Metronome

# Metronome

## Feasibility Study

We will be creating a web-based metronome. A metronome is a device which ticks at regular intervals, and is used by musicians to keep track of their timing

This problem is solvable as it has already been done several times on the internet, and it can be solved in a finite number of steps. The main calculation is converting a BPM to the delay between each beep. This is computationally simple using a theoretical approach.

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

## Analysis

### Stakeholders

Hermit the Frog is a frog who is practicing the drums. He seems to have trouble keeping in time because he can’t reach the kick drum because his legs are too small. He wants to have a metronome which can keep the time for him (simulating a kick drum) to bide by until his legs grow long enough to reach the kick drums

### Essential features

My metronome must allow the stakeholder to choose the beat per minute counter, and start/stop the beat. On each beat it should make an audible tick sound, as well as giving a visual indication of when the beat is. It should show the musical tempo name (allegro, presto, lento, etc.), and the text must be readable for anyone. A slider will allow you to set the beats per minute manually, allowing any song tempo to be played on the metronome.

### Limitations

There is insufficient time to add additional features, however they can be implemented later on. The solution will not be able to detect the beat of the music by listening to a piece of music and adjusting the BPM to the time of the music.

The metronome is purely web-based, and as a result there is no need to develop it natively for Android/iOS devices.

### Hardware/Software requirement

For development, Visual Studio 2017 is required. This requires at a minimum:

* >1.8GHz processor
* 2GB Minimum, 4GB recommended
* >130GB Hard Drive space

The app will also use Chrome. This is to ensure that all features of HTML/CSS/JavaScript work as intended

| **Platform** | **Version** | **Release Date** |
| --- | --- | --- |
| Chrome on **Windows** | 109.0.5414.74 | 2023-01-11 |
| Chrome on **macOS** | 109.0.5414.74 | 2023-01-11 |
| Chrome on **Linux** | 109.0.5414.74 | 2023-01-11 |
| Chrome on **Android** | 108.0.5359.128 | 2022-12-14 |
| Chrome on **iOS** | 109.0.5414.83 | 2023-01-11 |

### Success criteria

|  |  |
| --- | --- |
| Criteria | Justification |
| Must be accessible to a Chrome browser on a PC which is connected to the internet | The stakeholder has a Windows PC with Chrome on it with a Wi-Fi connection. Any update can be distributed to the user easily. |
| The user can set the BPM | The stakeholder wishes to practice between 50BPM and 150BPM |
| Any invalid BPM will be automatically corrected to the closest sensible value between 50-150BPM | The stakeholder wishes to focus on drumming and may accidentally input invalid data |
| There is a start and stop button that toggles playing when pressed | The user should be able to control the metronome by pressing the button on the website, or through the spacebar using a simple UI |
| When the metronome starts, there should be an audible beep and visual cue | The stakeholder needs to drum along a to a click sound produced by the metronome. |
| There should be a visual cue when each beat is played | So, the stakeholder can use the metronome without audio |

### Research

|  |  |  |  |
| --- | --- | --- | --- |
| App | Examples | Advantages | Disadvantages |
| Mobile | SoundBrenner's Metronome (AppStore) | + Mobile - can take with you anywhere on the go, and can be used anywhere for whatever purpose (more convenient)  + Usually does not cost any money at all | - If your phone runs out of battery, you cannot use the metronome anymore |
| Web-based | Metronome Online (metronomeonline.com)  Metronome Click (metronome.click) | + Usually does not cost any money  + Can be taken anywhere with an internet connection | - Requires an internet connection in order to work, so it is not a definitive solution |
| Physical Mechanical | Donner Mechanical Metronome  Martisan Mechanical Metronome | + Does not require any power/battery in order to operate normally | - Prone to breaking, rendering it unusable if exposed to damage  - Costs money, and it may cost more than other paid solutions. |
| Physical Digital | Lekato Digital Metronome  Sondery Digital Metronome | + Portable, and does not require an internet connection to work  + Batteries that come with the product are usually replaceable and last a long time | - Requires a battery in order to operate, which will run out, so it requires a power source to operate, |
| Paid | Metronome Pro by ONYX 3  Moises Pro Metronome by Moises Systems | + Usually higher quality than free products, with more functions to boot (drums beats, songmaking) | - Sometimes, it may not have any more functions than a free product  - You have to pay to gain (full) access to the product, depending on if it is a trail product |
| Free | Musicaa Metronome (musicca.com/metronome)  SoundGrail Metronome (soundgrail.com/metronome) | + Does not cost any money to use | - May not be as high quality product as a paid product  - Usually has advertisements which may irritate the consumer |

## Design

### Design Heirarchy chart

### Algorithms

BPM = USER INPUT

INTERVAL = 1\*60\*1000/bpm

The user should be able to enter the BPM score. The browser needs to know how long to pause between each tick in milliseconds. This can be done using this algorithm.

### Usability Features

84 BPM

Moderato

50

150

START/STOP

SETTINGS

The Interface is simple, and it is easily understandable what to do. The settings button allows the customization of the interface, allowing the user to change the colour scheme from a selection of black, white, or high contrast. The white mode is not fully white so it is easy on the eyes. A slider allows the user to change the BPM level from 50 to 150

## Implementation

## Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test number | Description | Success Criteria | Test Data | Expected Result |
| 1a | Basic Web Page | 1 | Page load | A web page with a title and text box to enter the BPM. Displays in Chrome |
| 1b | BPM | 2 | 120 | Valid data accepted |
| 1c | BPM | 3 | “” | BPM set to 50 |
| 1d | BPM | 3 | 40 | BPM set to 50 |
| 1e | BPM | 3 | 160 | BPM set to 150 |
| 1f | BPM | 3 | “Idk what I’m doing anymore” | BPM set to 50 |
| 1g | BPM | 3 |  |  |

### Variables and validation

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Data Type | Validation | Justification |
| BPM | Integer | Is between 50-150 | The stakeholder requires the BPM be between 50-150 BPM |
| BPM |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### post development testing

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

#### Functionality

What did you want the program to do? Does the program fulfil these needs? What did you click on in what order? What happened? Were there any features that you wish were there but aren’t?

#### robustness

Did the program crash? If so, when? What did you put in to make it crash?

## Installation

## Evaluation

## Maintainence