**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**MÔN: LẬP TRÌNH TRỰC QUAN**

**ĐỀ TÀI: PHẦN MỀM PAINT**

**GIẢNG VIÊN HƯỚNG DẪN**

* Thầy Lê Thanh Trọng
* Thầy Huỳnh Tuấn Anh

**SINH VIÊN THỰC HIỆN**

Bùi Đình Lộc thọ 13520844

Lê Tấn Thịnh 13520836

# **I – GIỚI THIỆU**

# **II – QUY TRÌNH**

## **Phân chia công việc**

|  |  |
| --- | --- |
| **Họ và tên** | **Công Việc** |
| Bùi Đình Lộc Thọ |  |
| Lê Tấn Thịnh |  |

## **Giới thiệu GDI+**

- GDI (Graphics Development Interface) là các API của Windows cung cấp các hàm và các CTDL cần thiết để tạo ra những hình ảnh đồ họa 1 cách nhanh chóng và hiệu quả ra các thiết bị phần cứng (như màn hình, máy in).

- GDI+ cải thiện GDI với các đặc điểm mới và tối tưu các đặc điểm đã có ở GDI.

- Trong Microsoft Visual Studio, GDI+ có trong khối hợp ngữ System.Drawing.dll, 1 vài namespace trong đó là:

System.Drawing.Design;

System.Drawing;

System.Drawing.Imaging;

System.Drawing.Drawing2D;

System.Drawing.Printing;

System.Drawing.Text;

- Các thư viện sử dụng trong đồ án:

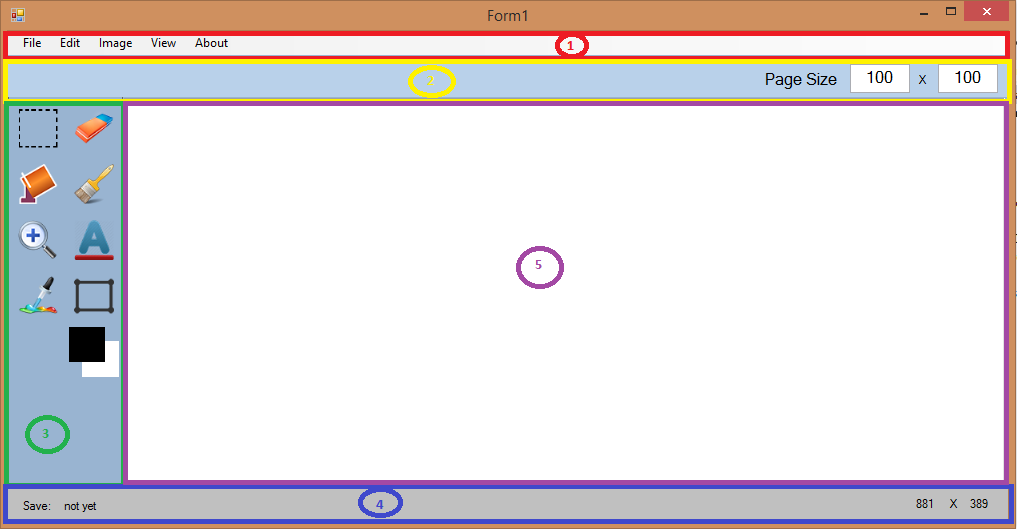
System.Drawing;

System.Drawing.Imaging;

## **Thiết kế phần mềm**

### **3.1 Màn hình**

- Phần mềm có 1 màn hình chính như dưới.



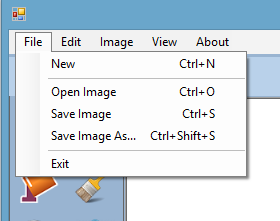
|  |  |  |
| --- | --- | --- |
| **STT** | **Tên** | **Chức năng** |
| 1 | MenuBar | Chứa các menu để người dùng thao tác lên bảng vẽ như(New, Open, Save, Undo, Redo, Convert…) |
| 2 | PropertiesBar | Hiển thị các thuộc tính của công cụ mà người dùng chọn, từ đó có thể chỉnh sửa sao cho phù hợp |
| 3 | ToolBox | Chứa các công cụ hỗ trợ vẽ ( như: Brush, Eraser, Marquee, Zoom, Fill, Text, Shape, LeftColor, RightColor, ColorPicker) |
| 4 | StatusBar | Hiện thị trạng thái của bảng vẽ(Kích thước của trang vẽ, đã được lưu hay chưa) |
| 5 | WorkSpace | Bảng vẽ (Người dùng sử dụng các công cụ để vẽ lên đây) |

### **3.2 Chức năng Chính**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Biểu tượng** | **Loại Control** | **Chức năng** |
| Marquee | F:\Github\Paint\Paint-Crazyland\Resources\Marquee-Rectangle.png | Button | Hỗ trợ cắt và di chuyển 1 phần ảnh trên WorkSpace |
| Fill | F:\Github\Paint\Paint-Crazyland\Resources\fill-tool.png | Button | Tô màu trên 1 vùng |
| Eraser | F:\Github\Paint\Paint-Crazyland\Resources\eraser-tool.png | Button | Cục tẩy, dùng để xóa các nét vẽ |
| Brush | F:\Github\Paint\Paint-Crazyland\Resources\brush-tool.png | Button | Cọ vẽ, dùng để vẽ lên WorkSpace |
| Zoom | F:\Github\Paint\Paint-Crazyland\Resources\zoom-tool.png | Button | Hỗ trợ phóng to ảnh trên WorkSpace |
| Text | F:\Github\Paint\Paint-Crazyland\Resources\text-tool.png | Button | TextBox để nhập các ký tự |
| Color Picker | F:\Github\Paint\Paint-Crazyland\Resources\colorpicker.png | Button | Lấy màu tại vị trí click chuột và cài đặt màu cho LeftColor |
| Shape | F:\Github\Paint\Paint-Crazyland\Resources\rectangle_stroked.png | Button | Hỗ vẽ các đường thẳng, hình tam giác, tứ giác, ngũ giác, lục giác lên WorkSpace |
| Colors |  | Button | Hiển thị màu hiện tại(LeftColor: chuột trái, RightColor: chuột phải) của công cụ vẽ, khi click vào sẽ hiển thị hộp thoại chọn màu |

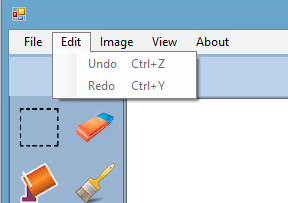
### **3.3 Chức năng Menu**

#### **3.3.1 Menu File**



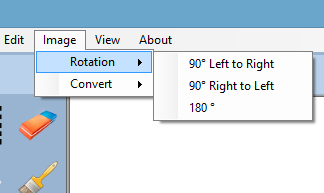
|  |  |
| --- | --- |
| **Tên MenuItem** | **Chức năng** |
| New | Tạo WorkSpace mới |
| Open Image | Mở một hình có sẵn |
| Save Image | Lưu WorkSpace |
| Save Image As | Lưu WorkSpace dưới dạng.... |
| Exit | Thoát chương trình |

#### **3.3.2 Menu Edit**

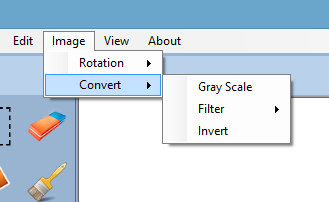


|  |  |
| --- | --- |
| **Tên MenuItem** | **Chức năng** |
| Undo | Lùi về trước 1 thao tác |
| Redo | Tiến trước 1 thao tác |

#### **3.3.3 Menu Image**

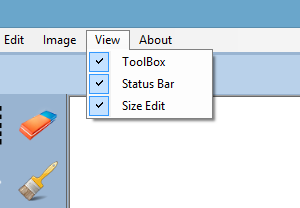
****

|  |  |
| --- | --- |
| **Tên MenuItem** | **Chức năng** |
| 90° Left to Right | Xoay ảnh sang phải 90 độ |
| 90° Right to Left | Xoay ảnh sang trái 90 độ |



|  |  |
| --- | --- |
| **Tên MenuItem** | **Chức năng** |
| Invert | Chuyển sang màu âm bảng |
| Filter | Lọc màu |
| Gray Scale | Chuyển sang ảnh xám |

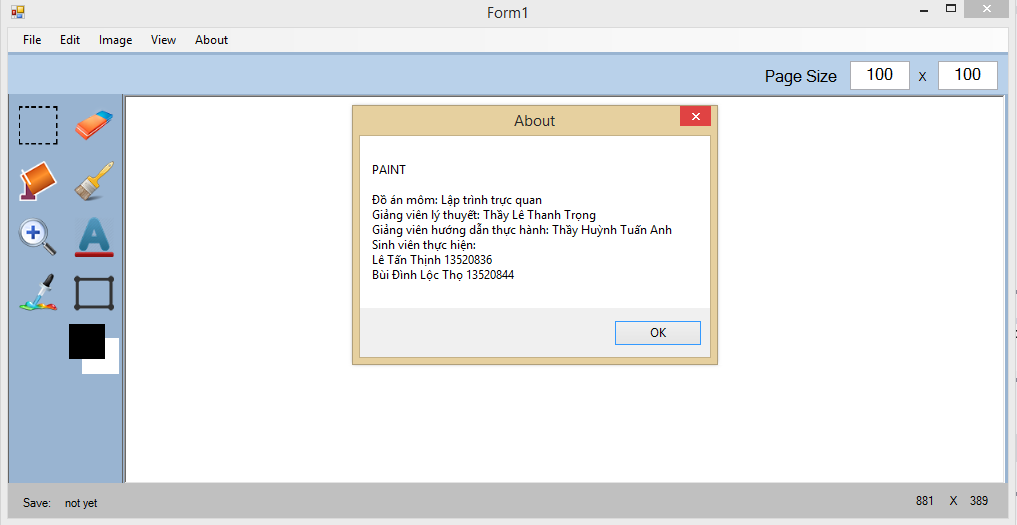
#### **3.3.4 Menu View**

****

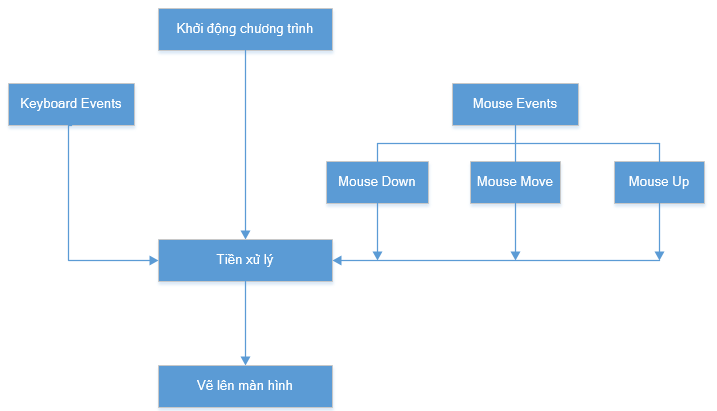
|  |  |
| --- | --- |
| **Tên MenuItem** | **Chức năng** |
| ToolBox | Ẩn, hiện thanh ToolBox |
| StatusBar | Ẩn, hiện thanh StatusBar |
| Size Edit | Ẩn, hiện công cụ thay đổi kích thước WorkSpace (Page Size nằm bên góc phải) |

#### **3.3.5 Menu About**

- Thông tin về tác giả và giảng viên hướng dẫn



## **Luồng sự kiện**



- Khối khởi động chương trình: lúc chương trình được mở lên khởi tạo các thành phần cần thiết cho chương trình.

- Khối Keyboard Events: các sự kiện từ bàn phím

- Khối Mouse Events: các sự kiện từ chuột, chia ra làm 3 event nhỏ

+ Mouse Down: sự kiện khi chuột được nhấp xuống.

+ Mouse Move: sự kiện khi chuột di chuyển.

+ Mouse Up: sự kiện khi chuột được nhả ra.

- Khối tiền xử lý: xử lý chương trình, các sự kiện trước khi vẽ lên màn hình.

- Vẽ lên màn hình: sau khi xử lý các sự kiện xong bắt đầu vẽ lên màn hình.

## **Kĩ thuật**

### **Thuật toán Flood Fill**

- là thuật toán dùng tô màu được dùng cho chức năng fill.

**- Nguyên lý hoạt động:**

* + Lấy 1 điểm gốc ban đầu đó là điểm nhấp chuột xuống vùng tô màu và nhét vào Stack.
  + Tiếp đến ta quét các pixel phía bên trên đến khi gặp pixel màu khác với màu nền hoặc là chạm tới cạnh bên trên của Bitmap.
  + Ta bắt đầu tô màu các pixel từ trên xuống dưới cho đến khi gặp pixel khác màu mới màu nền hoặc chạm cạnh dưới của Bitmap. Trong quá trình tô màu xuống bên dưới thì ta sẽ kiểm tra phía bên trái và bên phải của pixel đang tô màu có giống màu nền không (tức là chưa được tô) nếu giống thì ta sẽ thêm vào Stack, và chỉ lấy 1 lần bên trái và 1 lần bên phải (dùng 2 biến bool để xác định xem đã lấy chưa, một biến cho bên trái và 1 biến cho bên phải).
  + Sau khi tô xong đường thẳng đầu tiên thì ta tiếp tục lấy điểm tiếp theo trong Stack và thực hiện lại, vòng lặp sẽ kết thúc khi số phần tử trong Stack hết.

### **Lock bits**

- Là kĩ thuật dùng để cải tiến việc Get/Set màu của pixel.

- Trong C# nếu chúng ta truy xuất để lấy màu và tô màu cho pixel trong 1 Bitmap lớn nhiều lần và liên tục sẽ gây ra hiện tượng đơ chương trình nếu máy yếu có thể bị crash.

**- Nguyên lý hoạt động:**

* Đầu tiên ta sẽ cần một đối tượng BitmapData và khóa vùng cần xử lý của Bitmap lại thông qua phương thức LockBits.
* Sau khi lấy được BitmapData ta sẽ dùng 1 con trỏ int (IntPtr) để lấy pixel đầu tiền trong BitmapData.
* Tiếp đến ta sẽ lấy độ dài của của ma trận pixel, ở đây ma trận là mảng 1 chiều, BitmapData có hỗ trợ thuộc tính Stride = Width \* 4 (4 ở đây là vì 1 pixel có 4 thuộc tính là ARGB).
* Sau khi có độ dài ma trận pixel của Bitmap ta tạo 1 mảng kiểu byte sau đó copy dữ liệu từ BitmapData sang ma trận này thông qua con trỏ Int đã lấy ở trên, ta dùng hàm System.Runtime.InteropServices.Marshal.Copy của C#.
* Mọi việc xử lý sẽ được thực hiện trên ma trận kiểu byte trên.
* Sau khi xử lý xong thì ta UnlockBits bằng cách copy lại dữ liệu từ mảng kiểu byte sang vị trí con trỏ int và dùng phương thức UnlockBits của Bitmap với tham số truyền vào là BitmapData.

# **III – TỔNG KẾT**

## **Đánh giá**

* Phần mềm chạy tốt trên các máy hỗ trợ .NET framework 3.5 trở lên.
* Kích thước chương trình nhỏ gọn (hơn 100KB)
* Chiếm tài nguyên máy ít (khoảng 10 – 15MB RAM)
* Tốc độ xử lý chưa tốt lắm, vẫn còn giật, lag đặc biệt chức năng fill

## **Website và tài liệu tham khảo**

**- Websites:**

* + Thuật toán tô màu:

<http://vietgamedev.net/forum/thread/206/help-gi%E1%BA%A3i-thu%E1%BA%ADt-t%C3%B4-m%C3%A0u-loang/>

* + Tham khảo các thư viện của C#:  
    http://stackoverflow.com, <https://msdn.microsoft.com> và cùng các website khác.