

```

#####
## Kandinsky Visualizer Ver 0.1
## 5th July 2018
## Author: Mahtab ES Email: mahieelzz@gmail.com
## Descriptions: This Program Read a Wave music file from Local Hard Drive
## and Visualize the fourier transformation processed music with Kandinsky-like output and animate them.
## I try to make a Web app with ShinyR so the user can interact with the prameters
#####

### Install these package and call them for future use
## for building animation Imagemagick must be installed

library(kandinsky)
library(tuneR)
library(sound)

#Setting up address where the source music is
target_address <- "/home/mahtab/Music/test.wav"
## These are parameters that can be setup by end user
## 1. step1 : is the amount of each batch we wish to use for each image
step1 <- 30

## 2. slider1 : A Vector where define where are the 3.start and 4.end point of wave file
slider1 <- c(1,2)

start1 <- slider1[1]
end1 <- slider1[2]

## Fourier Transformation of wave file
myfourier1 <- readWave(target_address, from = start1 , to = end1 , units = "seconds" )

#Parse Left Channel and Right Channel
LeftChannel <- myfourier1@left
RightChannel <- myfourier1@right

#Calculation of average of total channel amplitude
CenterChannel <- (RightChannel + LeftChannel)/2

# Plot the time series of fourier transformation of wave file and see if it's read correctly
ts.plot(CenterChannel)

# Define Indexer for crawling over the curve in the next loop
indexer1 <- seq( from = 1, to = 44100 , by = step1)
# adding end point to the indexer (seq() function doesnt have the extreme value)
indexer1 <- c(indexer1, 44100)

# Slice the Curve and use each batch(slice) for a unique kandinsky-like image
for(i in 1:(length(CenterChannel)/step1)){
  KK<-CenterChannel[(indexer1[i]:indexer1[i+1])]
  kandinsky(KK)
  #save the file of each batch to address where we can finally animate them
  dev.copy(jpeg, paste("/home/mahtab/Pictures/exports/",i,".jpg",sep = ""))
  dev.off()
}

#animate the images in the folder using ffmpeg
system("cd /home/mahtab/Pictures/exports;ffmpeg -framerate 9 -pattern_type glob -i '*.jpg' -r 30 out4.mp4")

```