

# Game simulation

This exercise pertains to the game Chutes and Ladders.

Using the programming language of your choice, implement a simulation of the game mechanics based on the game's rules. The game simulation should use three players and two dice (one 6-side, one 4-sided, rolled together producing a roll from 2 to 10) as an example, but should allow for any number of players to play with any number of dice. The board definition used in the simulation should be provided as a CSV file at runtime. In this packet, a picture of a board (board1.webp) and its associated CSV descriptor file (board1.csv) are included. Two other CSV files for other boards (board2.csv and board3.csv) are available as alternative inputs for the game. At the end of the game simulation, a winner should be identified. The dice used in the game must strictly follow the provided interface, which delivers random rolls of the dice (presented here in Python):

---

```
class Rollable:
    @abstractmethod
    def roll(self) -> int:
        pass
```

---

The solution should implement a Game class using a constructor with the following signature:

---

```
class Game:
    def __init__(self, players: List[Player], rollable: Rollable) -> None:
        # constructor code
```

---

## Rules of the game:

- Move one square for each number rolled on the Rollable.
- If a player stops at the bottom of a ladder, they move up to the top of the ladder.
- If a player lands at the top of a chute or snake's head, they move down to the bottom of the chute or snake's tail.
- Two or more players may stop on the same square at the same time.
- Square 100 must be reached through an exact roll, or the Rollable must be rolled again.

## Submission:

Submit a response as a zip file containing all source code and any other supporting code or documents.