Metronome

Table of Contents

[Feasibility study 1](#_Toc123728149)

[Analysis 1](#_Toc123728150)

[Design 1](#_Toc123728151)

[Implementation 2](#_Toc123728152)

[Testing 2](#_Toc123728153)

[Installation 2](#_Toc123728154)

[Evaluation 2](#_Toc123728155)

[Maintenance 3](#_Toc123728156)

# Feasibility study

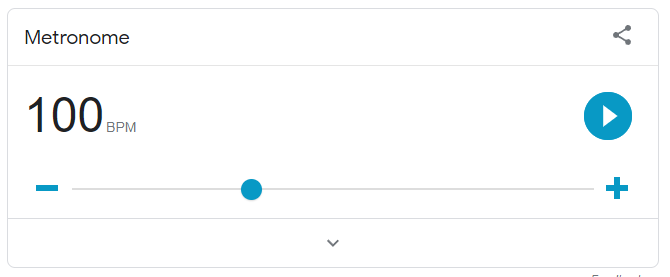
I would like to make a web based metronome. A metronome is a timer used by musicians so that they can keep track of their timing.

I know that this problem is solvable as there are online solutions already, (working out how fast to make the beep sounds) can be solved in a finite number of steps. The main calculation is converting a BPM (e.g. 60 beats per minute) to the delay (in milliseconds) between each beep. This is computationally simple. Using a theoretical approach.

The budget is limited (£0) and we have 2 weeks to complete the project.

# Analysis

Elliot is a 21-year-old man who has a hobby of music. He needs a simple metronome application that he can use easily each day.



<https://g.co/kgs/Ehuk8J>

Google has a built-in metronome. It’s free and easy to use. It has simple UI and also visually pulses meaning you don’t necessarily need audio on. Although it doesn’t show how far each beat you are like a traditional metronome.

# Essential features

The solution needs to allow the user to set a specific BPM and start/stop the beat. On each beat it should make an audible noise (tick) and give visual indication in case someone cannot hear the audio.

The solution needs to work on any web browser.

There should be limited amounts of text so it’s more suitable for children.

There needs to be some buttons for adjusting the BPM.

A mini tutorial to quickly teach the user how the app works.

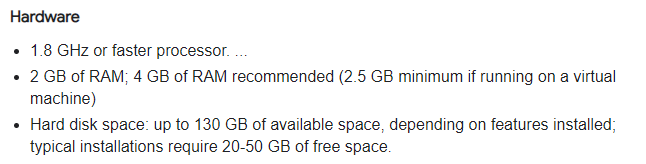
# Limitations

There isn’t enough time to add all additional features to the app however they could be added into later. My solution will not be able to compare your beat to see if you are in time. There is also no feature to log your progress. Also, another good feature would be to set the BPM over time so it changes.

There is no requirement to make a native android/apple app: the metronome is going to be purely on the web to make it as freely available as possible.

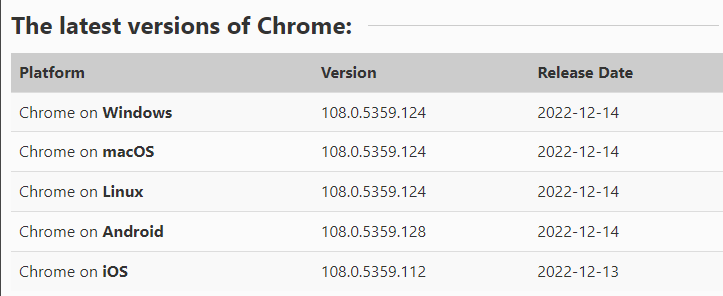
# Hardware and software requirements

For development, Visual Studio 2017 will be used which requires:



<https://learn.microsoft.com/en-us/visualstudio/releases/2017/vs2017-system-requirements-vs>

The user will also need to use a modern browser (e.g. chrome)

https://www.whatismybrowser.com/guides/the-latest-version/chrome

This is so that all the features of css / javascript / HTML5 work as expected.

My stakeholder has a Samsung Galaxy Tab A7 32 GB Wi-Fi Android Tablet – Dark Grey (UK Version)

|  |  |
| --- | --- |
| Screen size: | 10.4 inches |
| OS: | Android 12 |
| Storage: | 32GB |
| RAM: | 4GB |

# Success Criteria

|  |  |  |
| --- | --- | --- |
| Number | Criteria | Justification |
| 1 | Must be accessible on an android 12 device in Chrome which is connected to the internet | The stakeholder has an android 12 device with reliable WiFi connection. Any updates to the app can be automatically rolled out to the users. |
| 2 | User can set the BPM | The stakeholder wants to practice between 50BPM and 120BPM |
| 3 | Any invalid BPM will be automatically corrected to the closest sensible value between 50-120BPM | The stakeholder wants to focus on drumming and might accidentally input invalid data |
| 4 | The user is able to shift the BPM by 5 | The stakeholder wants to be able to practice drumming at different speeds as they get better |
| 5 | There should be a start and stop button that toggles playing when pressed | The user should be able to control the metronome with their finger using the simple UI |
| 6 | When the metronome starts there should be an audible beep and visual cue | The stakeholder needs to drum along to a click sound using headphones |
| 7 | When the metronome starts there should be a visual indication showing when it would make a sound | Sometimes the metronome would be used without sound |
| 8 | A high contrast mode button to toggle between dark and light mode. Inside a settings menu | Sometimes the metronome will be on a small screen a reasonable distance from the drummer and they need to see it at a glance without misreading anything |

# Design

# Algorithms

The user should be able to enter the BPM (beats per minute). The browser then needs to know how long to pause for between each tick. This can be done using the following algorithm.

BPM = user input

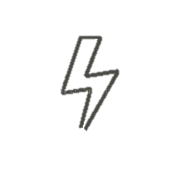
Interval = 1 \* 60 \* 1000 / BPM

# Usability features

67 BPM

v

v



# Implementation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test number | Description | Success criteria | Test data | Expected result | Actual result |
| 1 | Basic web page | 1 | Page load | A web page with a title and text box to enter the BPM into. Two buttons to adjust the BPM |  |
| 2 | BPM | 2 | 120 | Valid data is accepted |  |
| 3 | BPM | 2 | “” | BPM set to the default of 50 |  |
| 4 | BPM | 2 | 30 | BPM set to the default of 50 |  |
| 5 | BPM | 2 | 130 | BPM set to 120 |  |
| 6 | BPM | 2 | Potato | BPM set to the default of 50 |  |
| 7 | Increase BPM | 4 | BPM set to 60 and press increase | BPM increases to 65 |  |
| 8 | Increasing BPM | 4 | BPM set to 120 and press increase | BPM stays at 120 |  |
| 9 | Increase BPM | 4 | BPM set to 119 and press increase | BPM set to 120 |  |
| 10 | Decrease BPM | 4 | BPM set to 60 and press decrease | BPM set to 55 |  |
| 11 | Decrease BPM | 4 | BPM set to 0 and press decrease | BPM stays at 0 |  |
| 12 | Start metronome | 5 | BPM set to 60 press the start button | Metronome starts to tick at 60 BPM |  |
| 13 | Stop metronome | 5 | While metronome running press the stop button | Metronome stops ticking |  |
| 14 | Ticking noise | 6 | BPM set to 60 and press start | Makes a tick noise every second |  |
| 15 | Visual ticking | 7 | BPM set to 60 and press start | Makes a visual tick every second |  |
| 16 | Settings menu | 8 | Press settings menu button | Shows the settings meu |  |
| 17 | Toggle light mode | 8 | Press the settings button and toggle light mode | Changes the visual theme |  |

# Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test number | Description | Success criteria | Test data | Expected result |
| 1 | Basic web page | 1 | Page load | A web page with a title and text box to enter the BPM into. Two buttons to adjust the BPM |
| 2 | BPM | 2 | 120 | Valid data is accepted |
| 3 | BPM | 2 | “” | BPM set to the default of 50 |
| 4 | BPM | 2 | 30 | BPM set to the default of 50 |
| 5 | BPM | 2 | 130 | BPM set to 120 |
| 6 | BPM | 2 | Potato | BPM set to the default of 50 |
| 7 | Increase BPM | 4 | BPM set to 60 and press increase | BPM increases to 65 |
| 8 | Increasing BPM | 4 | BPM set to 120 and press increase | BPM stays at 120 |
| 9 | Increase BPM | 4 | BPM set to 119 and press increase | BPM set to 120 |
| 10 | Decrease BPM | 4 | BPM set to 60 and press decrease | BPM set to 55 |
| 11 | Decrease BPM | 4 | BPM set to 0 and press decrease | BPM stays at 0 |
| 12 | Start metronome | 5 | BPM set to 60 press the start button | Metronome starts to tick at 60 BPM |
| 13 | Stop metronome | 5 | While metronome running press the stop button | Metronome stops ticking |
| 14 | Ticking noise | 6 | BPM set to 60 and press start | Makes a tick noise every second |
| 15 | Visual ticking | 7 | BPM set to 60 and press start | Makes a visual tick every second |
| 16 | Settings menu | 8 | Press settings menu button | Shows the settings meu |
| 17 | Toggle light mode | 8 | Press the settings button and toggle light mode | Changes the visual theme |

# Variables and validation

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Data type | Validation | Justification |
| BPM | Integer | Range check between 20 and 240 | The beats per minute should be limited |
| BPMUserInput | String | Presence check and type check | The BPM should be able to be converted to an integer between 50 and 120 BPM |
| IntervalMS | Integer | Make sure it is an integer | The browser needs to have an interval between ticks |
| LightMode | Boolean | Switch that can be toggled | To toggle the theme |
| IsPlaying | Boolean | Can be either true or false | The metronome can only be either be playing or paused |
|  |  |  |  |
|  |  |  |  |

# Post development testing

Some questions to give to my stakeholders after I’ve finished developing the solution:

Functionality

What did you want the program to do? What did you click on (exactly, in what order)? What happened? Did the program do everything you wanted it to do? Were there any features missing?

Robustness

Did the program crash? If so, when (what did you do immediately before? What were you trying to do?)

Usability

How simple was the program to use? Is a tutorial needed?

# Installation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.

Aenean nec lorem. In porttitor. Donec laoreet nonummy augue.

Suspendisse dui purus, scelerisque at, vulputate vitae, pretium mattis, nunc. Mauris eget neque at sem venenatis eleifend. Ut nonummy.

# Evaluation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.

Aenean nec lorem. In porttitor. Donec laoreet nonummy augue.

Suspendisse dui purus, scelerisque at, vulputate vitae, pretium mattis, nunc. Mauris eget neque at sem venenatis eleifend. Ut nonummy.

# Maintenance

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna.

Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.

Aenean nec lorem. In porttitor. Donec laoreet nonummy augue.

Suspendisse dui purus, scelerisque at, vulputate vitae, pretium mattis, nunc. Mauris eget neque at sem venenatis eleifend. Ut nonummy.