Metronome

Table of Contents

[Feasibility study 1](#_Toc124168458)

[Analysis 1](#_Toc124168459)

[Stakeholders 1](#_Toc124168460)

[Research 2](#_Toc124168461)

[Essential Features: 2](#_Toc124168462)

[Limitations: 2](#_Toc124168463)

[Hardware and Software requirements: 2](#_Toc124168464)

[Success Criteria: 4](#_Toc124168465)

[Design 6](#_Toc124168466)

[Algorithms: 6](#_Toc124168467)

[Usability features: 6](#_Toc124168468)

[Variables and validation 7](#_Toc124168469)

[Testing 7](#_Toc124168470)

[Implementation 9](#_Toc124168471)

[Installation 9](#_Toc124168472)

[Evaluation 9](#_Toc124168473)

[Maintenance 9](#_Toc124168474)

# Feasibility study

I would like to make a metronome. A metronome is a tool used by musicians to keep in time.

I know that this problem is solvable because the problem (Working out how fast to make beep sound) can be solved in a finite number of steps. The main calculation is converting a BPM (e.g. 50 beats per minuts) to the delay (in milliseconds) between each beep. This is computationally simple using a theoretical approach.

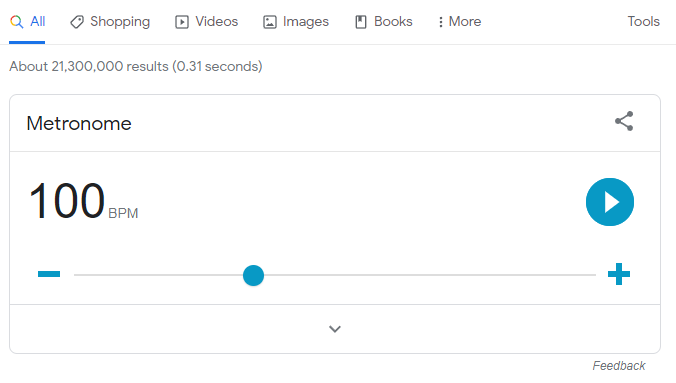
The budget is limited (£0) and I have 2 weeks to complete the project.

# Analysis

## Stakeholders

Tim is a 37 year old musician showing an interact in using computers to generate music. He will use the metronome to stay in time with music when he is playing instruments at home. Tim is hard of sight so the metronome will use noise to particularly aid him.

## Research



<https://www.google.com/search?q=metronome>

Google has a built-in metronome. It’s free and really easy to use. It has a nice visual pulsing feature so you don’t need audio enabled. However, it doesn’t indicate how far through each measure you are like a traditional metronome.

## Essential Features

My solution must include an audible ticking noise and allow you to set the BPM. It must allow you to start / stop the beat and give a visual indication in case someone can’t hear the tick.

The solution must work on any web enabled device with a clear, touchscreen friendly interface.

There should be limited text so it is suitable for all people.

There should be buttons to allow you to increase or decrease the BMP by 5 without starting / stopping the beat.

It should be freely accessible online.

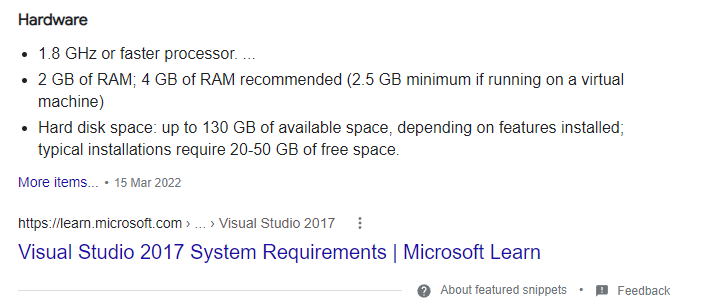
## Limitations

There is insufficient time to add additional features but these could be added later. My solution will not allow you to compare your playing to the beat to see how in time you are. There will be no feature to log in and track your progress. It would be great if there was a feature to change the BPM automatically over time but this is not essential as development time is limited.

There is no requirement to make a native android or apple app: the metronome will be purely web based in order to make it freely available to as many people as possible.

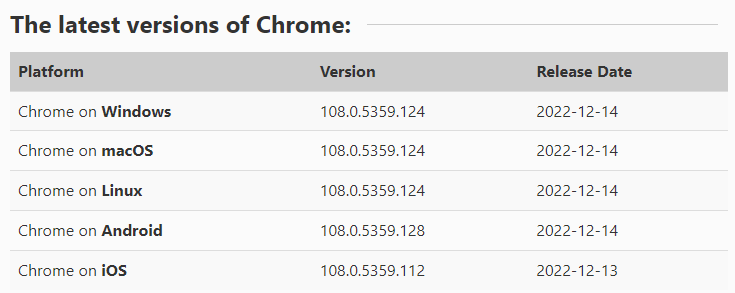
## Hardware and Software requirements

For development, I need to be able to use Visual Studio 2017 which requires:



<https://www.google.com/search?q=visual+studio+2017+requirements>

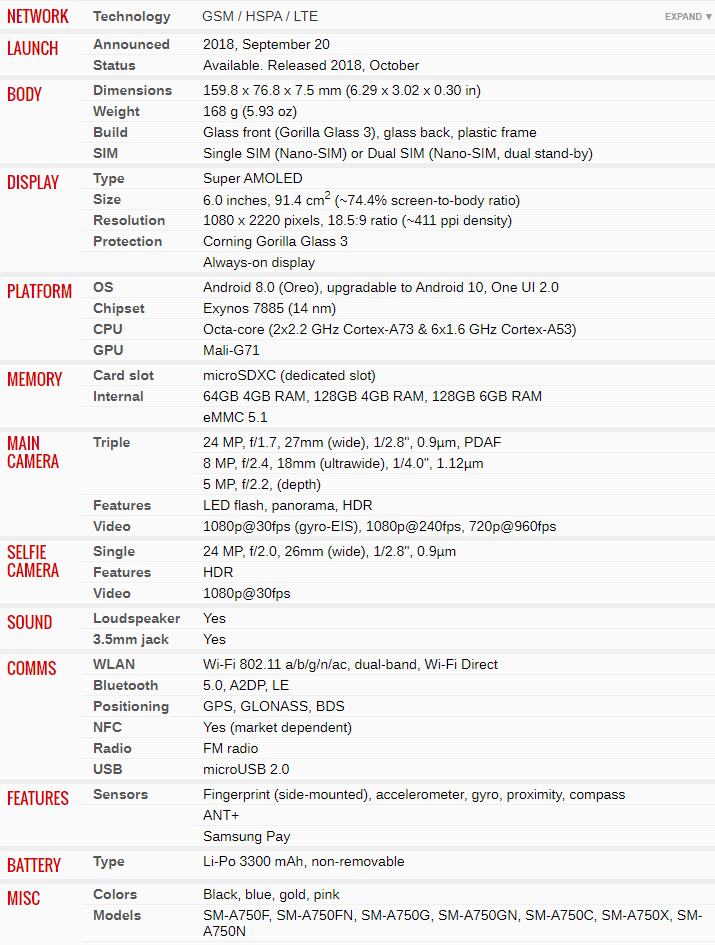
For the stakeholders to run the metronome they need a web enabled device running a modern browser (e.g. chrome)



<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 A7 (2018) 64GB Wi-Fi Android Phone – Black (UK Version)



<https://www.gsmarena.com/samsung_galaxy_a7_(2018)-9340.php>

## Success Criteria

|  |  |  |
| --- | --- | --- |
| Number | Criteria | Justification |
| 1 | Must be accessible on an android 8.0 device in Chrome connected to the internet | The stakeholder has an android 8 device with a reliable WiFi connection. Any updates to the app can be automatically rolled out the users |
| 2 | User can set the BPM | The stakeholder wants to practice between 50BPM and 120 BPM |
| 3 | Any invalid BPM will be automatically corrected to the closest sensible value between 50-120BPM | The stakeholder wants to focus on playing an instrument and might accidentally type invalid data |
| 4 | The user should be able to increase the BPM by 5 using a touch button | The stakeholder wants to be able to practice playing an instrument at different speeds as they become more proficient |
| 5 | The user should be able to decrease the BPM by 5 using a touch button |
| 6 | There should be a start and stop button that toggles when you press it | The user should be able to control the metronome with their finger with a really simple UI |
| 7 | When the metronome starts there should be an audible beep | The stakeholder wants to play an instrument along to the click sound in headphones |
| 8 | When the metronome starts there should be a visual indication showing whenever it would ‘tick’ | Sometimes the metronome would be used with the sound turned off |
| 9 | There should be a high contrast mode button which toggles a black and white colour scheme with default colours | Sometimes the metronome will be on a small screen a reasonable distance from the user and they need to see it at a glance without misinterpreting any numbers |

# Design

## Algorithms

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

BPM = user input

Interval = 1 \* 60 \* 1,000/BPM

## Usability features

Normal mode

Start

-5

+5

80

High contrast mode

**80**

**-5**

**+5**

**START**

## Variables and validation

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Data type** | **Validation** | **Justification** |
| BPM | Integer | Range check (between 50-120) | The beats per minute should be anything from one to two beats per second |
| BPMUserInput | String | Presence Check and TypeCheck | The BPM should be able to be converted to an integer between 50 and 120 BPM |
| IntervalMs | Integer | Make sure it’s rounded to 0 d.p. | The browser needs an interval between ‘ticks’ in milliseconds as a whole number |
| HighContrastMode | Boolean | True or false | Checkbox that can be ticked or unticked |
| IsPlaying | Boolean | Either true or false | The metronome can either be playing or paused |
|  |  |  |  |
|  |  |  |  |

# 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 that allows the user to enter a BPM with buttons that allow BPM to be increased/decreased by 5 |
| 2 | BPM | 2 | 120 | Valid data accepted |
| 3 | BPM | 3 | “” | BPM set to 50 |
| 4 | BPM | 3 | 30 | BPM set to 50 |
| 5 | BPM | 3 | 130 | BPM set to 120 |
| 6 | BPM | 3 | Potato | BPM set to 50 |
| 7 | Increasing BPM | 4 | BPM set to 60. Press increase | BPM increases up to 65 |
| 8 | Increasing BPM | 4 | BPM set to 120. Press increase | BPM set to 120 |
| 9 | Increase BPM | 4 | BPM set to 119. Press to increase | BPM set to 120 |
| 10 | Decrease BPM | 5 | BPM set to 80. Press decrease | BPM decreases down to 75 |
| 11 | Decrease BPM | 5 | BPM set to 50. Press decrease | BPM set to 50 |
| 12 | Decrease BPM | 5 | BPM set to 51. Press decrease | BPM set to 50 |
| 13 | Start/Stop | 6 | BPM set to 60. When stopped, press start toggle | Start ticking at 60 bpm. Start button turns to stop. |
| 14 | Start/Stop | 6 | BPM set to 60. When running, press start toggle | Stop ticking. Stop button turns to start |
| 15 | Audible beep | 7 | BPM set to 60. Press start | Hear a tick every second |
| 16 | Audible beep | 7 | BPM set to 120. Press start | Hear two ticks every second |
| 17 | Visual indication | 8 | BPM set to 60. Press start | See animation that indicates when a tick occurs once a second |
| 18 | High contrast | 9 | Click high contrast button when high contrast disabled. | UI changes to black and white high contrast mode with larger text |
| 19 | Disable high contrast | 9 | Click high contrast button when high contrast enabled. | UI changes to regular dark blue stylised UI |

## Post development testing

Here are some questions to give my stakeholder 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 that you wished were there but weren’t?

Robustness

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

Usability

Did you understand what you had to do? Did you encounter any difficulties? Would any hints have benefitted your experience?

# Implementation

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.

# 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.