CSCI576 Spring 2012– Assignment 1
Instructor: Parag Havaldar

Assigned : Mon - 1/23/12
Date Due : Mon - 2/13/12 at 12:00 noon

In this assignment you will create a re-sizable image/video jigsaw puzzle or game. This assignment will help you gain a practical understanding of displaying, manipulating and working with media objects like images and video as well as understanding elementary signal processing problems such as resampling and aliasing. Your code will be called as follows

YourProgram.exe C:/myDir/inputImage.rgb  xTiles yTiles scale

where,
- C:/myDir/inputImage.rgb - the image in rgb format. Please see website for details.
- xTiles - the number of tiles horizontally
- yTiles – the number of tiles vertically
- scale – scale factor, if any, for the image, defaults to 1.0 and is generally < 1.0

We have provided a Microsoft Visual C++ project, and a java project to display an image on the screen, along the necessary UI for interaction. This code, along with the accompanying comments should get you started. This source is provided as a reference for students who may not know how to read and display images. You are free to use this, or write your own in code in any programming environment except matlab.

IMPORTANT: Your assignment must compile on the machines provided by the university.

The following sub parts are required to be implemented to set up your game. Each is later described with examples:
1. Display an image – this part is given to you already., but not the resizing part
2. Split Image in a set of tiles – based on a user action of pressing a button
3. Initialize Puzzle – based on a user action of pressing a button.
4. Implement Reset functionality - based on a user action of pressing a button
5. Implement Interactivity to move tiles
6. Start playing!
7. Extend the puzzle to work with video
8. EXTRA CREDIT – Solve the puzzle from its current state programmatically
The image/video data files are given on the DEN class web page with all the format details. You may assume that all the images/video have the same size and frame rate (for video) and other format details as specified on the website. **Remember that when using video, it should play at the given frame rate, and repeatedly in a loop all the time while your application runs.**

**What should you submit?**
Your source code, and your project file or makefile, if any, using the submit program. **Please do not submit any binaries.** We will compile your program and execute our tests accordingly. If you used any special compiler commands then you should also include a README.txt file with your submission explaining to the graders how to compile your code.

Detailed explanations of the above steps with examples are given below.

**1. Display Image:**
This part is already done for you if you choose to use the code we gave you – else write a program, which can display an image. Please refer to the data format given on the class assignment page to understand the size and other format details for the image/video.

![Image Display](image.png)

However, you will need to down sample the image depending on the value of *scale*, which will be < 1.0. The resized image should also have minimal aliasing artifacts. This can be achieved by implementing anti-aliasing using the theory taught in class.
2. **Split Image in a set of tiles:**
Add functionality for the provided “Split” button. Pressing the button should result in subdividing the image into small tiles depending on the input parameter \(xTiles\) and \(yTiles\). Note that, depending on these parameters, the tiles may not necessarily be squares. Each tile is of the same size and must display its corresponding part of the image.

3. **Initialize Puzzle:**
Add functionality for the provided “Initialize” button. Pressing this should shuffle at random all tiles and remove one of the tiles. You may choose any one specific tile to remove or any one at random. Just leave the tile area blank, (or render white).
4. **Implement Reset functionality:**
Add functionality for the provided the “Reset” button. Pressing this should reset everything to where the image is displayed all connected properly and untiled. If in video mode, the video should be playing correctly in a loop as before you split the video.

5. **Implement Interactivity to move tiles:**
The code given to you already traps mouse events. Use that to implement moving tiles. Remember you only need to move one tile position on a mouse click. A tile is selected if the mouse is over a tile when clicked and you will move the tile only if it is vertically or horizontally adjacent to the blank tile. No diagonal movement. For the example shown below, clicking on any vertical or horizontal tile next to the blank tile should trigger an event to exchange tiles. Exchanging your tiles could be made visibly aesthetic by having it animate from one position to another in a second.
6. **Start Solving!**
You should be able to simulate a game playing condition.

7. **Extend the puzzle and all the functionalities to play video!**
You should be able to simulate a game playing condition with video. Make sure the video plays at the frame rate. All functionality – split, initialize, reset, should work with a video playing at the defined frame rate.

8. **Extra credit**
After initializing and or playing for some time, you could be in any unsolved state. For extra credit implement a solve button which when clicked, shows you a step by step way to get from your current state to the solution.
Clicking “solve” should result in an animation towards the solution.