Ex. 1 (2 pts.)

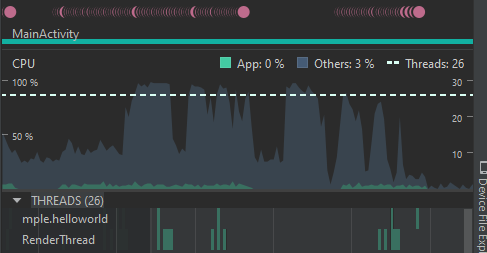
A. Out of the code that you wrote (methods you wrote or methods you called), which

consumes the most CPU time? Hint: try searching for classes that you created, like your

Activity.

The button’s on click listener consumes the most cpu time, the listener is in mainactivity.

B. Include a screenshot to support your answer to the previous question.



C. Discuss whether the results were expected, or if there were any surprises, and why.

The areas where there is a spike on the timeline and green marks on the render thread is when I am spam clicking the button to change the randomly generated image. This is expected as this is where the app is finding a new random integer and rendering this image onto the screen as well as having a little animation on the button to indicate that it has been pressed.

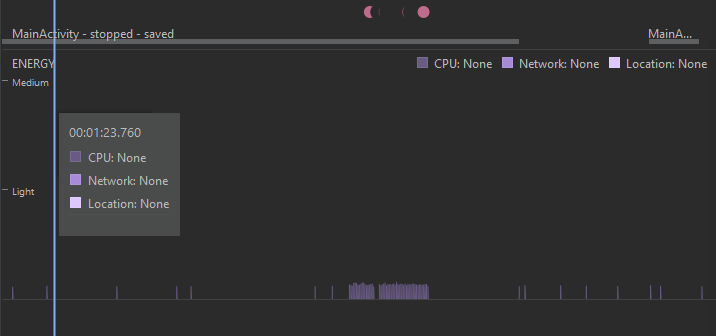
Ex. 2 (3 pts.)

A. Zoom out on your timeline until you can see all of the interesting features from your

session. Take a screenshot. On your screenshot, label the interesting features. For

example, label where you started playing the game and where the game ended. Include

this labeled screenshot in your report.



App is not on main screen

App is open and running

Button Press

B. Based on what you see here, does your app use more or less energy than you expected?

Discuss.

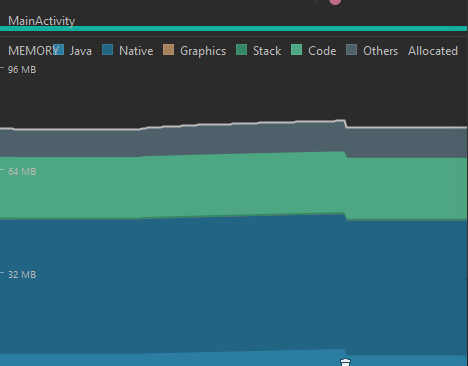
It uses about the amount of energy I would expect. The only action that this app is performing is the action to generate a new number. What is surprising to me is that the energy consumption seems to increase whenever the app is not on the main window.

Ex. 3 (2 pts.)

A. What is the general trend in memory consumption of your app over time? Include a

screenshot to support your answer.

The general trend is an increase in consumption over time as the button is pressed, otherwise it stays stable.



B. Did you expect the trend in the previous question, or is it a surprise? Discuss.

I did expect this trend since the apps action would increase the overall memory being consumed by the java section since the app is running on java.

C. Based on your observations, which of the memory buckets is used to store objects (i.e.

the object heap)? Why do you think that?

The stack memory would store the object heap since it is the only bucket which increases after the garbage dump. It doesn’t show in the image but before the dump it was at 618 KB and after it was at 620kb.

Ex. 4 (3 pts.)

1. Examining the code, how does this memory leak work?

This memory leak creates a million cat objects but never removes them from memory, even after the app is closed. Whenever a new cat is made it gets added to a list of cats which never gets cleared in BagOfCats.

B. Force your app to run out of memory on a virtual device. How much memory is being

consumed by the app when this happens? What specific Java exception is thrown (check

Logcat)?

The OutOfMemoryError occurs when the app is using over 414.9 Mb of memory.

C. Open the advanced settings for the virtual device (AVD manager -> Edit the AVD -> Show

Advanced Settings). Scroll down to the Memory and Storage settings. Based on the

settings shown here and your answer to the previous question, why is the app crashing?

Is the device out of memory? Note: you probably won't see exactly the numbers shown

here in your profiling session.

The device is set to have 1536 MB of ram but the app runs out of memory because most of that ram space is used to run essential processes that the phone needs to operate normally. The amount of memory allocated to app processes must have run out since the cats were not being cleared.