

## Practice Problem for Section 8

### A Little Adventure

To practice with the Visitor pattern and OO/functional decomposition a bit, we'll do something similar to the homework problem.

You're given a Standard ML program implementing a small bit of a very simple role-playing engine. This is not an actual game, just something that pits a character (either a knight or a wizard) against a series of challenges. Since knights and wizards have very different approaches to solving the problems facing them, we need to implement small pieces of logic changing the state of the game depending on the combination of character's type and the type of the challenge. Sounds like a perfect match for the Visitor pattern!

Your task is to inspect the provided Standard ML definitions, and reimplement the same logic in Ruby using a principled object-oriented approach. A template file for your solution is also provided, as well as some code responsible for advancing the overall world state.

Note that the double dynamic dispatch is asymmetric in this case.