

# Assignment 3 Report

## Shadi Zoldjalali and Jeffrey Ramacula

### Code Smell: Duplicate Code

CR: [https://csil-git1.cs.surrey.sfu.ca/cmpt276f22\\_group17/project/-/merge\\_requests/55](https://csil-git1.cs.surrey.sfu.ca/cmpt276f22_group17/project/-/merge_requests/55)

#### Description:

We found that in the panel class, there was a second constructor that did the same exact thing as the main constructor but turned off the sound for testing using a given noSound boolean flag. We decided that this was redundant, and combining into one constructor with an optional flag was more clear, concise, and efficient.

### Code Smell: Bad/Confusing Variable Names

CR: [https://csil-git1.cs.surrey.sfu.ca/cmpt276f22\\_group17/project/-/merge\\_requests/56](https://csil-git1.cs.surrey.sfu.ca/cmpt276f22_group17/project/-/merge_requests/56)

**Description:** We found that *kh* was not a clear variable name, so we refactored it to *keyHandler*. Similar to *keyHandler*, *tm* was not clear, so it was refactored to *tileMap*.

### Code Smell: Data clumps

CR: [https://csil-git1.cs.surrey.sfu.ca/cmpt276f22\\_group17/project/-/merge\\_requests/57](https://csil-git1.cs.surrey.sfu.ca/cmpt276f22_group17/project/-/merge_requests/57)

#### Description:

We found that in the Game class, there was a startCol, startRow (that was meant for finding where the cage/starting position of the monkey is) variable and an exitCol, exitRow (that was meant for finding where the exit/door position is) variable. Meanwhile, we have a position class to encapsulate these variables that is meant for this exact use case. So by removing this data clump in the game class, it is clearer and more concise to access the positions of the cage and door.

### Code Smell: Duplicate Code

CR: [https://csil-git1.cs.surrey.sfu.ca/cmpt276f22\\_group17/project/-/merge\\_requests/59](https://csil-git1.cs.surrey.sfu.ca/cmpt276f22_group17/project/-/merge_requests/59)

#### Description:

We found that in the Panel class, there was duplicate code regarding rows, cols, tileSize, width, and height. So instead of having this duplicate code in two places, we kept it only in the game class and had other classes reference the information from the game class rather than the panel class.

### Code Smell: Poorly Structured Code

CR: [https://csil-git1.cs.surrey.sfu.ca/cmpt276f22\\_group17/project/-/merge\\_requests/58](https://csil-git1.cs.surrey.sfu.ca/cmpt276f22_group17/project/-/merge_requests/58)

**Description:** We found that the nested while loops in the *drawMap()* method were unclear, so we changed them into nested for loops. The outer for loop iterates over the

rows, while the inner for loop iterates over the columns. This is clearer to read and better for possible modification.

**Code Smell: Classes that are too large and/or try to do too much**

**CR:** [https://csil-git1.cs.surrey.sfu.ca/cmpt276f22\\_group17/project/-/merge\\_requests/60](https://csil-git1.cs.surrey.sfu.ca/cmpt276f22_group17/project/-/merge_requests/60)

**Description:** The Game class had many spawn methods with similar functionality, so we extracted those methods and encapsulated them into a Spawner class. This made the Game class smaller as the responsibility of spawning was moved into another class.

**Code Smell: Methods that are too long and that could benefit from being refactored**

**CR:** [https://csil-git1.cs.surrey.sfu.ca/cmpt276f22\\_group17/project/-/merge\\_requests/61](https://csil-git1.cs.surrey.sfu.ca/cmpt276f22_group17/project/-/merge_requests/61)

**Description:** We found that the update method in the Monkey class was too long and was doing too much (had semi-scattered code). Thus, we decided to extract a method *checkLionPit* for handling logic between lion pit and monkey interaction. The code is now clearer and bugs can be found more easily.