Labyrinth

A text-based adventure

Objective: create a text-based adventure game engine

- I/O `print` and `input`
- Comments
- Strings and string formatting
- Numeric data types
- Conversion to/from strings/numbers
- Date data types
- Error handling
- Conditional logic (`if` and `while`)
- Collections
- Loops
- Functions
- Modules and packages
- API calls
- JavaScript Object Notation (JSON)

You don’t need to use all of these.

But, you cannot use something that is not on this list, `classes` for example.

Playing the game

The game is played within a maze or Labyrinth.

When the game starts - the Player is given a starting position.

The aim of the game is simple - get out alive!

The player moves from one position to another within the maze.

There are perils along the way in the form of Minotaurs - most encounters are fatal, but you might just get lucky!

The game ends when either the player is killed or they escape from an exit - alive.

The Player ⇒

⇐ A Minotaur

A maze is a grid

The maze is simply an x, y indexed grid.

At each position, the Player can only move in one of four directions: North, South, East or West (NSEW).

You should inform the player which directions are available at each move so they know what moves are available to them.

The game definition is written in JSON and is available via an API.

This describes the maze, the available moves, the type of position (entrance or exit), the Player's health status and some text describing the action.

Here is a brief example of the JSON API Schema ⇒

```
{
  "name": "Game name",
  "description": "Game desc",
  "maze": {
    "1:0": {
      "entrance": true,
      "exit": false,
      "alive": true,
      "description": "Node desc",
      "nodes": ["1:1"]
    },
    // Additional nodes go here!
  }
}
```

Moving around the grid

The **key** to each position (or node) is a String (e.g. "1:0") - this describes the grid (**x**, **y**) position of the node within the maze.

x is a horizontal movement - **y** is a vertical movement. **Important:** horizontal, then vertical.

This means you'll need to convert between "1:0" and array of integers (e.g. [1, 0]) in order to move around the maze.

However, we don't want the player to have to know about x, y nature of the grid. We'd like them to simply move in a direction - North, South, East or West.

Here are the rules for converting between a NSEW move and the new grid reference ⇒

A move **North**, *increases* the **y** value by 1
A move **South**, *decreases* the **y** value by 1
A move **East**, *increases* the **x** value by 1
A move **West**, *decreases* the **x** value by 1

So, given a starting point of **1:1**...

A move **North** the player is now at **1:2**
A move **South** the player is now at **1:0**
A move **East** the player is now at **2:1**
A move **West** the player is now at **0:1**

An example (JSON) game definition…

https://7w298.wiremockapi.cloud/labyrinth

IMPORTANT

Your game engine should be able to take any game definition JSON file - as long as it is in the schema described.

Your game engine code should **NOT** need to change when switching games.

The Labyrinth