COMP 3008 – Metronome Assignment

Assignment 3 – Due Thursday Match 7th

Group 2

Danny Luu

Chris Cowan

Bjørn Vårdal

Bora Sabuncu

Table of Contents

Installing Instructions

1. Before running our program, you must have the Microsoft .NET framework 4.5 installed
2. Next locate the zip folder where you have downloaded our assignment
3. Extract the zip file into a directory where you will remember
4. Open the folder Metronome App and the necessary files needed to run the program will be there.

Running Instructions

After following the steps in the Installation Instructions, open the folder *Metronome App* and double-click *Metronome Program.exe* to run the metronome app and use it.

Note:

- The folder *MetronomeWPF* contains all of our code and images we used in the program
    - In order to open and run it in debug mode, it needs VS2012 already installed along with the Microsoft .NET 4.5 framework to actually run in debug mode

Software Details

A metronome is used generally by musicians to keep track of the beat by feedback of sound or visual cues. It provides this feedback to the user by playing a sound of a steady pulse within the given tempo and given note length (whole, half, quarter, eighth and sixteenth). There are various versions of metronome software where all of them have this functionality. The Users have the functionality of changing the tempo along and the time signature. Different implementations come from different ways to change these variables and sometimes offer extra functionalities and usability.

Our implementation of a metronome includes all the requirements listed in Assignment 3 [MR-1 to MR-17] along with the open ended extra functionalities [MR-18 to MR-20], that it would be useful to have but wasn't specifically listed as a specific requirement by the assignment. One idea we had was to give the user ability to select which beats to emphasize. This would be useful if the user is practising on a genre's traditional rhythm pattern. Such examples for this would be Bossa Nova, Reggae and some odd time signature like 5/4. Some rhythms are hard to get at first couple of tries, so we thought this function will help users understand them by making it easier to set up. Metronome users sometimes have trouble hearing the metronome clicks for various reasons like the click sound is too similar to what they are playing or because of the environment. We thought it would be useful if the user could pick the click sound from various sounds like a cowbell or even to the generic beep. This way user could pick the sound that will be easy to hear through the instrument being played. Lastly we added so that user could change the beats per bar allowing the user to use the metronome with different time signatures. The open ended functionalities we have provided for MR-18 to MR-20 was the ability to have the Metronome live update the tempo click sounds as the user adds, changes or slides the tempo bar. The next one was to include the addition of an emphasis beat and lastly the ability for the app to be dragged to portray a landscape or slimmed to portrait view.

Software Design and Decisions

For our software design, each one of our group members had our own original designs with some good aspects that we wanted in our metronome. As a group was we got together and collaborated for a couple hours on what we believe was the best UI possible for our metronome (**Figure 1**). Before this we took into account each one of our design implementation of the metronomes and gathered information on what would be the best functions, design implementation and GUI designs that would make our metronome successful. The other **Figures** will show the designs of our group members' original metronome and what influenced our current design we have now.

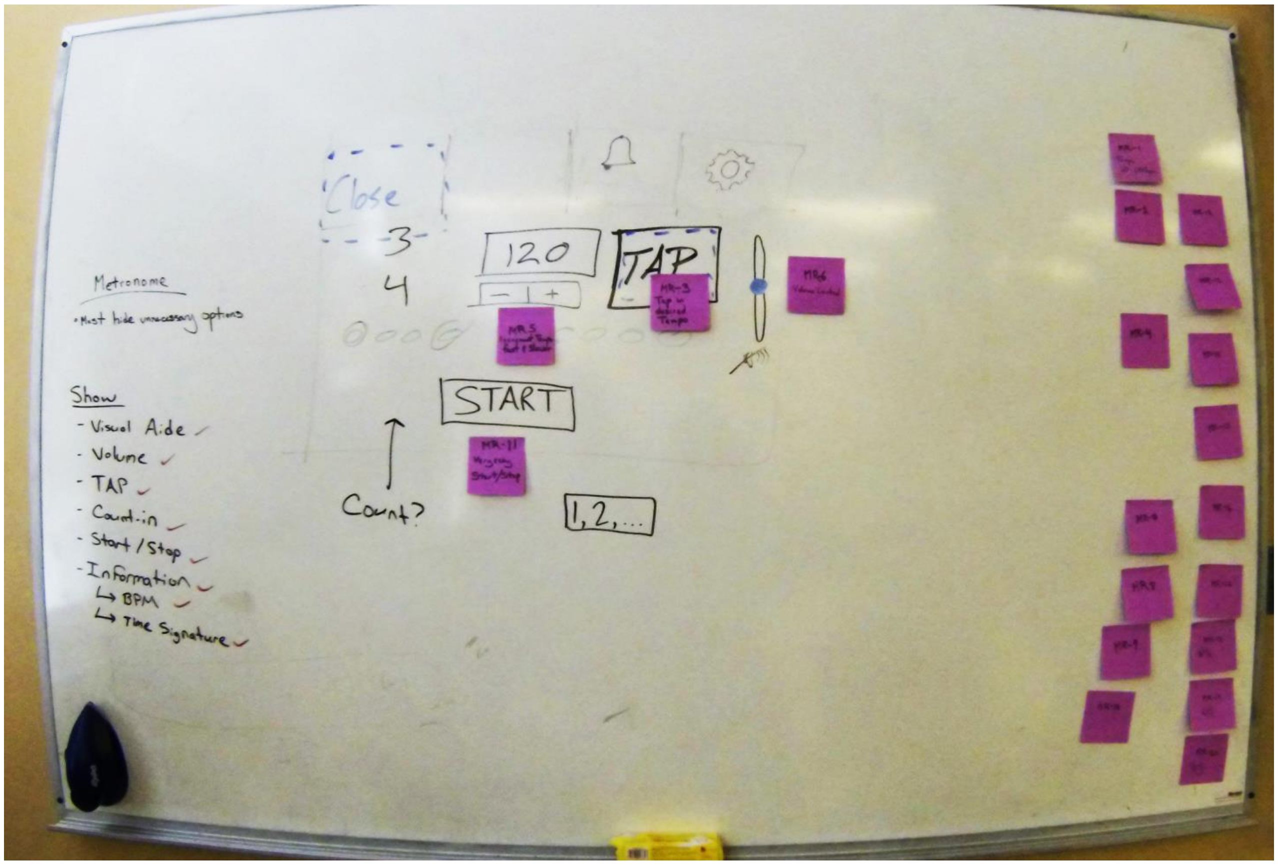


Figure 1 – Group Metronome Design Sketch

This was what came out of our first group meeting on designing a pseudo metronome. While some of the implementation shown here didn't make it to our final design, it gave us a good base on what we needed to implement.

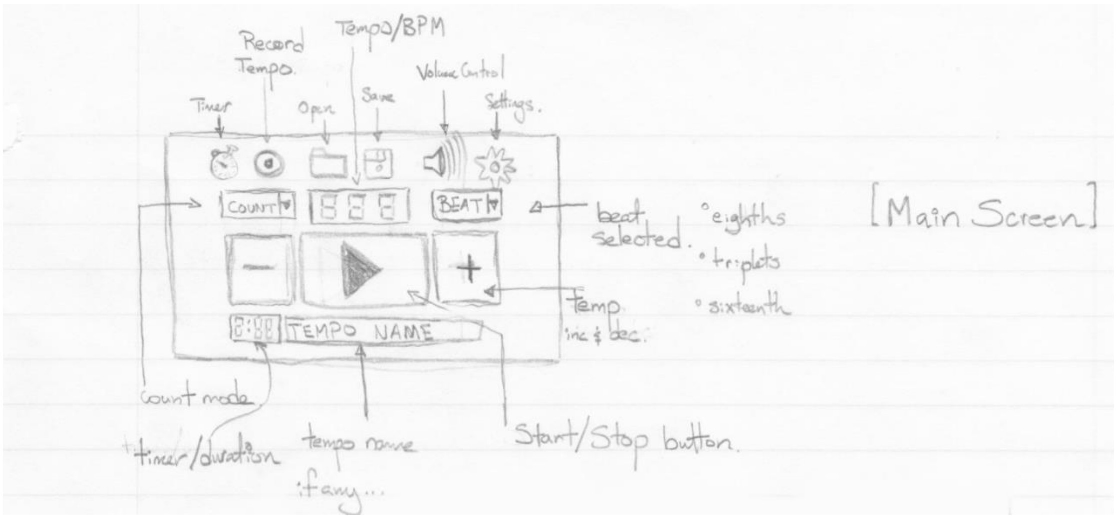


Figure 2 – Danny's Main Screen

This was from our group member Danny; we looked at his menu bar and really liked how he encapsulated some of the unnecessary options that were cluttering the other metronomes and took this idea into our design.

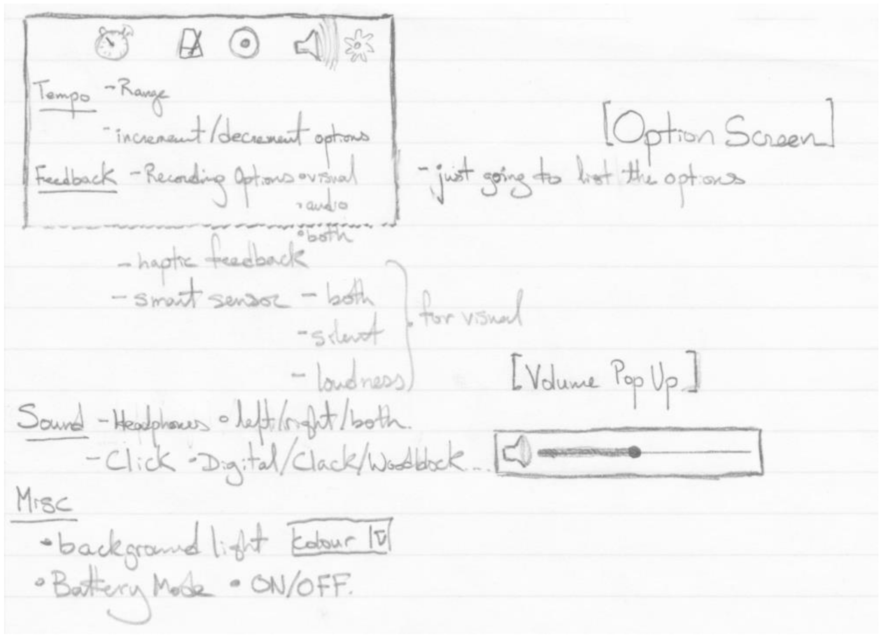
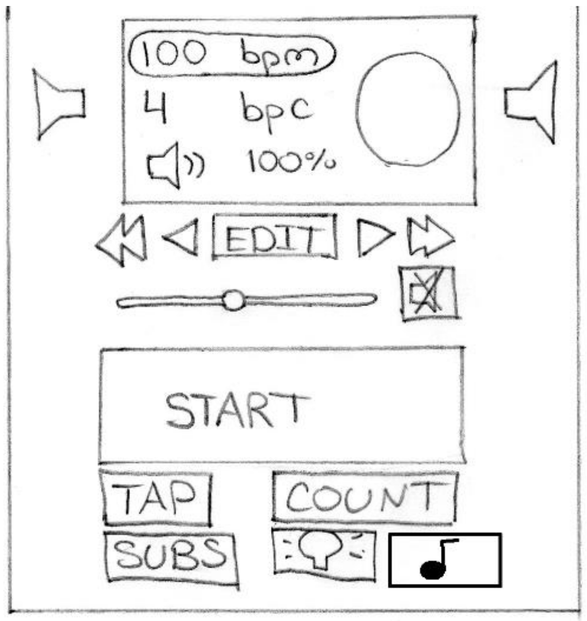

Tempo - Range
- increment/decrement options
Feedback - Recording Options · visual · audio · both
[Option Screen]
- just going to list the options
- haptic feedback
- smart sensor - both - silent - loudness
for visual
[Volume Pop Up]
Sound - Headphones · left/right/both.
- Click - Digital/Clack/Woodblock..
Misc
· background light Colour
· Battery Mode · ON/OFF.

Figure 3 – Danny's Option Screen

This was what one of our group member wanted to display the options for changing some option required from the metronome but as we started making the implementation, the need for an Option/Setting screen became less and less desirable as the number of functions that needed to be changed by the metronome would be easier to do just on the main application itself. We wanted to do this because we believed as a group that going back into the settings to change an option that could easily be implemented on the main screen would reduce the need to always visit this screen. Thus reducing the need to always visit this page and make changes to the metronome cumbersome.



* START changes to STOP when the app is running
* COUNT is highlighted when activated

Figure 4 – Chris's Main Screen of the metronome

We started to see a trend in our designs with how information would be displayed and took this idea into making the display of the BPM centered on the screen. We also saw a new feature that would make our metronome even easier to change the BPM seamlessly, that was with the addition of single and bulk increment buttons on the sides of the BPM display. The best feature we extracted from Chris' design was the slider bar along the bottom of the BPM display where a User could in theory just slide the bar to a desired BPM and from there either select the textbox or buttons to change the metronome to the desired BPM.

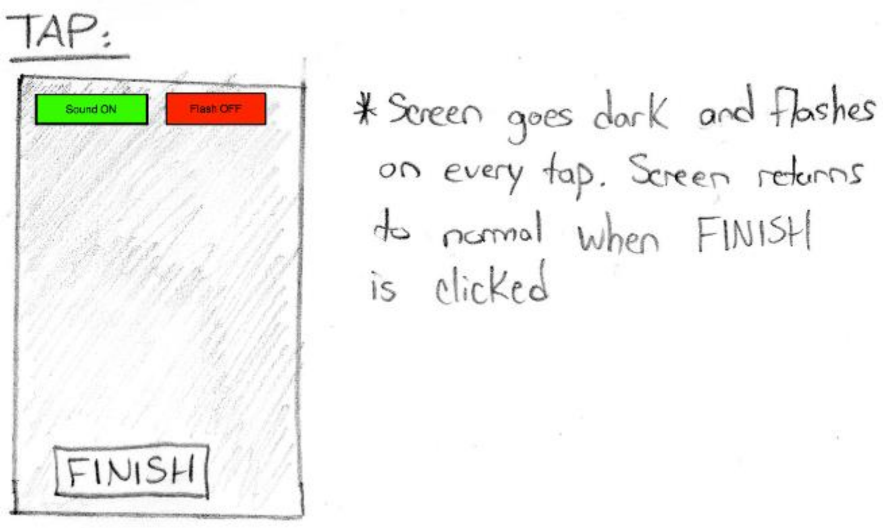

TAP:
Sound ON
Flash OFF
FINISH
* Screen goes dark and flashes on every tap. Screen returns to normal when FINISH is clicked

Figure 5 – Chris' Tapping Implementation sketch

This sketch was by far the best implementation in how we wanted to implement the tapping feature of capturing our user BPMs. This design was simple elegant and reduced unnecessary clutter of information that would confuse the users of our metronome.

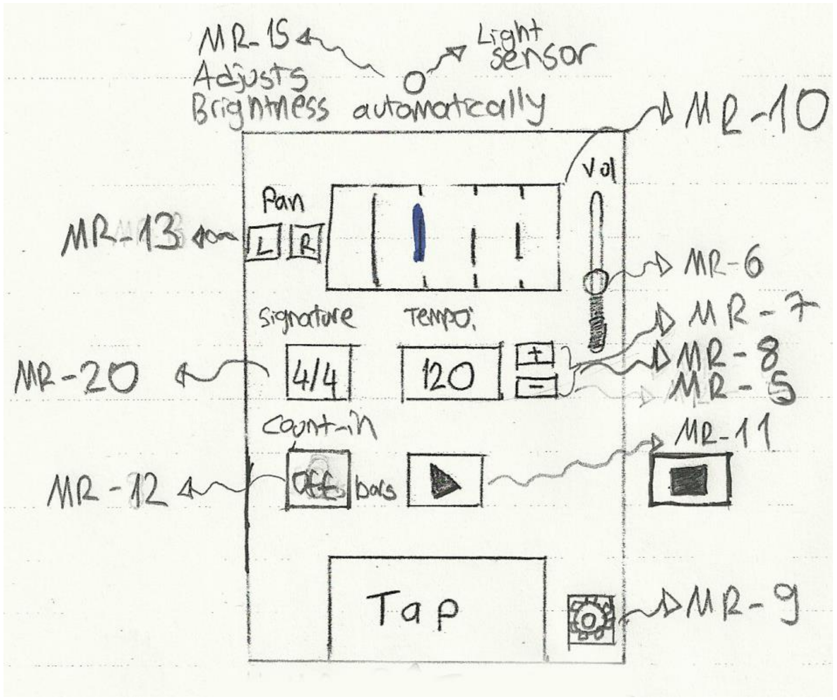


Figure 6 – Bora's Main Screen Sketch

This was from our other member of our group whose design we looked at and took the slider idea of volume as an onscreen button. Originally we were going to give users the ability to click a sound icon that would bring up a screen where the user would then slide the volume to the desired level but discarded that idea very quickly. The reason we implemented the slider onto the main screen was for the main reason we didn't implement the settings button on the top if the menu bar, to reduce complexity of just changing a single attribute of the metronome and make it easily accessible to the user of the program.

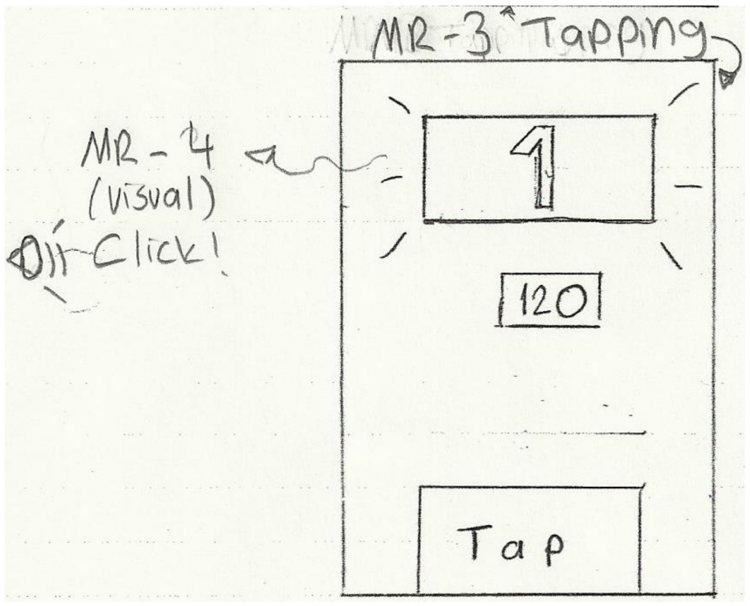


Figure 7 – Bora's Tapping Implementation Sketch

Like **Figure 5**, we saw that this was the best implementation on how to approach the tapping function of our metronome. We started to see a pattern with using a button for the tapping but realized that users aren't even going to be looking at the application when tapping or maybe they would and shouldn't need to focus on the area of tapping in order to record the taps. So we modified **Figure 5** & **Figure 7** by making the whole screen able to tap and when the user has finished tapping their desired tempo, they would just have to select an ok button or cancel button to capture their inputs and place it within the main metronome to use.

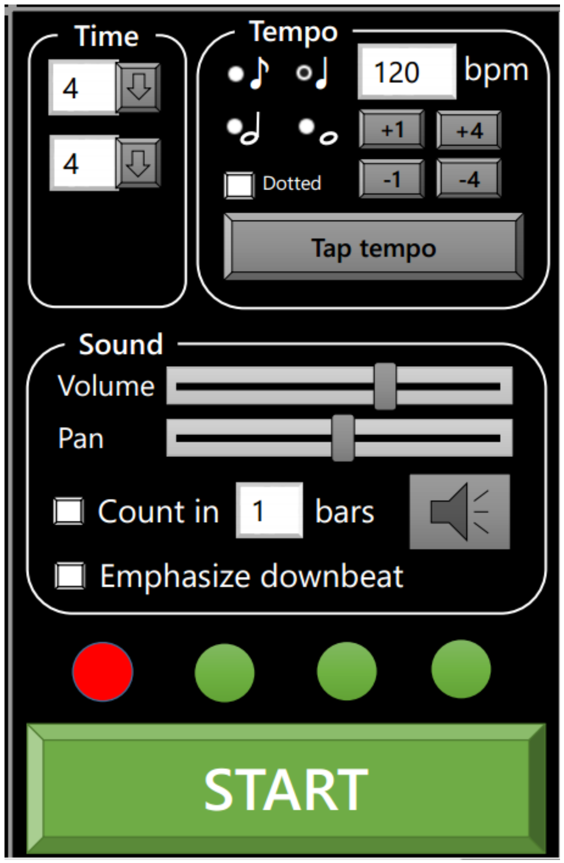


Figure 8 – Bjørn's Main Screen Metronome Design

This design was well done and what constituted most of our design in our metronome. We included the emphasized down beat coloured circle implementations and took on the idea of having a very large start and stop button on the very bottom of the screen of the metronome application. We placed the start/stop button along the bottom to fulfill the requirements of easily making the user start and stop the beats without hindering the view of the applications main functionalities itself. This was also placed along the bottom to take into account that users cannot miss-click a function accidentally changing the metronomes beat, sound, etc. What we left out was the beat length from this design because we felt that it would have cluttered our application more than needed. Even though this beat length would have been an amazing addition we felt that the design of our metronome needed to be simple and easy to pick-up and use

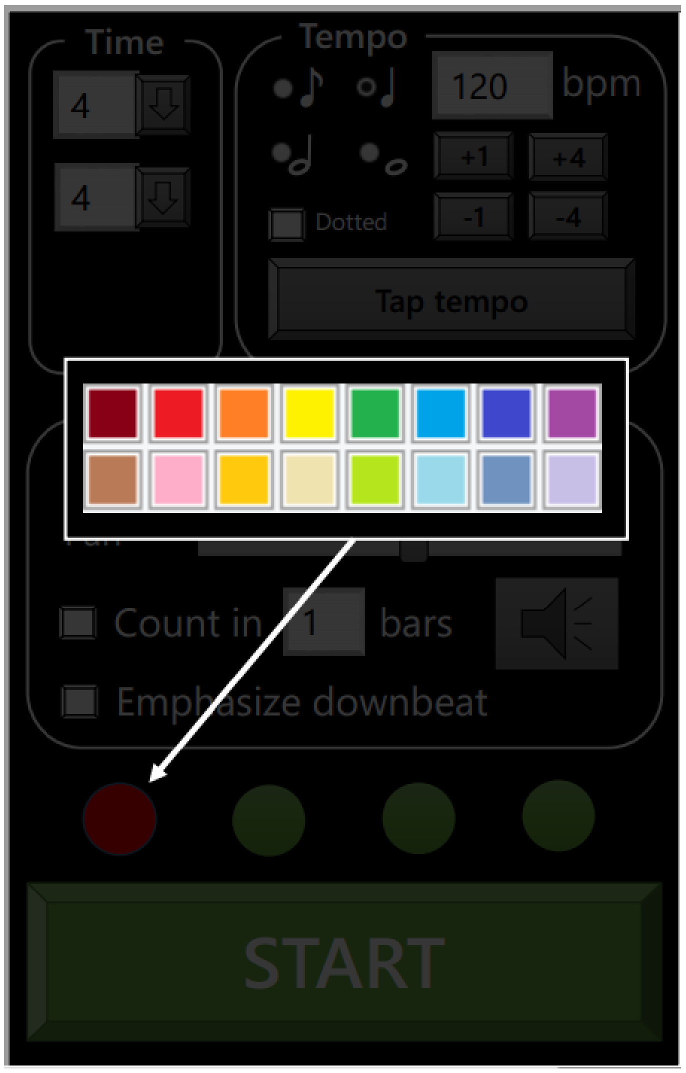


Figure 9 - Bjørn Vårdal colour scheme for displaying beats

This was by far the best solution for approaching the lights to display as a feedback to the users of our metronome on a stage with coloured stage lights. We implemented this function slightly differently but with the same idea in mind. What we did was allow the users to click on the colour wheel on the actual metronome program and select what colours they wanted with which beat to display it as. Our metronome handles 3 types of beats, on, off and emphasizes.