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

## 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 |
|  |  |  |
|  |  |  |
|  |  |  |

# 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 10 in seconds

60