Assessment 2

Keniel Dacosta

###### MSc Audio Technology

###### 7MU007 Musical Human Computer Interactions 2019/20

###### For the attention of: Mat Dalgleish

# INTRODUCTION

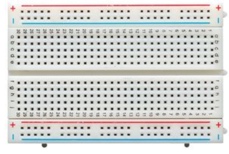
The idea behind this project is to create a children’s musical learning-based game for ages 4-8. It is based around timing with the player having to play along to the sequence in time in order to progress onto the next level. Each level advances in complexity, thus engaging with the players listening skills, timing, and developing rhythm. It is a hardware/software game.

# What you will need

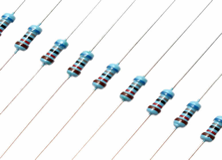
1 x Arduino Board



1 x Solderless Breadboard



3 x 1Mom Resistor



3 x Momentary Push To Make Buttons -(Red, Yellow, Green)



1 x USB 2.0 cable, Type A / B



6 x Solderless (male to male) Breadboard Jumper Cables



# How to build your own – (Configuring the hardware)

Step 1:

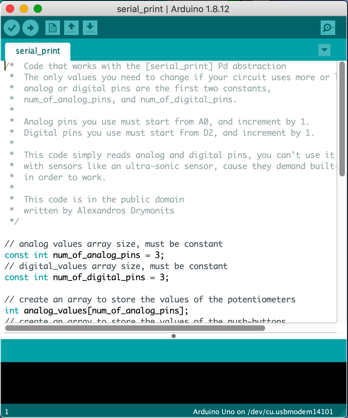
Download and install, then open Arduino IDE on your computer. (Software can be found at: <https://www.arduino.cc/en/Main/Software>)

Step 2:



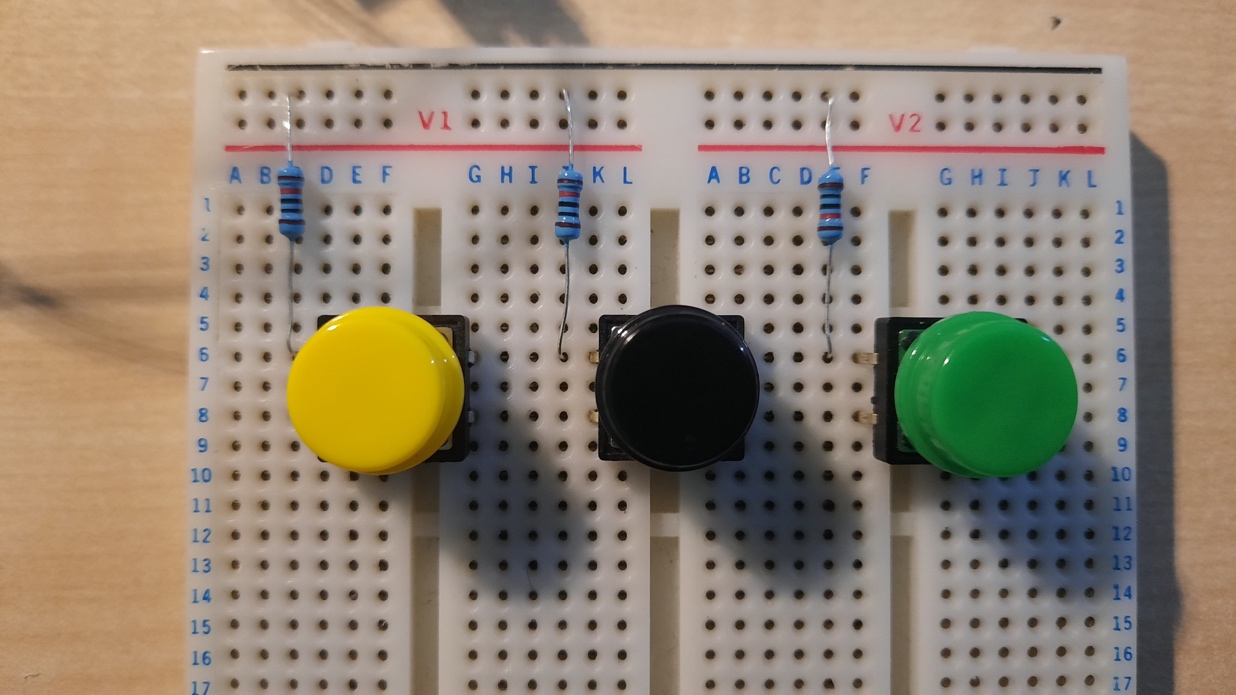
Connect Arduino board to the computer using the USB cable (Type A/B)

Step 3:



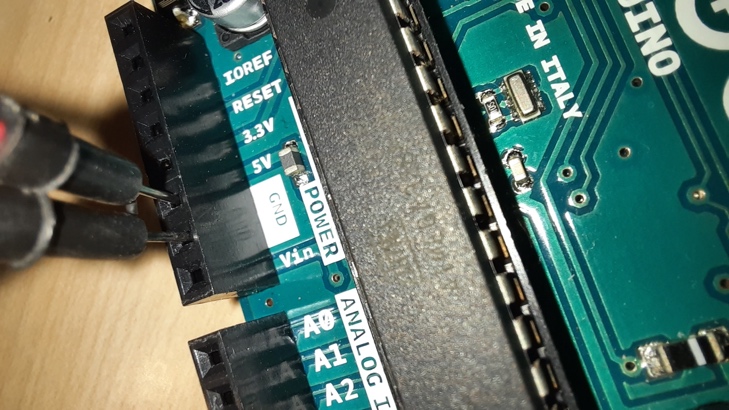
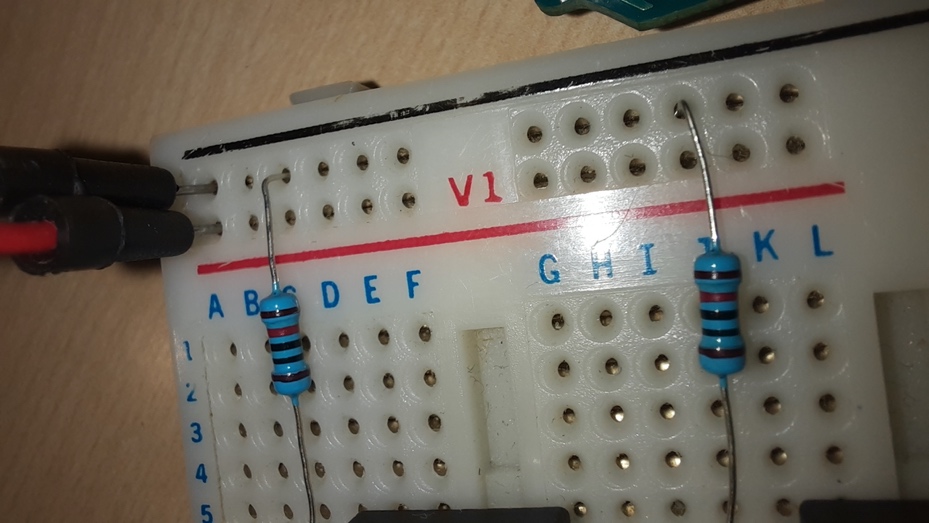
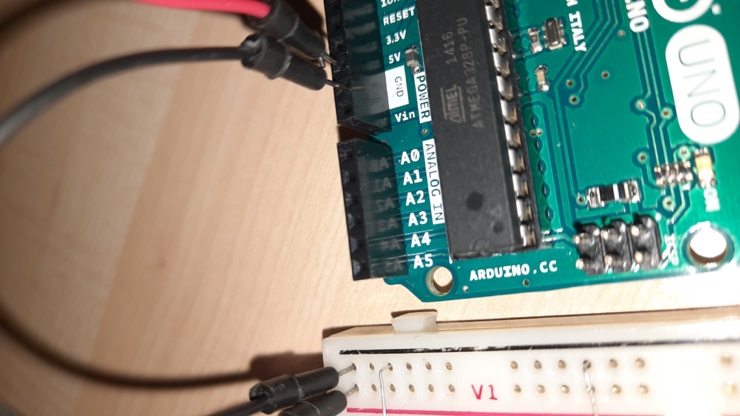
Configure Arduino by uploading Serial\_Print.ino to the board

Step 4:



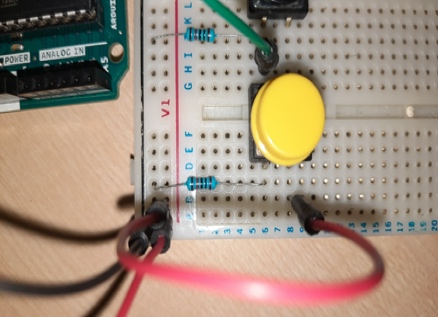
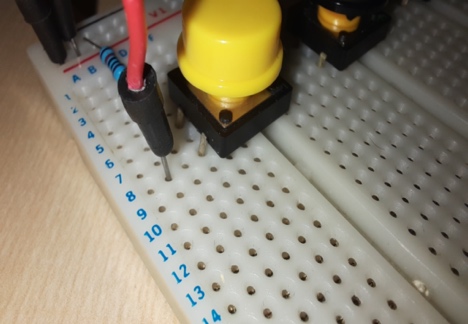
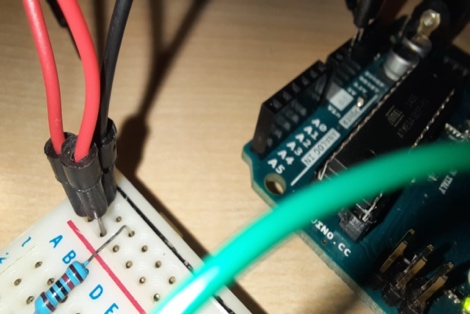
Connect the Momentary push buttons to the Breadboard and the 1mom resistors in the sequence as displayed

Step 5:

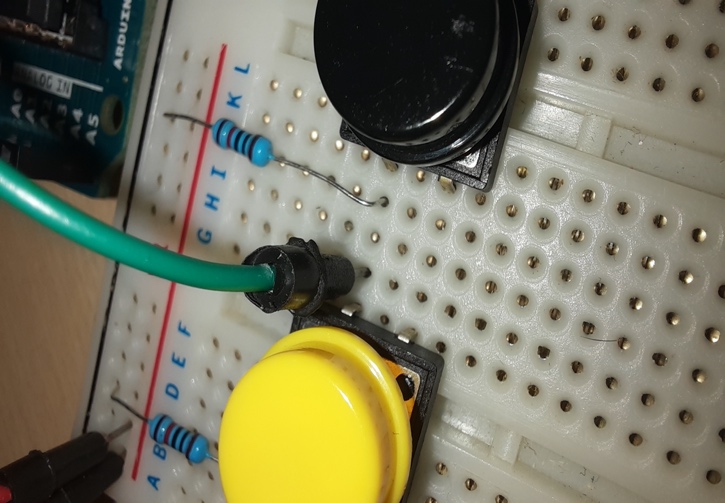
  

Connect the ground (GND) from the Arduino to the negative (black) strip on the breadboard using solderless male to male jumper cables. Connect the power (5V) from the Arduino to the positive (red) strip on the breadboard.

Step 6:

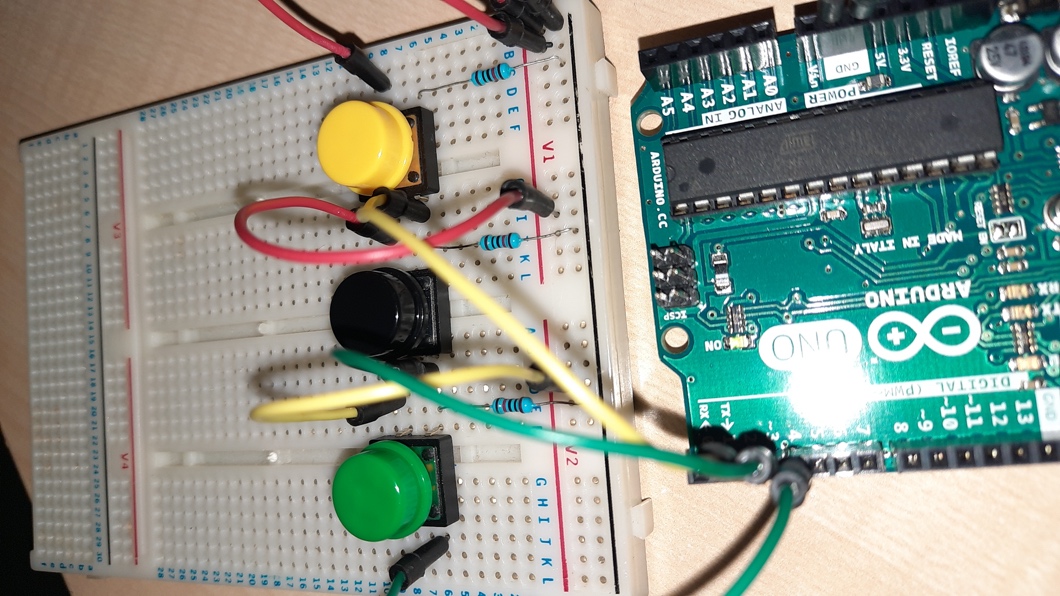


Insert a jumper cable next to the jumper cable carrying the 5V from the Arduino to the breadboard. Insert the other end of the jumper cable next to the Momentary button as displayed.

Step 7:

  
connect a jumper cable next to the momentary button in order to carry the signal from the button to the analogue input of the Arduino. Repeat this step for the other two buttons.

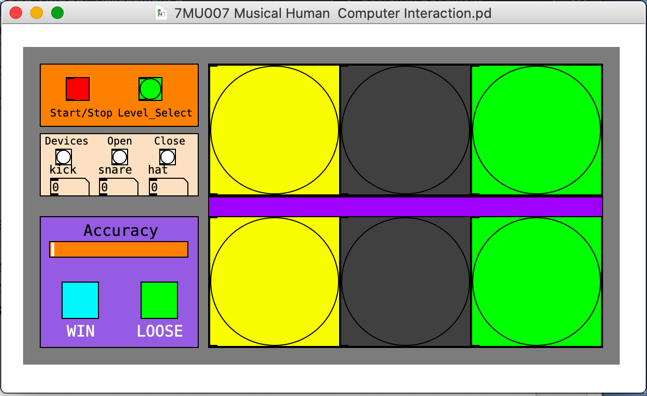
Step 8:



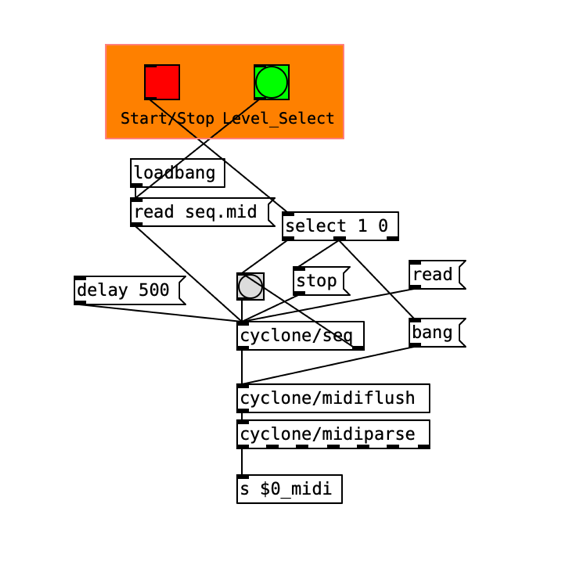
The final setup should look like the image displayed with all the momentary buttons supplied with 5V and a analogue output connected to the Arduino

# 

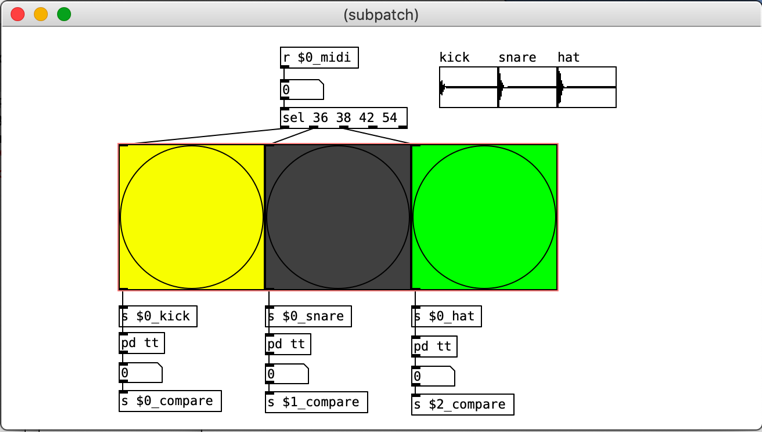
# How the Patch works



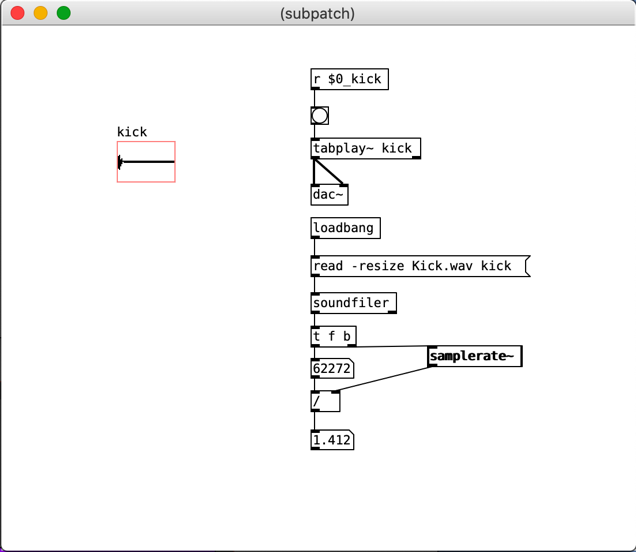
The main graphic user interface is designed with the particular user in mind which is ages 4-7. Its visually pleasing for this purpose and must conveys as much information visually. The colours of the buttons correlate with the physical buttons. The user can see what’s being played by themselves and the computer. As the game is focused around timing and mimicking the drum sequences, this is important for the user to interpret on a visual level and an audio. The yellow box represents the Kick, the black box represents the snare and the green representing the hihat.



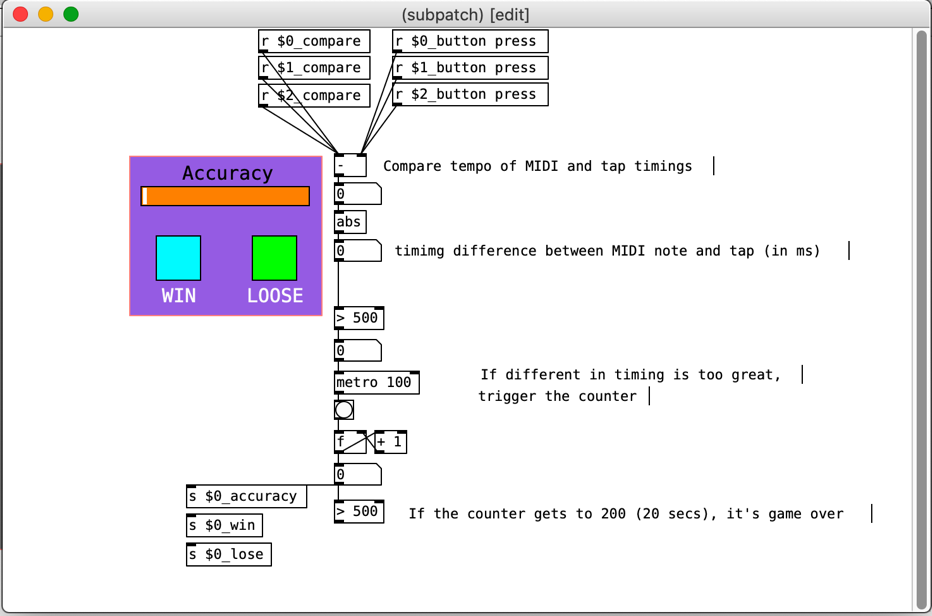
From the section of the patch the user can select the midi file with the level of complexity ranging from 1 to 3. The user can also start and stop the sequence. The midi file is loaded and read

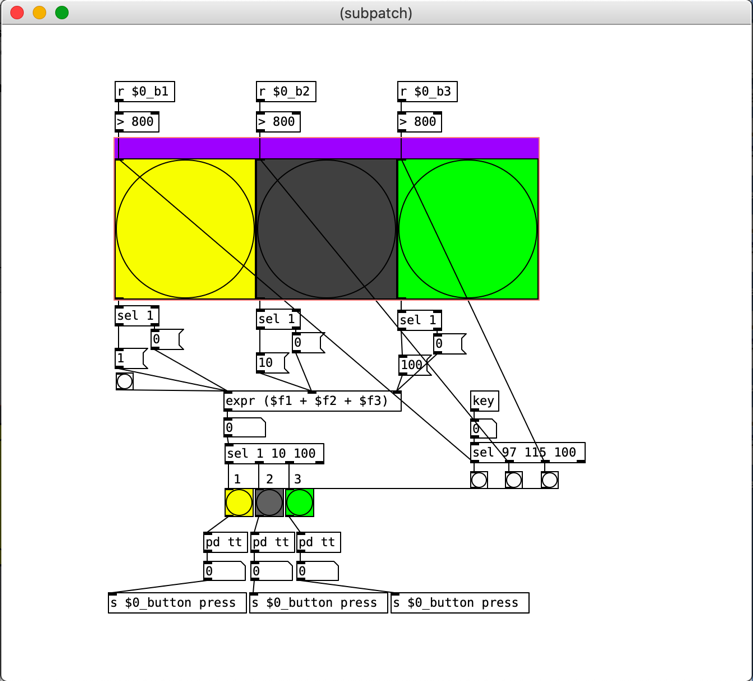


The midi notes within the midi file are then received remotely, are then shown as bangs each box representing either the kick snare and hi-hat.

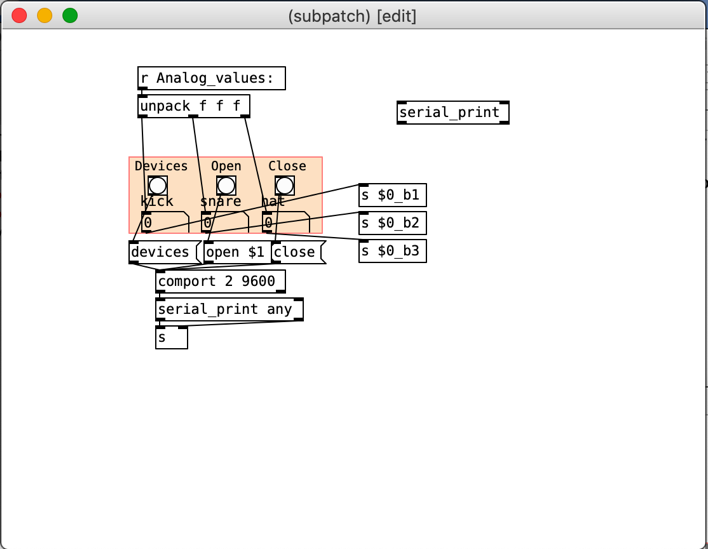


The audio samples are triggered when they receive a bang from their corresponding colour. This allows the user to hear the sequence and see it in the flashing of the bangs. the timing information is then compared with the timing information of the user and the difference is used to then give the user a visual feedback via the Accuracy bar.





The users physical input from the buttons pressed, trigger the corresponding bangs and it can be seen by the user so the timing is based around the user seeing that the buttons are pressed in time, so there is no audio triggered when the user presses a button it is only convey visual feedback. The timing information of the user is compared to the computer.



When the users timing falls out of time by 500ms it will trigger the metronome to count down for 20 secs and then display a x in the loose box if the user maintains timing for 20 secs then an x is displayed in the win box

# REFERENCES