

1. **Code Duplication** on part of several method in UserInterface.java

Details: in drawWinScreen() and drawDeathScreen(), part of the codes are sharing the exact same code

Solution: create a new method holding the same code

```
private void drawSprite(String imgDirectory, int x, int y, int width, int height) {
    spriteCount++;
    if(spriteCount > 10) {
        if(spriteNum == 1) {
            spriteNum = 2;
        }else if(spriteNum == 2) {
            spriteNum = 3;
        }else if(spriteNum == 3) {
            spriteNum = 4;
        }else if(spriteNum == 4) {
            spriteNum = 5;
        }else if(spriteNum == 5) {
            spriteNum = 1;
        }
    }

    spriteCount = 0;
}

if(spriteNum==1){
    g2.drawImage(setup(imgDirectory+"0"), x, y, gp.tileSize*2, gp.tileSize*2, observer:null);
}
if(spriteNum==2){
    g2.drawImage(setup(imgDirectory+"1"), x, y, gp.tileSize*2, gp.tileSize*2, observer:null);
}
if(spriteNum==3){
    g2.drawImage(setup(imgDirectory+"2"), x, y, gp.tileSize*2, gp.tileSize*2, observer:null);
}
if(spriteNum==4){
    g2.drawImage(setup(imgDirectory+"3"), x, y, gp.tileSize*2, gp.tileSize*2, observer:null);
}
if(spriteNum==5){
    g2.drawImage(setup(imgDirectory+"2"), x, y, gp.tileSize*2, gp.tileSize*2, observer:null);
}
if(spriteNum==6){
    g2.drawImage(setup(imgDirectory+"1"), x, y, gp.tileSize*2, gp.tileSize*2, observer:null);
}
```

Commit: Refactoring for duplicate code (8f9774e8df940a8ff57e31f3b395e0fdd14c3aec) and Debug drawSprite method (f294283508028e71673441fead595c032b9dfa8f)

2. **Code Duplication** on 'gp.tileSize'

Details: It is being called 67 times throughout UserInterface.java, which might reduce efficiency due to lots of memory access.

Solution: create one variable that does memory access, and reused the variable that is just being created. (E.g. : int tileSize = gp.tileSize;)

Commit: Refactoring for duplicate variable (730fbb0ea2f2e310c245e9cb4a7df2155d218645)

3. **Poorly Structured Code** on most of the classes in object folder

Details: Lack of error handling when object image is not found.

```
try {
    img = ImageIO.read(getClass().getResourceAsStream(name="/objects/reward.png"));
    tools.scaleImg(img, gp.tileSize, gp.tileSize);
} catch(IOException e) {
    e.printStackTrace();
}
```

Solution: Can be handled by adding exception throwing for image that failed to load

```
try {
    img = ImageIO.read(getClass().getResourceAsStream(name="/objects/reward.png"));
    if (img == null) {
        throw new IOException(message:"Failed to load reward image");
    }
    tools.scaleImg(img, gp.tileSize, gp.tileSize);
} catch(IOException e) {
    e.printStackTrace();
}
```

Commit: Add error handling when image not found
(887c1112ba5aba3110e228b54a32fa5d3f8faee2)

4. **Poorly Structured Code** on Sound.java

Details: Need a close() method for the Clip to completely close the music sound. Depends solely on the stop() method is not enough, because it will only pause the video and keep waiting for instruction to resume again.

Solution: Create a close() method in Sound.java

Commit: Add close method to the Sound (d42e8624aa39bceb21212584939c8999f420808f)