Qijun Zhou's Finding

Code Duplication

1. In the Sound.java. Instead of initializing the "soundURL" array in a static way and it contains the same code to load sounds' files with static array size, we can create a separate method named initializeSoundURL() that initializes the array using a loop and an array of file names. In this way, the array becomes dynamic so that we do not lose a lot of space on static array size. Therefore, if we need to add more music files in the future, we can easily modify the array of file names without changing the code.

   Related Commit: @dbe72520
   https://csil-git1.cs.surrey.sfu.ca/cmpt276s23_group21/group21/-/commit/dbe725206206881497731607 4a823c87d86d59f1

Dead Code/Unused Comment

1. In the line 20, and line 21 of class Sound.java, there are unused codes as comments exist in the constructor. Those comments are not meaningful so they should be removed.

   Related Commit: @dbe72520
   https://csil-git1.cs.surrey.sfu.ca/cmpt276s23_group21/group21/-/commit/dbe725206206881497731607 4a823c87d86d59f1
   //NOTE: The above two issues(Code Duplication, Dead Code/Unused Comment) were committed together, so they share the link.

2. Line 3 in UI.java, there is an unused header file that was commented out, it is supposed to be removed.
   Related Commit: @9285cdcd
   https://csil-git1.cs.surrey.sfu.ca/cmpt276s23_group21/group21/-/commit/9285cdcdc592c23706b3b5eb57af4a787c1f1dc6

Error handling

1. In the setFile() method in Sound.java, if an exception occurs while creating the clip object, it will be null, but it is not checked before calling the start() method. This can cause a NullPointerException. You should consider checking if the clip object is not null before calling its methods.

   Related Commit: @1ffb1d04

https://csil-git1.cs.surrey.sfu.ca/cmpt276s23_group21/group21/-/commit/1ffb1d0428c9a3dfcbdfd759b9a9f9ed0d96e11d

Encapsulation

1.

In variables of class in Sound.java and UI.java. It is very dangerous if we do not make the variables to protected or private. There is a risk of being accessed by other classes. We can declare the class variables as protected or private to enforce encapsulation. This means that the variables can only be accessed within the class, and any modification or access from outside the class should be done through public methods.

Related Commit: @5085bd55
https://csil-git1.cs.surrey.sfu.ca/cmpt276s23_group21/group21/-/commit/5085bd55ef5bcb44ca8e119dcf83f16f31c17fe3

Resource Management

1.

The clip object is created in the setFile() method, but it is not explicitly closed in the Sound class. It is always a good practice to close the resources as soon as they are no longer needed.

Related Commit: @9285cdcd
https://csil-git1.cs.surrey.sfu.ca/cmpt276s23_group21/group21/-/commit/9285cdcdc592c23706b3b5eb57af4a787c1f1dc6
//NOTE: this link of commit also contain some part of “Dead Code/Unused Comment”

Bad/Confusing Variable Names

1.

If we look at the code for a long time, we may be confused by the words times and timer, which look alike, leading to the wrong choice of the variable. Now, these two variables are given clearer names and are easier to identify. At the same time, the variable retro is also changed. If another developer comes, he may not know what this variable does, so he is given a clearer name so that other people in the team can identify it more easily when editing this class. is a variable of the Font.

Related Commit: @03baea74
https://csil-git1.cs.surrey.sfu.ca/cmpt276s23_group21/group21/-/commit/03baea7410e3db3780df2a9486408acebf2a0f4c