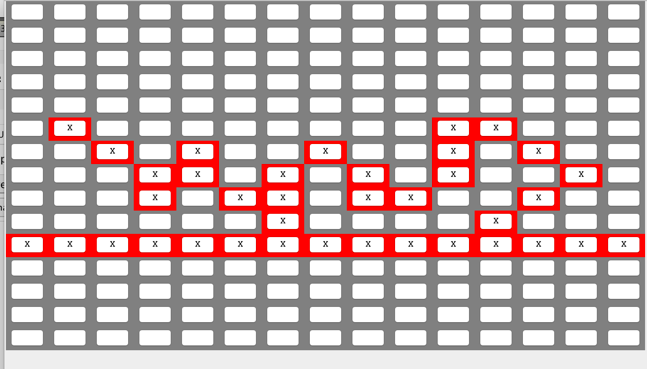
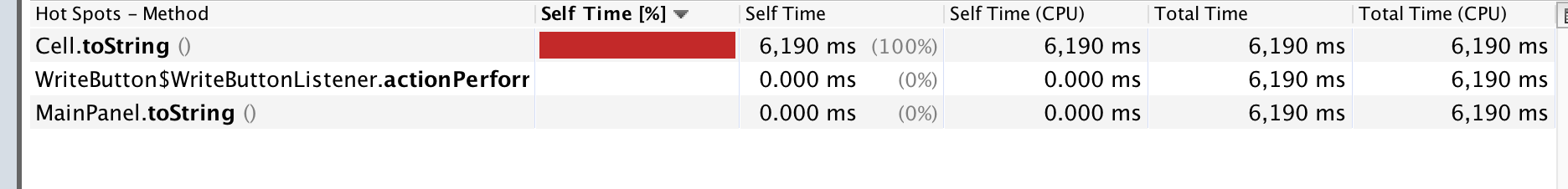
**Summary:**

Using Visual VM as profiling tool. First I click some cells shown below.

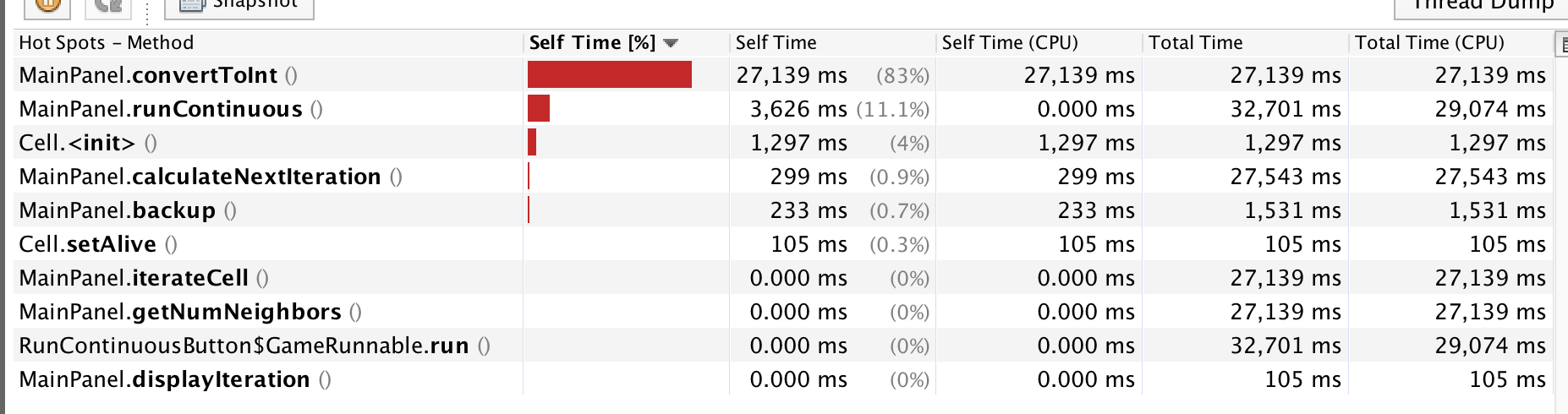


I clicked the “Write” while VM was running, then I noticed that the toString() method in Cell class costs relatively too much time.

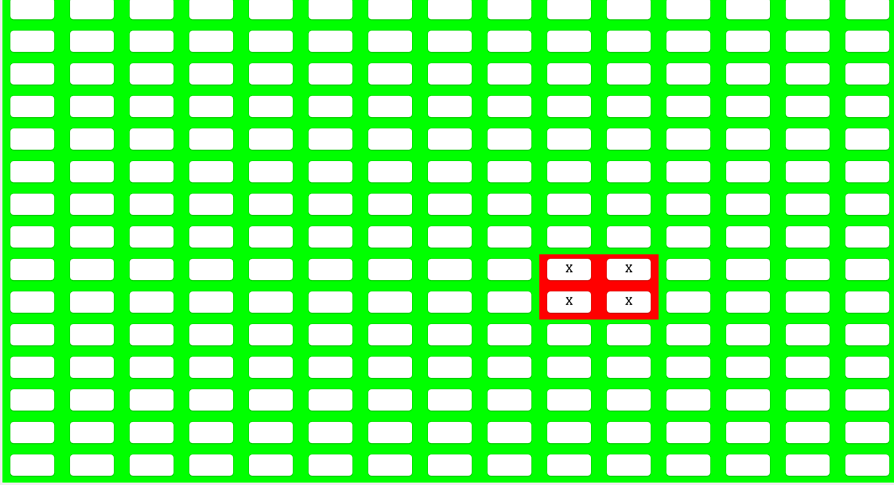


With the modified toString() method nothing appears on the VM profiler I assume that time to operate that method is low enough. In addition to that, I got the real time to operate “write” with timer and checked the result by “load” the written pattern, the function after modification runs very quick and result is not changed.After that tried clear, load, undo. Nothing happened in the VM.

Then I clicked “run continuous” with this pattern and 30 seconds (time control) after it was stared I stopped the running. The profiling showed as below:

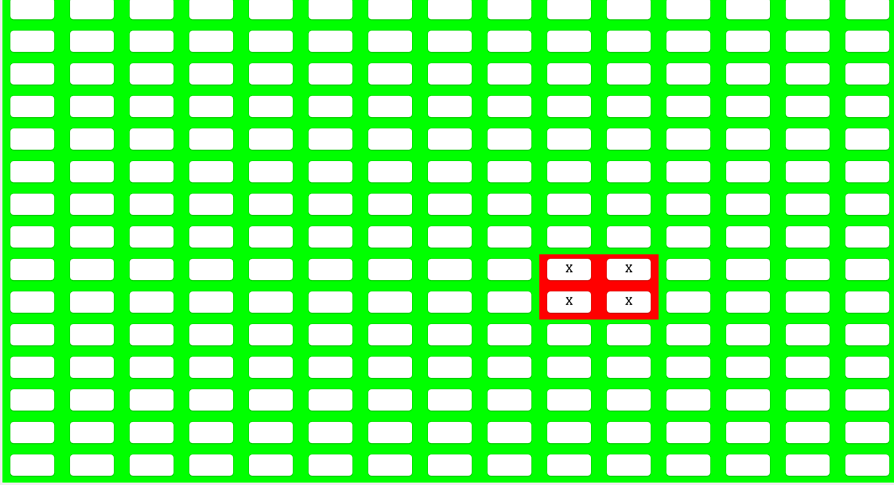


and the result is showing below:

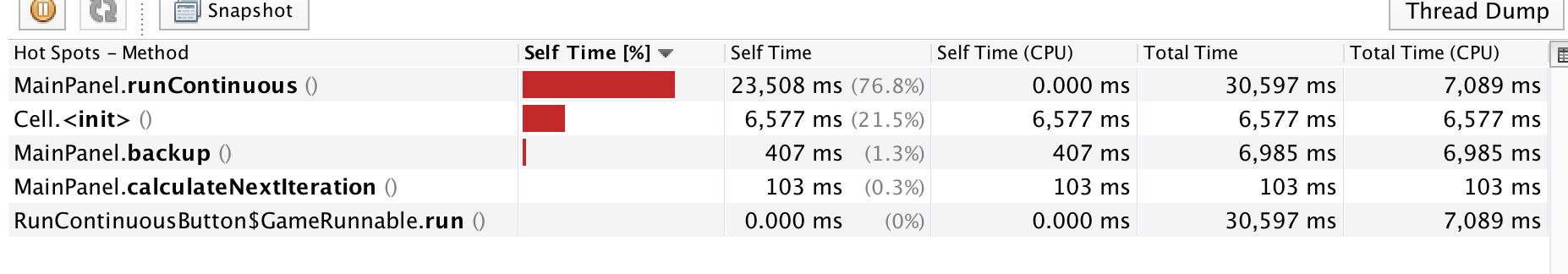


I noticed that there are some methods should be modified to improve their performance based on the results. convertTolnt(), runContinuous(), toString() and so on.

This is the result after modifying the method,



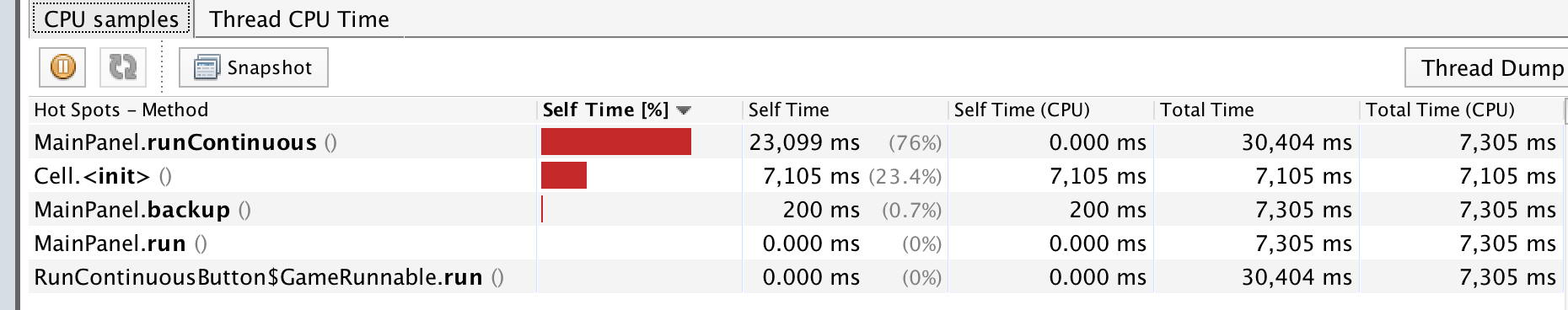
After convertToInt is modified() with deleting meaningless while loop, it turns out time for running convertToInt() method dose not make it being accounted.



Then I turned to runContinuous() method. Although it does not take much CPU time, the running time of this method could be reduced by taking the “for” loop away from the code.

I used manual test for this method since it has no return value which makes it hard to be unit-tested. After I modified the runContinuous() method time for the pattern finally becomes to that result above (picture does not change any more) comes to 9 seconds from 12 seconds.

The profiling result after runContinuous() modified:



Test:

1. MainPanel.convertTolnt(int)

Give the argument to convertToInt, should return the integer type of the argument based on 3 tests with different size of word.

1. MainPanel.runContinuous()

Test the results between the original and modified method runContinous() are same or not.

1. Cell.toString()

It returns the string type of the current state of cell. When initiate a cell with different type, it should return differently.