puzzle4);
```

   a. A new puzzle receives in its constructor a name, the difficulty, the text with the question, an array of choices and the number of valid answers.
   b. An escape room receives in its constructor a name, and an image.

h. Render all the entities in the given HTML.
i. By clicking a Room in the GUI, the corresponding puzzles must be rendered in the box below.
j. You should have a submit button that validates the puzzles in each escape room. This should update the icons of completion in the GUI (use the chars &#9989; and &#10060;)



Escape Rooms

Lake Room

Castle Room

Forest Room

Lake Room

Math Puzzle

Difficulty: hard

What is the result of 3 x 7 ?

- ○ 1
- ○ 2
- ○ 14
- ○ 21

✅

Geography Puzzle

Difficulty: medium

Which city is the capital of france ?

- ○ Lisbon
- ○ Madrid
- ○ Paris
- ○ Rome

❌