

31.2 Sources of Complexity

There are two primary sources of complexity:

- Dependencies: when a piece of code cannot be read, understood, or modified independently.
- Obscurity: when important information is not obvious.

In the last section, we showed a code snippet from Project 3 BYoW:

```
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;
```

The complex manual computation of whether a move is valid is an example of obscurity. In order to understand the code, we would also need to understand things about the structure of the `world` array. We also do not know if `player.yycenter+1` is for moving in the up direction or down direction. The repetitive code is an example of dependencies. The updates cannot be modified independently--they all have to be modified in conjunction, or else the behavior is incorrect.

In order to create good systems that last, it is best to keep dependencies and obscurity at a minimum.

Tactical Programming

Tactical programming describes a linear approach to writing code, where functions are written only when they are needed. In other words, just trying to get something working instead of first thinking about the overall architecture. It is impractical to use this style of programming when writing anything larger than a small script.

[Previous](#)
[31.1 Complexity II](#)

[Next](#)
[31.3 Modular Design](#)

Last updated 1 year ago

