

Performance in Music Blocks

Name: Favour Kelvin .A.

Email: favourkelvin17@gmail.com

Github Profile: <https://github.com/fakela>

First Language: I speak English very well

Location: Port Harcourt , Rivers State , Nigeria WAT + 1

Education: I am an undergraduate student studying Microbiology at Abia State University ,Abia ,Nigeria

Your Motivation

What is your motivation to take part in Google Summer of Code ?

Google Summer of Code is a place where you do not just get to apply your skills but also get to acquire and learn new ones and also build your current skills. It is also a great initiative to help the open source community and i want to be part of the open source community where i can make contributions, write codes, interact and share ideas with some great people ,discuss fun projects ,get feedback on code and just about anything but most importantly, become a better software developer. Through GSOC ,organisations give undergraduates like me and newbies in open source the opportunities to contribute , propose ideas and features and also implement them.

Why did you choose Sugar Labs ?

Sugar Labs is an open source learning platform for children , it gives me the opportunity to be able to join in the contribution .collaboration and creation of software that works well for children as well as their teachers .The people in the community are also friendly and always ready to help and direct you and i feel very comfortable working around them and i know i would learn a lot from contributing to this organisation

Why do you want to work on this particular project ?

Music Blocks is used by many and i liked the idea of music blocks from when i was working on my code project that has to do with sounds and musics and i stumbled upon music blocks and i felt i could learn more from contributing to music blocks and thereby committed myself to exploring their codebase. I have developed a fascinating interest in this project and eventually i have resolved some issues and made some contributions.

What are your expectations from us during and after successful completion of the program ?

My expectations is to be able to interact more with the organisation and share ideas and also to integrate more into the open source community. I also expect the same support i received from when i starting contributing to the organisation till now.After the completiion of the program i still want to be able to interact with the organisation and also intergrate more into the open source community

Project Details

What are you making ?

- The main goal of this project is to analyze results and determine where the application performance can be improved , the goal of performance testing is not to find bugs but eliminate performance bottlenecks
- Performance issue varies ,through this testing we identify the performance problem by highlighting where the application might fail or lag(e.g identifying for memory leaks
- I would like to improve the performance of music blocks by tuning the performance taking note of both the memory footprint and the CPU consumption, using various performance test tools,making changes to the code base that will enhance performance

How will it impact Sugar Labs ?

Music blocks is used by many many people and we want the users to interact meaningfully with music blocks hence its performance like its response time, reliability, resource usage and scalability do matter for better user experience and case studies also show that high performing web apps engage and retain users better than low-performing web apps

What technologies (programming languages, etc.) will you be using ?

I will be using HTML, CSS and javascript to implement the features

PLAN OF ACTION

The main objective of testing is to analyze results and determine where the application's performance can be improved and also improve users experience

Music Blocks is to be used in schools where old and under powered computers are used and sometimes slow network, they may be delays depending on network conditions and hardware

Goal: Try to reduce user waiting time and deliver content and become interactive, tune the performance by identifying the causes of memory leaks, memory footprints and garbage collection and CPU consumption, finding out what resources are currently being consumed, optimizing JavaScript and CSS for performance.

Before doing a performance testing we basically need to know the following points

1. Expected no of concurrent users or HTTP connections with music blocks
2. Acceptable response time for your the music blocks

For performance tuning am using this two approach.

In Approach 1(**white-box**),

- Code Analysis, searching for poor algorithms or looping which is the reason for inefficiency,
- Database Analysis, query optimizers and profilers to optimize the database.
- Hardware & Network, use of utilities such as top, iostat to monitor hardware

In Approach 2(**black-box**), using of test tools that simulate concurrent users/HTTP connections and measure the response times automatically. If the response time does not meet your expectations tuning has to be done at application/hardware/database level.

In Tuning,

First we need to enhance the application code efficiency, then we can optimize the database

If still your application doesn't meet your requirements then the following steps will help you.

1. Using cache mechanisms.
2. Publish highly requested code statically, so that they don't hit the database.
3. Scaling Web servers horizontally via load balancing.
5. Scale the servers vertically by adding more hardware resources (CPU, RAM)

Tools for measuring Performance

- Chrome dev tools
- WebPage Test
- Lighthouse
- Page Insight and other testing tools

Optimizing JavaScript

JavaScript is the slow resource and a dynamic garbage collection programming language and the first step to fixing JavaScript performance issues finding the problem .

The use of too many dependencies (libraries), poor event handling, the use of inefficient iterations, too many host interactions can cause a lag in performance

Some methods i will use for optimization of JavaScript

Throttling and debounce: When we add event listeners to user actions like a scroll, we ignore the fact that listener(s) fire when our events get triggered. This is a potential bottleneck for our JavaScript application

Everytime a user scrolls, "large" gets logged to the console.

This doesn't seem bad, but with more expensive operations like checking if an element is in the viewport so we can animate, this becomes expensive with time and uses more memory. A simple way to fix this is to either debounce or throttle the heightBasedScroll function.

Minify

Removal of unwanted and cumbersome files ,Unwanted refers to comments, semi-colons, whitespace etc. While cumbersome refers to shortening function and variable names, reducing an if-else into ternary etc.

Promises

JavaScript promises use a fluent API to describe code and being a native function, they are completely optimised and should be used frequently. The fact that promises are asynchronous means they are not blocking thus improving the speed of your application

Async & Defer

async and defer are attributes we can add to script tags to make them either load asynchronously to the page or defer until the page has completely loaded. Doing either async or defer doesn't block the DOM from rendering, which makes the application feel faster.

Use of reference types

While primitive value types like strings and integers get copied every time they are passed into a new function, reference types, like arrays and objects, are passed as light-weight references, comparing strings always takes longer than comparing references.

Cut down scope chain

When functions are executed in JavaScript, a set of first order variables including the immediate scope chain, the arguments of the function and any locally-declared variables are instantiated. Therefore, it takes time to climb up the scope chain when you try to access a globally-declared variable. Reducing the the call stack depth and taking advantage of the this keyword can speed up execution.

Replace 'click' with 'mouseup'

Binding functionality to the 'mouseup' event, which fires before the 'click' event, provides a performance boost by ensuring that no interactions are missed if the user makes several mouse clicks in rapid succession.

Minding event handlers

Since events like 'mousemove' and 'resize' execute hundreds of times per second, paying special attention to any event handlers bound to those events. If they take more than 2-3 milliseconds to complete, optimization of the code becomes an option

Cache as much as i can

Caching is your greatest asset for speeding up load times. Leveraging browser caching as well as intermediary caching mechanisms such as a content delivery network. This will ensure that your assets load quickly both for previous visitors as well as first time visitors.

Profiling the code

Chrome developer tools is a very robust range of tools. Apart from a console, DOM inspector, it also comes with a profiler. Chrome developer tools is a very robust set. Apart from a console, DOM inspector, it also comes with a profiler,

This tool checks runs a series of tests on your web application and looks for memory leaks. If it discovers anything, it is then displayed in a graph, showing you potential bottlenecks and memory leaks.

After fixing these problems, there would be a noticeable improvement in music Blocks performance

TimeLine

Community Bonding Period (May 6 -May 26)

During this period i aim to go through the entire code base thoroughly and set up test environments . Currently i have enough knowledge to modify some features.

However,going through the code base will make me with the work faster during the coding period

I will also like to interact with my mentors and also get to know the community and also help in whatever way i can

Week1(May 27-june 3)	Test musicblock speed using test tools (ie
----------------------	--

	webPage test,chrome dev tools, lighthouse) to analyze the overall speed of music blocks putting in mind low-end computers
Week2(June 3-June 10)	Load test the site to see what might be causing the bottlenecks using test tools (ie loader)
Week3(june 11-june 17)	Check the causes of memory leaks and
Week 4(june 18-june 24)	Time period for an expected delay
June 24(First Evaluation phase)	Deliverables: Result of the various test carried out, knowing the potential bottlenecks and causes of memory leaks
Week 5(june 25-july 7)	Working on the identified problems ,optimization of the javascript
week 6(jul 8-jul 14)	Continue with optimizing javascript and adjusting the codebase
week 7(july 16-july 22)	Time period for unexpected delay
july 22(second Evaluation phase)	Deliverables: Optimized javascript code and fixing of issues that were identified in the process of testing
week 8(july 23-july 29)	Test run again and identify bottlenecks that needs to be taken care of
week 9(july 30-august 7)	Continue with fixing of issues and performance bottlenecks encountered in the second testing
week 10(august 7-august 13)	Run final test and check for noticeable improvement in music blocks

week 11(august 14-august 19)	Time period for any unnecessary delay
------------------------------	---------------------------------------

august 19(Final evaluation)

Deliverables:The different performance factors (i.e causes of slow response etc)that was identified ,resolved and tested

Mention how much time will you spend each week working on your project

I will be working for 5-6 hrs in a day

Sugar Labs' Motivation:

Convince us that you will be a good fit for this project, by sharing links to your contribution to Sugar Labs?

To get familiar with the code base i made a few contribution here Is a link to that

<https://github.com/sugarlabs/musicblocks/commits?author=fakela>