Metronome Documentation

# Feasibility study

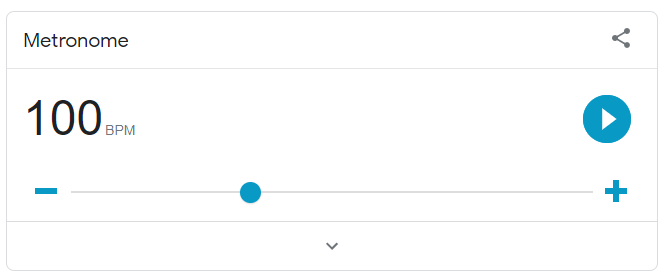
I would like to make a web-based metronome. A metronome is

I know that this problem is solvable because the problem 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.

# Analysis

## Google Metronome

This metronome is built into the google search engine and can be found by searching metronome on google.



This metronome is very intuitive and easy to use. You set how many beats per minute and press play. However, ease of use comes at the cost of a limited set of features.

This metronome cannot do any of the extra features offered by other digital metronomes but is quick and easy to use which for many musicians is the priority.

## Mechanic Physical metronomes



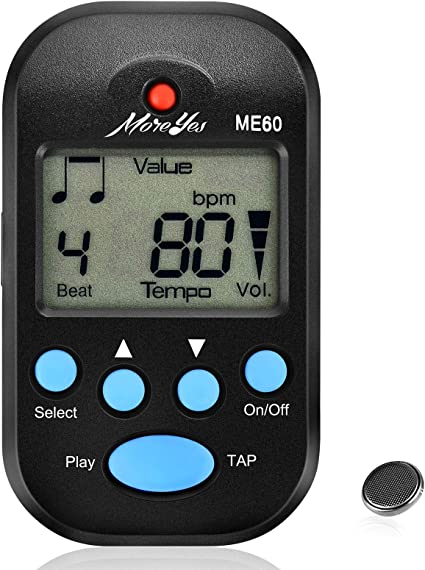
Physical metronomes were the original solution to the problem of needing a steady beat at a set BPM and still have some advantages over digital metronomes.

* As they are mechanical devices, they do not require power or internet connection.
* They can be more accurate than digital metronomes although this level of precision is rarely if ever noticeable or necessary.

However, for most people these advantages will not make it worthwhile to buy a physical metronome as the disadvantages outweigh them so are usually just bought by enthusiasts.

* Digital metronomes are free or far cheaper
* More difficult to transport
* They may not have as many features as digital metronomes, such as the ability to set different time signatures or create complex rhythms.

## Digital physical metronomes



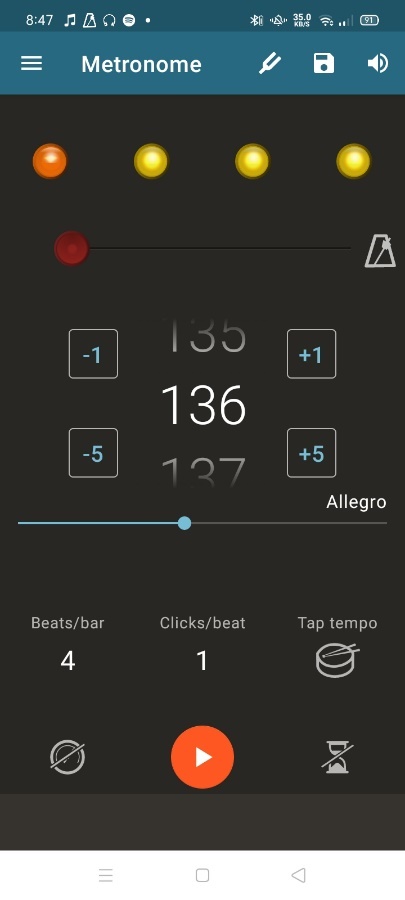
Physical digital metronomes are metronomes that combine the features of both physical and digital metronomes. Advantages of these metronomes include:

* They have a wider range of features and customization options than physical metronomes, as they have an electronic component.
* They are be more portable and easier to transport than purely physical metronomes.

However, they still have the disadvantages inherent to a physical device when compared to an application.

* They are more expensive to purchase than purely digital metronomes.
* They are not be as accurate or reliable as purely physical metronomes, as they have electronic components that can be affected by issues such as interference.
* They may have a shorter lifespan than purely physical metronomes, as they have electronic components that can fail over time.

## Metronome beats

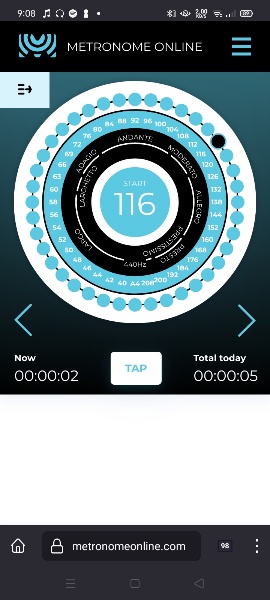
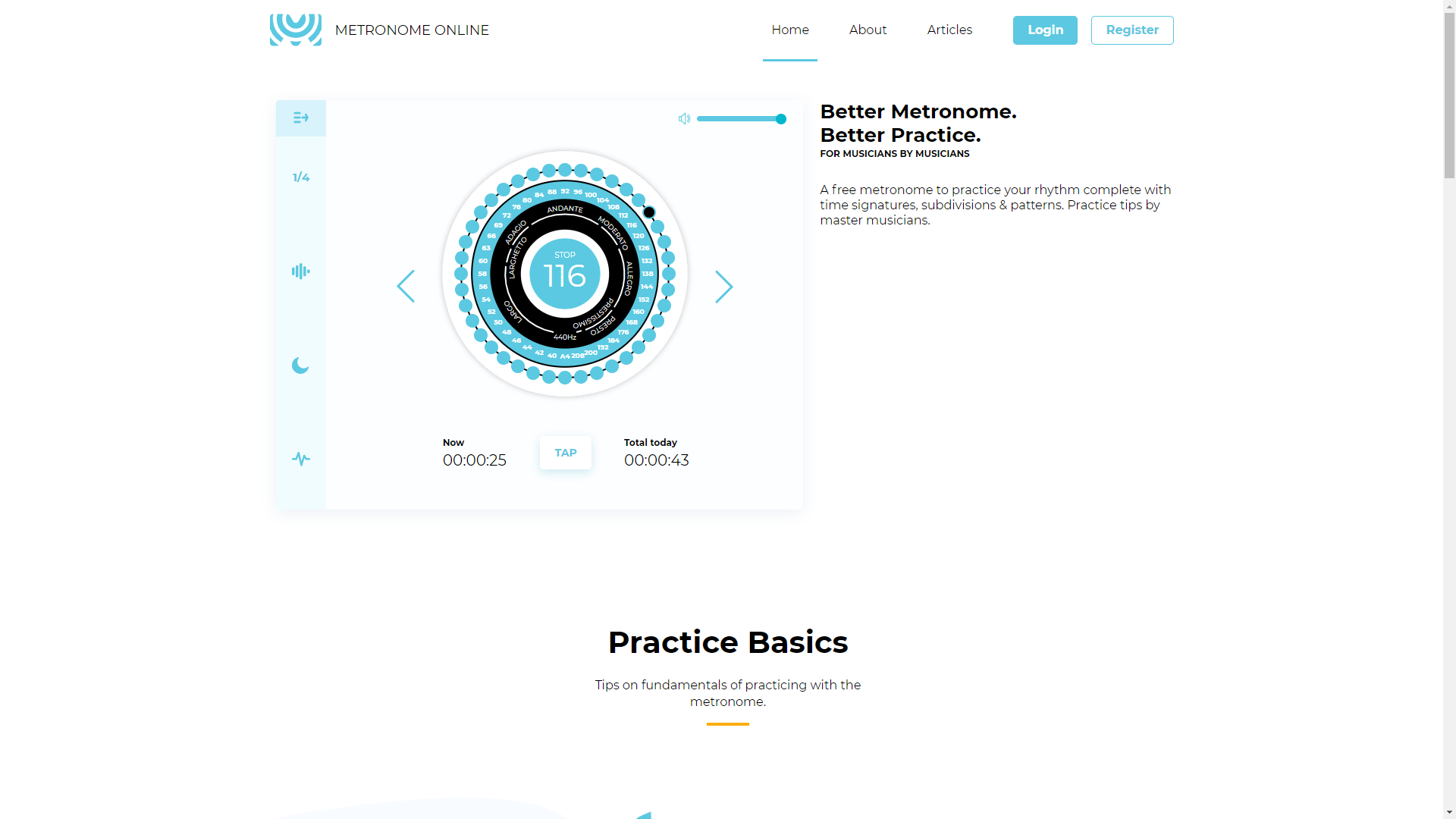


Metronome beats is the most downloaded metronome app on the google play store with over 10 million downloads. The app is free to download but for access to the full range of features and to remove the adds, a one-time payment of £4.49 is required. It has a wide range of features including:

* Different ways to control the BPM
  + Buttons to increment the BPM by 1 on 5
  + A slider
  + A button to tap that the metronome will copy the copy the tempo from.
  + The BPM can also be incremented by 5 from within a notification that appears whilst the app is running
* The first beat of each bar is a different sound to the others so that musicians don’t lose count
  + The number of beats per bar and clicks per beat can be configured
* The app can be configured to count in or stop after a set amount of time.
* The app has a “tempo trainer” feature that increases or decreases the BPM by a configurable amount until a set value

Having such a wide feature set is useful to those who require it although for many, it can be overwhelming, especially for a first-time metronome user.

## Metronome Online



“Metronome Online” is a very popular, free, online metronome that’s UI is designed around a wheel

* This metronome is unique in that the BPM can be set to many different values by clicking one of the circles around the outside of the wheel. Unlike other digital solutions where you either have to use a slider or increment through different values. This allows a user to quickly switch between different tempos without typing a value in. E.g. from 60 to 116 to 42
* Once a user understands how to use the wheel, it becomes quite intuitive and easy to use.
* However, it has a limited feature set compared to other digital metronomes
* On a small screen such as a phone, it becomes hard to accurately press the correct circle to set the BPM to what you need or even reading the small text.

## Stakeholders

Tim is a 15-year-old boy who’s just started drumming. He wants a simple to use metronome that he can use on his PC.

## Essential features

The metronome must make an audible tick on each of the beats at a user defined BPM.

The BPM should be changeable without stopping the beats

The beat should be programmable with the ability to set the beat to automatically change BPM after a set amount of time.

## Limitations

There should not be too many features as to detract from the ease of use.

## Development Hardware and software requirements

For development, I need to be able to use visual studio 2017 which requires:

# 

## User Hardware and software requirements

The user must have a mobile device with a browser that supports all the features of css / html / javascript

## Success Criteria

|  |  |  |
| --- | --- | --- |
|  | Criteria | Justification |
| 1 | Must be accessible on a web browser on a computer | The stake holder has a PC |
| 2 | User can set the BPM easily | The stakeholder wants to practice drumming at a variety of different BPMs |
| 3 | The BPM can change speed according to predetermined timings | The stakeholder wants to practice at different BPMs without stopping to change the metronome speed |
| 4 | There should be an audible beep at each beat | So that the stakeholder can hear the beat whilst practicing |
| 5 | There should be a visual cue at each beat | So that the stakeholder can use the metronome without audio |
| 6 | There should be a button to toggle between dark and light mode on the website | So that people with different preferences can use the metronome |
| 7 | Start / Stop button should turn the metronome on/off |  |
|  |  |  |
|  |  |  |

# Design

## Algorithms

BPM = user input

Wait timeInput seconds

BPM = userInput2

Interval = 60000/BPM

## Usability features

Input BPM

Input seconds

Input BPM

Input seconds

BPM

After

100 BPM in 10 seconds

60

+

-

# Variables and validation

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Data type | Validation | Justification |
| BPM | Integer | Range check (between 50 - 120) | The BPM should be anything from less than one to two BPS |
| 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 dp | The browser needs and interval between `ticks` in milliseconds as a whole number |
| DarkMode | Boolean | Either true or false | The Page can be in light or dark mode |
| IsPlaying | Boolean | Either true or false | The Metronome can be playing or not playing |

# 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 displays in chrome |
| 2 | BPM | 2 | 120 | Valid data accepted |
| 3 | BPM | 2 | “” | BPM set to default 50 |
| 4 | BPM | 2 | 30 | BPM to be set to minimum 50 |
| 5 | BPM | 2 | 150 | BPM set to 120 |
| 6 | BPM | 2 | wasd | BPM set to 50 |
| 7 | Increasing BPM | 2 | 60 BPM press + 5 | BPM set to 55 |
| 8 | Decreasing BPM | 2 | 60 BPM press – 5 | BPM set to 65 |
| 9 | Changing BPM after 2 mins | 3 | The BPM is set to change to 120 BPM after 2 mins | The BPM is changed to 120 BPM after 2 mins |
| 10 | Audio test | 4 | Listen for the audible beep at each tick | Hear an audible tick |
| 11 | Visual test | 5 | Look for a visual cue at each beat | See a visual cue |
| 12 | Light / dark mode test | 6 | Press Light / dark mode button | Change in colour scheme of page |
| 13 | Turn on the metronome | 7 | Press the start / stop button | The metronome starts ticking |

# Post Development Testing

Here are some questions to give my stakeholders after I’ve finished developing the solution:

## Functionality

Does the program do what you want it do

## Robustness

Did the program crash, if so, what were you doing when it crashed?

## Usability

Did you understand how the program works when using it

# 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 displays in chrome |  |
| 2 | BPM | 2 | 120 | Valid data accepted |  |
| 3 | BPM | 2 | “” | BPM set to default 50 |  |
| 4 | BPM | 2 | 30 | BPM to be set to minimum 50 |  |
| 5 | BPM | 2 | 150 | BPM set to 120 |  |
| 6 | BPM | 2 | “wasd” | BPM set to 50 |  |
| 7 | Increasing BPM | 2 | 60 BPM press + 5 | BPM set to 55 |  |
| 8 | Decreasing BPM | 2 | 60 BPM press – 5 | BPM set to 65 |  |
| 9 | Changing BPM after 2 mins | 3 | The BPM is set to change to 120 BPM after 2 mins | The BPM is changed to 120 BPM after 2 mins |  |
| 10 | Audio test | 4 | Listen for the audible beep at each tick | Hear an audible tick |  |
| 11 | Visual test | 5 | Look for a visual cue at each beat | See a visual cue |  |
| 12 | Light / dark mode test | 6 | Press Light / dark mode button | Change in colour scheme of page |  |
| 13 | Turn on the metronome | 7 | Press the start / stop button | The metronome starts ticking |  |