

Bug ID S-10: Unable to change elapsedTime with new GUI and simple graph

Table of contents

| | |
|---------------------------|---|
| 1 Problem..... | 2 |
| 2 Solution..... | 2 |
| 3 Modified file list..... | 2 |
| 4 Code explanation..... | 2 |
| 5 JUnit test..... | 3 |

1. Problem

In simBio_private, when somebody resets the elapsedTime value to 0 after running a simulation, an error occurs. When they try to restart the simulation, the Exception below is thrown.

```
java.lang.NoSuchMethodError: org.simBio.sim.analyzer.graph.Viewer:
    method resetBuffer()V not found
```

This problem occurs when the org.simBio.sim.analyzer.graph package is used in the xml model, but not when org.simBio.sim.analyzer.graph.simple is used.

2. Solution

If a model uses a Viewer class which supports redrawing, the resetbuffer method, which is a method of VisualizeAnalyzer, should be called before calculation. Now the condition if the resetbuffer method is called is that the object is an instance of Viewer class. So change the condition from Viewer class to VisualizeAnalyzer class.

3. Modified file list

- org.simBio.sim.gui.GUI

4. Code explanation

The bug occurs because there are two different versions of the Viewer class with same package name; one of them is in the simBio project and the other one is in the simBio_private project. In simBio_private, the Viewer class does not support redrawing, so it does not use the resetBuffer() method. The Viewer class in simBio_private belongs to the org.simBio.sim.analyzer.graph package, which, even though it has the same name as a simBio package which supports redrawing, really matches the org.simBio.sim.analyzer.graph.simple package in simBio, which does not support redrawing. If this code in GUI.java

```
if(child instanceof Viewer) {
    ((Viewer) child).resetBuffer();
}
```

is changed into this

```
if ( child instanceof VisualizeAnalyzer ) {
    ((VisualizeAnalyzer) child).resetBuffer();
}
```

then the Viewer object in the simBio_private project will not try to call resetBuffer() (since Viewer does not inherit from VisualizeAnalyzer), the error does not occur, and the program runs normally. In the simBio project, the original code worked because in that project org.simBio.sim.analyzer.graph.Viewer inherits from VisualizeAnalyzer and so can call the resetBuffer() method. In simBio_private, however,

`org.simBio.sim.analyzer.graph.Viewer` is the same as the `simBio` project's `import org.simBio.sim.analyzer.graph.simple.Viewer`, and so is unable to call the `resetBuffer()` method.

The above code is called only if the flag "modelBufferClear" is true. The `modelBufferClear` variable is initially set to false, so the code is not called the first time the simulation is run. If the `elapsedTime` is changed after running the simulation, then `modelBufferClear` is set to true. Then this code is called when the simulation is run for the second time, and `modelBufferClear` is reset to false afterwards.

5. JUnit test

The test class

- `org.simBio.sim.analyzer.gui.GUITest`

The test method

| Method | File Used | Test |
|---------------------------------|--|---|
| <code>testDndActionStart</code> | <code>src/test/resources/org/simBio/sim</code> | This calls the <code>actionPerformed</code> method of <code>dndActionStart</code> with <code>modelBufferClear</code> set to true. |

`DnDAction dndActionStart` is a private field of the GUI class. In `GUITest`, `dndActionStart` is created using reflection and then its `actionPerformed` method is called with `modelBufferClear` set to true. The `actionPerformed` method checks the value of the `modelBufferClear` flag, and if it is true then the `resetBuffer` method is called.