JMotion Functions

John Westbrook

4/14/25

# syncfps()

### desc: determine speed of frame by frame animation to match bpm of song for a rythmic visual

### paremters: duration of animation loop, BPM of song, beats per animation loop, FPS of animation

### output: new speed of animation (percentage)

#example  
  
x=15 #frames in each loop  
y=177 #song BPM  
z=2 #beats per loop  
n=24 #FPS  
  
#formula  
  
newanim <- (100 \* x \* y) / (60 \* z \* n)  
  
if(newanim < 100){  
 paste('Decrease speed of animation to',round(newanim,1),'%')  
} else if(newanim > 100) {  
 paste('Increase speed of animation to',round(newanim,1),'%')  
} else {  
 print('No change')  
 }

## [1] "Decrease speed of animation to 92.2 %"

# syncbpm()

### desc: determine bpm of song to match fps of frame by frame animation for rythmic visual

### paremters: duration of animation loop, BPM of song, beats per animation loop, FPS of animation

### output: new speed of song (percentage)

#example  
  
x=15 #frames in each loop  
y=177 #song BPM  
z=2 #beats per loop  
n=24 #FPS  
  
#formula  
  
newsong <- (6000 \* z \* n) / (x \* y)  
  
if(newsong < 100){  
 paste('Decrease speed of music to',round(newsong,1),'%')  
} else if(newsong > 100) {  
 paste('Increase speed of music to',round(newsong,1),'%')  
} else {  
 print('No change')  
 }

## [1] "Increase speed of music to 108.5 %"

# checktiming()

### desc: determine frame timing of a gif

### parameters: .gif file

### output: prompt stating “Image changes every [value] frames”

#pseudo code  
  
#gif <- import gif to analyze  
#Locate block for frame delay (in centiseconds)  
#determine fps  
#calculate timing  
  
print("Image changes every ... frames")

## [1] "Image changes every ... frames"

# bouncin()

### desc: determine coordinate path of object for a parabolic bounce

### input: width of canvas, height of canvas, starting position x, starting position y, frames

### output: generated data frame of x & y coordinates for each frame of moving object

#example  
  
w=1920 #canvas width in pixels  
h=1080 #canvas height in pixels  
x=50 #starting x  
y=75 #starting y  
f=10 #frame duration  
  
#pseudo code  
  
#initialize canvas and starting position  
#based on # of frames, map out coordinates that remain within dimensions  
#generate dataframe  
  
#parabola must be relative to canvas dimensions