**Mighty text adventure**

As you open the rusty door you feel a warm breeze and a strong stench. Your dragon sense 2000 ring begins to glow, and you know that this will be the ultimate battle...

Have you ever played an old text-based adventure game? You don't just read them as a story (as with any other book), but you can also influence the environment with text commands. A story can have several outcomes and your choices determine which one you reach.

You will make our own adventure this weekend.

**Tasks**

Plan your game and sketch your story.

1. There is a main character to walk through your game. The character and the starting context are described to the user.
2. The user can name the character.
3. The game has a base storyline, separated into smaller stages. This is the plan of what can happen to the main character.

Create a data structure that to store the details of an area.

1. The description of the area is stored.
2. One or more actions that the player can do in the area are stored.
3. One or more connected areas where each action leads are stored.

The game has a well-defined starting and ending point.

1. The game has a starting point.
2. There is at least one goal for the game. The game checks in every turn whether the player reached the goal.

**Game logic**

Make the game playable. Have a class that handles the logic.

1. The user can move the Player in between areas, based on the selected action.
2. A message is displayed when the player wins the game.
3. A message is displayed when the player loses the game.

**Player**

Define a Player class that stores the necessary properties.

1. The name of the player is stored.
2. The current area where the player is located is stored.
3. The items picked up during the adventure are stored.

**Player actions**

Add multiple actions to the game, such as Go North, Go South, Pick up teapot.

1. There is a class that describes the Action that can be performed at an Area.
2. The descriptions for the actions are stored.
3. Both long and short forms of actions are accepted, such as 'N' or 'North'.

**Display**

Create a separate class which handles all output to the console.

1. All UI display tasks are organized separately from the logic.

**Input handling**

Create a class that handles user input.

1. The user input handler is case insensitive.
2. If h or help is typed into the console, information about the game controls is displayed.

**OPTIONAL: Persistence**

Implement a database or a file-based storage where all the areas, actions and player properties are stored.

1. Retrieve the game data from this data source
2. Player statistics are updated on the spot
3. The game can be “saved” and “resumed” based on the data stored in the database or file

**Hints**

* Draw the steps of the storyline on paper or in a digital notebook. This provides a detailed plan of the possibilities in the game in a diagram format.
* Get inspiration from [this](https://static.journey.study/media/mighty-text-adventure-example1.png) and [this](https://static.journey.study/media/mighty-text-adventure-example2.pdf) example story.

**Background materials**

* [Interactive fiction](https://en.wikipedia.org/wiki/Interactive_fiction)