Lab 09

CSCI 110-02 Shifting between Media types October 21, 2013

Name: _____

For this lab, the mediasources directory needs to be on your computer or USB drive where it can be written to.

1. We are going to take the sound in a file and write it out as numbers to a file.

```
def soundToText(sound, filename):
   file=open(filename,"wt")
   for s in getSamples(sound):
     file.write(str(getSampleValue(s)) + "\n")
   file.close()
```

   In your media sources on your USB drive create a new file called soundText.txt. At the command line type

```
>>> setMediaPath()  ## set it to the mediasources directory
>>> file=pickAFile()
>>> sound=makeSound(file)
>>> soundToText(sound,getMediaPath("soundText.txt"))
```

   Now look at your soundText.txt file. Is it full of numbers?

2. Now, lets take those numbers and make a sound from them.

```
def textToSound(filename):
   sound=makeSound(getMediaPath("sec3silence.wav"))
   soundIndex=0
   file=open(filename,"rt")
   contents=file.readlines()
   file.close()
   fileIndex=0
   while(soundIndex < getLength(sound)) and (fileIndex < len(contents)):
       sample=float(contents[fileIndex])
       setSampleValueAt(sound,soundIndex,sample)
       fileIndex = fileIndex + 1
       soundIndex = soundIndex + 1
   return sound
```

Test it by typing the following at the console:

```
>>> s2=textToSound(getMediaPath("soundText.txt"))
>>> play(s2)
```

3. Now let convert the sound to a picture

```
def soundToPicture(sound):
   picture=makePicture(getMediaPath("640X480.jpg"))
   soundIndex = 0
   for p in getPixels(picture):
      if soundIndex >= getLength(sound):
         break
      sample = getSampleValueAt(sound, soundIndex)
      if sample > 1000:
         setColor(p, red)
      if sample < -1000:
         setColor(p,blue)
      if sample <= 1000 and sample >= -1000:
         setColor(p,green)
      soundIndex = soundIndex + 1
  return picture
```

Try it:

```
>>> sound = makeSound(getMediaPath("thisisatest.wav"))
>>> pic = soundToPicture(sound)
>>> show(pic)
```

4. Now we are going to turn the picture back into sound:

```
def pictureToSound(picture):
   sound = makeEmptySoundBySeconds(10)
   sndIndex = 0
   for p in getPixels(picture):
     if sndIndex == getLength(sound):
        break
     if getRed(p) > 200:
        setSampleValueAt(sound, sndIndex, 1000)
     elif getBlue(p) > 200:
        setSampleValueAt(sound,sndIndex, -1000)
     elif getGreen(p) > 200:
        setSampleValueAt(sound, sndIndex, 0)
     sndIndex = sndIndex + 1
 return (sound)
```

5. Steganography is the science of hiding things inside of other things. Like hiding a message in a digital image file. First we will encode the message. Then we will decode it.

```
def encode(msgPic, original):
   # assume msgPic and original have the same dimensions
   # First, make all red pixels even numbers
   for pxl in getPixels(original):
      if (getRed(pxl) % 2 == 1):
         setRed(pxl, getRed(pxl) - 1)
   # Second, wherever there's black in msgPic
   # make odd the red in the original pic
   for x in range(0, getWidth(original)):
      for y in range(0, getHeight(original)):
         msgPxl = getPixel(msgPic, x, y)
         origPxl = getPixel(original, x, y)
         if distance(getColor(msgPxl), black) < 100.0:
           #it's a message pixel, make the red odd
           setRed(origPxl, getRed(origPxl) + 1)
```

Now to test it:

```
>>> beach = makePicture(getMediaPath("beach.jpg"))
>>> explore(beach)
>>> msg = makePicture(getMediaPath("msg.jpg"))
>>> encode(msg,beach)
>>> explore(beach)
>>> writePictureTo(beach,getMediaPath("beachHidden.png"))
```

Do you see much difference in the two pictures?

6. Now to decode the message:

```
def decode(encodedImg):
  #takes in an encoded image and returns the message
  message=makeEmptyPicture(getWidth(encodedImg), getHeight(encodedImg))
  for x in range(0,getWidth(encodedImg)):
     for y in range (0, getHeight(encodedImg)):
        encPxl = getPixel(encodedImg, x, y)
        msgPxl = getPixel(message, x, y)
        if (getRed(encPxl) % 2 == 1):
           setColor(msgPxl, black)
 return message
```

and now to test it:

```
>>> file = getMediaPath("beachHidden.png")
>>> pic=makePicture(file)
>>> msg=decode(pic)
>>> explore(msg)
```

Hand this in to get credit for this lab