

# Cấu Trúc Dữ Liệu và Giải Thuật

## Bài Tập Thực Hành Tuần #4

19/11/2015

**Deadline: 23h59 ngày 09/12/2015**

### 1. Nội dung bài tập thực hành

Sử dụng cấu trúc dữ liệu queue và thư viện xử lý ảnh OpenCV để mô phỏng việc mua vé xem phim.

### 2. Input, output và một số lưu ý

- Input:** file data.txt gồm 2 bộ test và có dạng như sau:

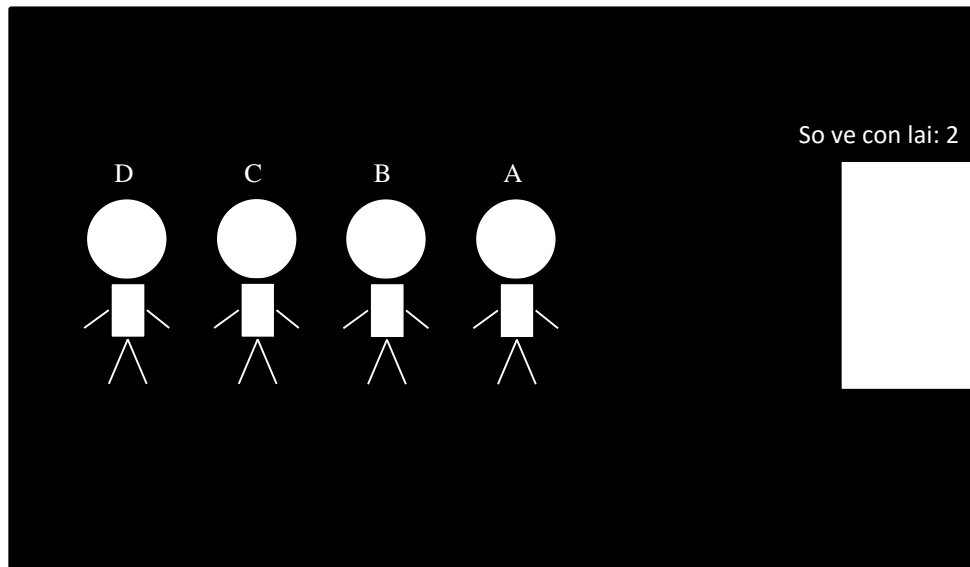
```
[queue_test1]
A B C 3
[queue_test2]
A B C D 2
```

Trong đó, A, B,... là tên người mua vé và cách nhau bởi 1 khoảng trắng. Dữ liệu được thêm vào queue theo thứ tự từ trái sang phải.

Ví dụ: theo file data.txt trên, A sẽ được mua vé trước B....

Số người mua vé là số lượng chữ cái trong từng dòng test. Tổng lượng vé xem phim là số nguyên ở cuối mỗi dòng test.

Ví dụ: dữ liệu test “A B C D 2” cho biết số người đến mua vé là 4 theo thứ tự ưu tiên A, B, C và D. Tổng số vé còn lại là 2 (Hình 1).



**Hình 1.** Mô phỏng mua vé xem phim.

- **Output:** không xuất kết quả vào file. Thay vào đó, tạo cửa sổ giao diện mô phỏng việc mua vé xem phim như trong mục 1.
- Một số lưu ý:
  - Màu cho các đối tượng tùy sinh viên quyết định.
  - Sinh viên có thể tạo hình người tùy ý thích. Có thể tạo thêm các hiệu ứng chuyển động.
  - Mỗi khi có một người mua được vé, cần cập nhật lại số lượng vé còn lại trên màn hình.
  - Hiện tên người mua vé.
  - Chạy lần lượt các bộ test. Mỗi bộ test được chạy trên một cửa sổ có tên lần lượt là [queue\_test1] và [queue\_test2] (cách đặt tên cửa sổ sẽ hướng dẫn ở phần sau).

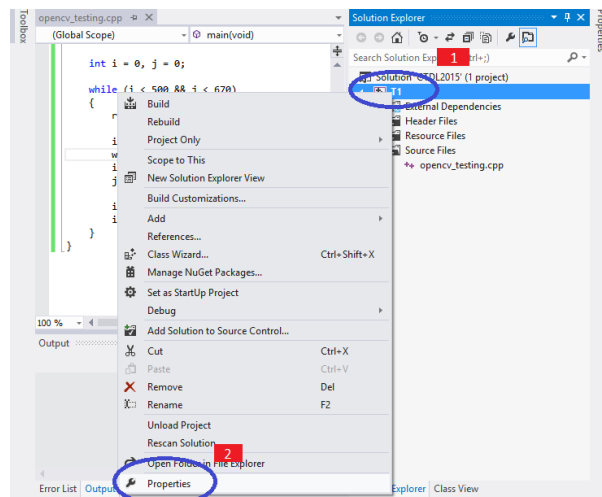
Điểm BTTH #4 dựa trên các tiêu chí sau:

TIÊU CHÍ	ĐIỂM
Có thể mô phỏng toàn bộ số người kèm theo tên	2
Vẽ cửa	1
Hiện số vé còn lại	2
Người có thể di chuyển về phía cửa	2
Người có thể di chuyển ngược phía cửa khi hết vé	1
Hình thức đẹp và sáng tạo	1
Programming style	1
Báo cáo	1
<b>Tổng</b>	<b>11</b>

### 3. Hướng dẫn sử dụng OpenCV trong Visual Studio (Windows)

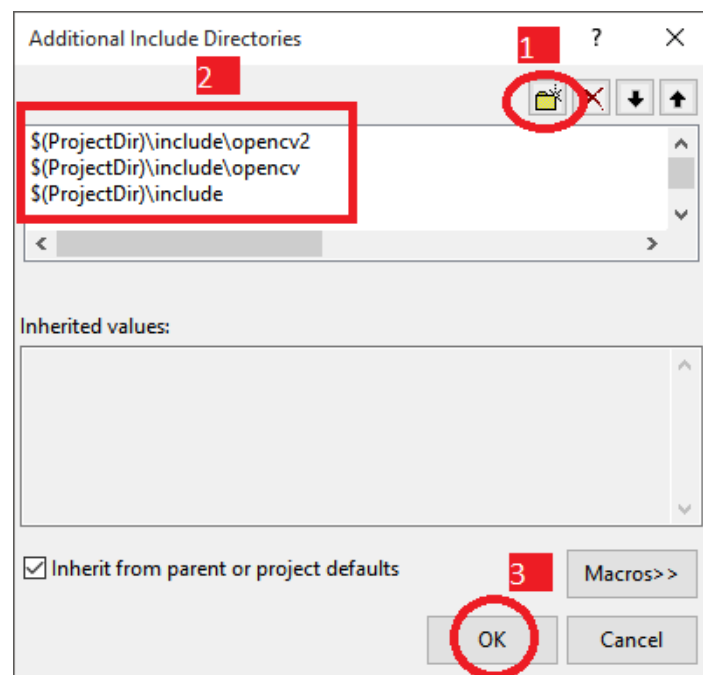
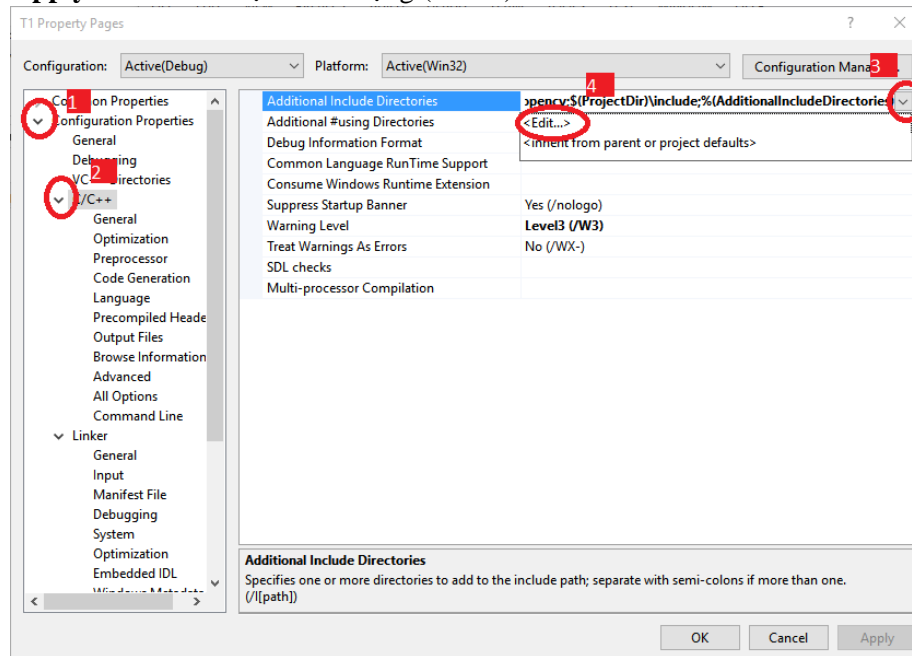
Mình giới thiệu OpenCV cho các bạn thay vì thư viện graphics.h và OpenGL vì những lý do sau:

- OpenCV cài đặt và sử dụng đơn giản hơn OpenGL.
  - Một số cú pháp của OpenCV gần giống với MATLAB (ngôn ngữ lập trình các bạn đã làm quen).
  - OpenCV tốt hơn thư viện graphics.h.
- a. Tải OpenCV 2.4.10.
    - Visual Studio 2010: [opencv2410\\_vs2010.zip](#)
    - Visual Studio 2012: [opencv2410\\_vs2012.zip](#)
    - Visual Studio 2013: [opencv2410\\_vs2013.zip](#)
  - b. Copy opencv2410\_vs201x.zip vào folder project vừa tạo (tương tự như việc copy file data.txt), bấm chuột phải và chọn **Extract Here** (nếu có WinRAR hoặc WinZip).
  - c. Mở Visual Studio, tạo một Win32 Console Application.
  - d. Nhấn chuột phải vào tên project và chọn Properties (Hình 2)



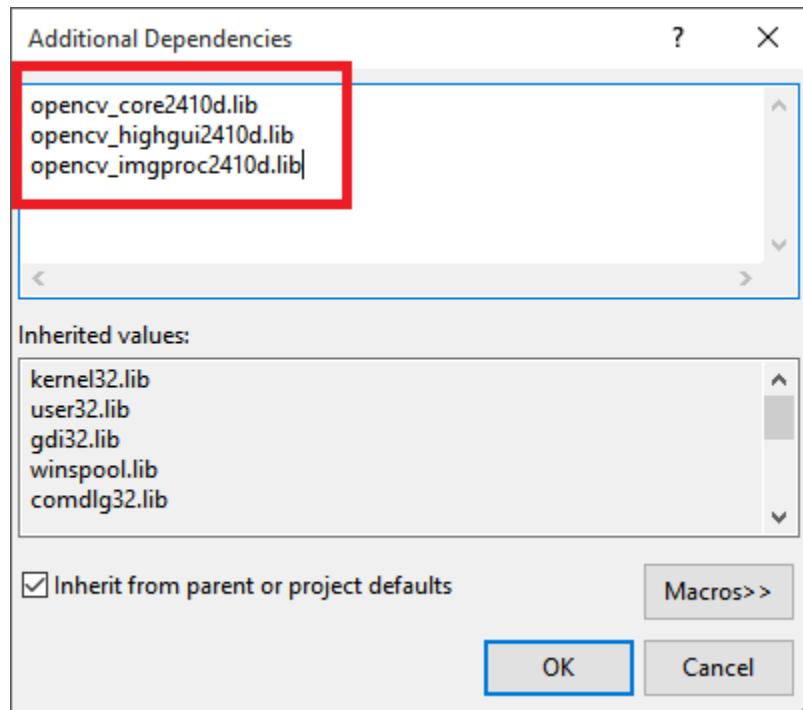
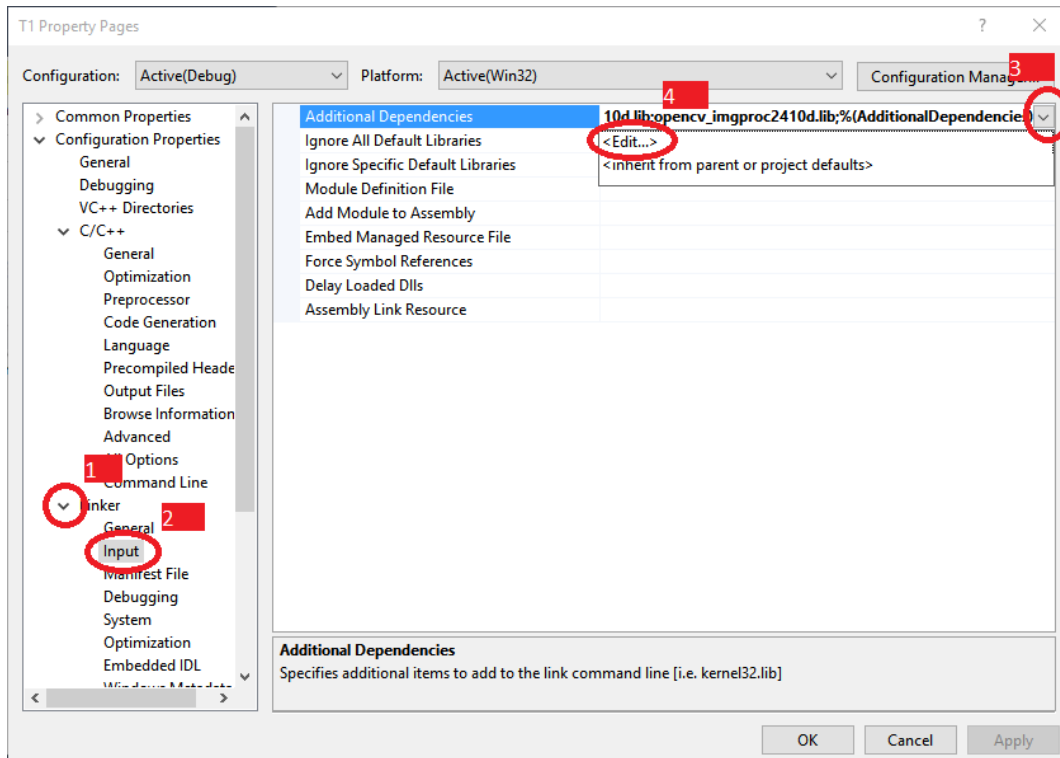
Hình 2. Project properties.

- e. Vào mục Configuration Properties > C/C++ > Additional Include Directories > Edit và khai báo thư mục chứa các file header của OpenCV. Chọn Ok để đóng cửa sổ Additional Include Directories. Nhấn **Apply** để các thiết đặt có tác dụng (Hình 3).



**Hình 3.** Khai báo thư mục chứa các file header của OpenCV.

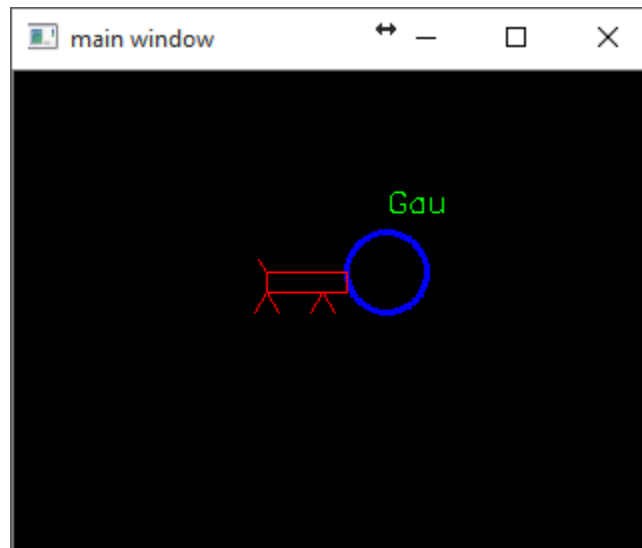
- f. Tại cửa sổ project properties > Linker > Input > Additional Dependencies > Edit để thêm các thư viện tĩnh của OpenCV > OK > Apply > OK để đóng cửa sổ project properties (Hình 4).



**Hình 4.** Thêm thư viện tĩnh của OpenCV.

- g. Các bạn vừa cài đặt xong để thư viện OpenCV có thể chạy được cùng Visual Studio. Sau đây là một chương trình mẫu.

Chương trình này mô phỏng hình ảnh một chú chó di chuyển từ trái sang phải màn hình (Hình 5).



**Hình 5.** Chương trình ví dụ.

Sử dụng project vừa tạo. Thêm file main.cpp có nội dung như sau:

```

1  #include <opencv2\core\core.hpp>
2  #include <opencv2\highgui\highgui.hpp>
3
4  using namespace cv;
5
6  void DrawADog(Mat &img, int x, int y, int stepStyle, char *text)
7  {
8      /* Kích thước đầu */
9      const int HEAD_SIZE = 20;
10     /* Vẽ đầu */
11     circle(img, Point(x, y), HEAD_SIZE,
12           Scalar(255, 0, 0), 2);
13
14     /* Vẽ thân */
15     rectangle(img, Point(x - 3*HEAD_SIZE, y),
16              Point(x - HEAD_SIZE, y + HEAD_SIZE/2),
17              Scalar(0, 0, 255));
18
19     /* Vẽ đuôi */
20     line(img, Point(x - 3*HEAD_SIZE, y),
21          Point(x - 3*HEAD_SIZE - HEAD_SIZE/5, y - HEAD_SIZE/3),
22          Scalar(0, 0, 255));
23
24     /* Kiểu bước chân */
25     const int STEP_STYLE = stepStyle;
26
27
28
29
30

```

```

31
32     /* Vẽ chân trước */
33     line(img, Point(x - 1.5*HEAD_SIZE - HEAD_SIZE/10, y + HEAD_SIZE/2),
34           Point(x - 1.5*HEAD_SIZE - HEAD_SIZE/10 -
35               STEP_STYLE, y + HEAD_SIZE),
36           Scalar(0, 0, 255));
37     line(img, Point(x - 1.5*HEAD_SIZE - HEAD_SIZE/10, y + HEAD_SIZE/2),
38           Point(x - 1.5*HEAD_SIZE - HEAD_SIZE/10 +
39               STEP_STYLE, y + HEAD_SIZE),
40           Scalar(0, 0, 255));
41
42     /* Vẽ chân sau */
43     line(img, Point(x - 3*HEAD_SIZE, y + HEAD_SIZE/2),
44           Point(x - 3*HEAD_SIZE - STEP_STYLE, y + HEAD_SIZE),
45           Scalar(0, 0, 255));
46     line(img, Point(x - 3*HEAD_SIZE, y + HEAD_SIZE/2),
47           Point(x - 3*HEAD_SIZE + STEP_STYLE, y + HEAD_SIZE),
48           Scalar(0, 0, 255));
49
50     /* Vẽ chữ */
51     putText(img, text, Point(x, y - 1.5*HEAD_SIZE),
52             FONT_HERSHEY_SIMPLEX, 0.5, Scalar(0, 255, 0));
53 }
54
55 /* Chương trình chính */
56 void main(void)
57 {
58     /* Khai báo biến img kiểu Mat (tương tự 1 ma trận trong MATLAB)*/
59     /* có kích thước 240 dòng và 320 cột */
60     /* ảnh gồm 3 lớp R, G và B (ảnh màu) */
61     Mat img(240, 320, CV_32FC3);
62
63     /* Khai báo tọa độ ban đầu của đối tượng (chó) */
64     int y = 100, x = 0;
65
66     /* Di chuyển đối tượng */
67     while (x < 350)
68     {
69         /* Vẽ đối tượng */
70         DrawADog(img, x, y, x % 15, "gau");
71
72         /* Dùng imshow (tương tự imshow/plot MATLAB) */
73         imshow("main window", img);
74         /* Tạm dừng màn hình trong 10 milliseconds */
75         waitKey(10);
76
77         /* Dùng hàm zeros (tương tự MATLAB)
78            để đưa img về ma trận 0 (màu đen) */
79         /* Thao tác này tương đương với xóa màn hình */
80         img = Mat::zeros(img.rows, img.cols, CV_32FC3);
81         imshow("main window", img);
82
83         x++;
84     }
85 }

```

**4. Một số thao tác trong OpenCV**

STT	THAO TÁC	CHỨC NĂNG
1	<code>Mat img(240, 320, CV_32FC3);</code>	Khai báo mảng số thực img 3 lớp, mỗi lớp có 240 dòng và 320 cột.
2	<code>Scalar(r, g, b)</code>	Định nghĩa màu qua 3 tham số: <ul style="list-style-type: none"> <li>• r = red,</li> <li>• g = green,</li> <li>• b = blue.</li> </ul> Xem thêm: <a href="#">pha màu RGB</a> .
3	<code>Point(x, y)</code>	Định nghĩa điểm (x, y).
4	<code>line(img, Point(x1, y1), Point(x2, y2), Scalar(r, g, b));</code>	Vẽ đường thẳng có màu <code>Scalar(r, g, b)</code> tạo bởi 2 điểm (x1, y1) và (x2, y2) lên ma trận img.
5	<code>circle(img, Point(x, y), R, Scalar(r, g, b));</code>	Vẽ đường tròn O((x, y), R) có màu <code>Scalar(r, g, b)</code> lên ma trận img.
6	<code>rectangle(img, Point(x1, y1), Point(x2, y2), Scalar(r, g, b));</code>	Vẽ hình chữ nhật có màu <code>Scalar(r, g, b)</code> được tạo bởi điểm trên trái (x1, y1) và điểm dưới phải (x2, y2) lên ma trận img.
7	<code>putText(img, "ABC", Point(x, y), FONT_HERSHEY_SIMPLEX, 0.5, Scalar(r, g, b));</code>	Vẽ chữ ABC có màu <code>Scalar(r, g, b)</code> tại vị trí (x, y) lên ma trận img.
8	<code>imshow("main window", img);</code>	Hiện ảnh img lên cửa sổ giao diện có tên <code>main window</code> .
9	<code>waitKey(t);</code>	Tạm dừng cửa sổ giao diện OpenCV trong t mili-giây.
10	<code>img = Mat::zeros(img.rows, img.cols, CV_32FC3);</code>	Đưa ma trận img về ma trận 0
11	<code>img = Mat::zeros(img.rows, img.cols, CV_32FC3);</code> <code>imshow("main window", img);</code>	Xóa màn hình cửa sổ giao diện <code>main window</code> .

Với những gì mình cung cấp về OpenCV, các bạn có thể hoàn thành tốt BTTH #4. Ngoài ra, nếu muốn tìm hiểu thêm về thư viện xử lý ảnh này, các bạn có thể vào trang web [OpenCV API Reference](#).

Chúc các bạn làm tốt.