

31.1 Complexity II

Complexity comshmexity.

In the previous Software Engineering lecture, we introduced the concept of complexity.

We define complexity as: anything related to the structure of a software system that makes it hard to understand and modify the system.

Project 3: Build Your own World provides the first opportunity to create complex code. We do not explicitly define what constitutes a room and what constitutes a hallway. There is a great deal of freedom for not just the implementation but the entire architecture of the project. As a result, it is of utmost importance to understand how to manage the complexity.

In this chapter, we will use examples from Project 3: BYoW to illustrate cases of poorly managed complexity. Consider the following code snippet:

```
if (move.equals("a")
    && !world[player.xxcenter - 1][player.yycenter].equals(Tileset.WALL)) {
    player.xxcenter -= 1;
    world[player.xxcenter][player.yycenter] = Tileset.PLAYER;
    world[player.xxcenter + 1][player.yycenter] = Tileset.FLOOR;
    steps += 1;
}
if (move.equals("s") &&
    !world[player.xxcenter][player.yycenter - 1].equals(Tileset.WALL)) {
    player.yycenter -= 1;
    world[player.xxcenter][player.yycenter] = Tileset.PLAYER;
    world[player.xxcenter][player.yycenter + 1] = Tileset.FLOOR;
    steps += 1;
}
if (move.equals("d")
    && !world[player.xxcenter + 1][player.yycenter].equals(Tileset.WALL)) {
    player.xxcenter += 1;
    world[player.xxcenter][player.yycenter] = Tileset.PLAYER;
    world[player.xxcenter - 1][player.yycenter] = Tileset.FLOOR;
```

There are multiple complexity issues with this code:

1. It lacks readability. The code itself is not narrative. It is not immediately clear to someone looking at the code for the first time to figure out what it does.
2. There are many complex manual computations of moving in each direction. The details of these should be hidden away.
3. The code is very repetitive. It will be difficult to make changes because the programmer will need to make certain that all of the updates have been fixed.

Ultimately, there is no *right* answer for what the best practice is for coding in this movement functionality. One could argue that this code could benefit from having helper methods to check if a move in a certain direction is valid and helper methods to do the updates. However, it could also be argued that this code is simple enough to not have any helper methods. It is up to the programmer to decide what would be easiest to write and maintain.

[Previous](#)
[31. Software Engineering II](#)

[Next](#)
[31.2 Sources of Complexity](#)

Last updated 1 year ago

