**Builder Pattern** separates the construction of a complex object from its representation so that the same construction process can create different representations.
HOW IT WORKS?
1. Client creates builder
2. Client creates director
3. Client starts the director
4. Director instructs builder
5. Client takes the product from builder
EXAMPLE
An open area. There are exits in all directions.
A mostly open area. The way north is blocked.
HMMMM ...
Does **StringBuilder** use the builder pattern?
WHEN TO USE?
- The algorithm for creating a complex object should be independent of the parts that make up the object and how they’re assembled.

- The construction process must allow different representations for the object that is being constructed.

- It should be easy to add new representations without modifying the algorithm.
TODO

- Read the Builder chapter in CGLearn
- Do the Builder task in CGLearn IN TEAMS (your final task)