**Application Parameters**

Before the application starts, get user to enter colours and mp3 file name and allow them to either run with default values or customise, in which case a new window appears with sliders set at the default values, allowing the user to customise the following values within the ranges given.

Files

Amplitude.java

**Amplitude Buffer Size**

* The larger this value is, the longer amount of time considered when looking for the previous minimum and maximum amplitude
* A larger value will work better for songs with a consistent volume throughout, but will result in little change in the size of the visualisation for periods of sustained amplitude
* A smaller value will work better for songs with sections of vastly varying amplitude, but may result in vast size changes for periods of similar amplitude
* Allow user to give a value in terms of seconds, which can be multiplied by the fps to pass into the function
* Range 1 – 60?
* Default = 10

**Amplitude Percentage Buffer Size**

* The larger this value is, the smoother the visualisation changes will be
* Range 0.2\*FPS - FPS

**Amplitude Minimum Size**

* The smallest percentage of the full visualisation size that the visualisation can take up
* Smaller values may result in very small visualisations for low amplitudes
* Larger values mean that there will be less variation in the visualisation size with changing amplitude
* Range 0 – 100
* Default = 20

Pixel Buffer Size

* NON-CUSTOMISABLE - based on window size

BPM.java

Amplitude Buffer Size

* NON-CUSTOMISABLE - based on min and max theoretical BPMs likely to encounter

**BPM Buffer Size**

* The larger this value, the more accurate the BPM will be, but will take a long time to adjust after changes in the BPM
* Choose a larger value for songs where the BPM is \*mostly\* consistent, a smaller value for songs where the BPM changes frequently
* Allow user to give the value as an integer
* Range 20 – 100?
* Default = 30

Key.java

**Key Buffer Size**

* The larger this value, the more accurate the key will be, but will take a long time to adjust after changes in the key
* Choose a larger value for songs where the key is \*mostly\* consistent, a smaller value for songs where the key changes frequently
* Allow user to give a value in terms of seconds, which can be multiplied by the fps to pass into the function
* Range 5 – 30?
* Default = 10

Other

**FPS**

* The larger this value, the slower the program will run, or it may fail completely if the processor speed is too low.
* The smaller the value, the less accurate the program will be, especially with songs where the music changes very quickly
* Allow user to give raw value for FPS, but should be a multiple of 10
* Range 10 – 120?
* Default = 30

**Width**

* The width of the window that the program runs in
* This affects the size of the visualisations

**Height**

* The height of the window that the program runs in
* This affects the size of the visualisations