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 a modern browser e.g., Chrome, Firefox, Safari, Edge. This is to ensure that all features of HTML/CSS/Javascript work as intended

|  |  |
| --- | --- |
| Firefox (Linux) | 108.0.2 |
|  |  |
|  |  |
|  |  |

### Success criteria

|  |  |
| --- | --- |
| Criteria | Justification |
| Must be accessible to a Firefox browser on a PC which is connected to the internet | The stakeholder has a Linux PC with Firefox 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 |

## 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, such as the colour scheme

## Implementation

## Testing

## Installation

## Evaluation

## Maintainence