**TRƯỜNG ĐẠI HỌC SƯ PHẠM KĨ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

****

**BÁO CÁO ĐẶC TẢ ĐỒ ÁN 1**

**ĐỀ TÀI SỐ 6**

**VIẾT CHƯƠNG TRÌNH REMOTE DRAW TẠO MÔI TRƯỜNG LÀM VIỆC CỘNG TÁC CHO PHÉP NHIỀU USER CÙNG VẼ TRÊN 1 BẢN VẼ Ở XA**

**GVHD: Ths. Nguyễn Đăng Quang**

**Nhóm 2 Ngô Đặng Đồng Khởi - 17110164**

**Lê Huỳnh Giảng - 17110128**

*Tp.Hồ Chí Minh, tháng 12 năm 2019*

**MỤC LỤC**

[TÀI LIỆU THAM KHẢO 2](#_Toc26693402)

[DANH MỤC HÌNH ẢNH 3](#_Toc26693403)

[DANH MỤC BẢNG 4](#_Toc26693404)

I. [ĐẶC TẢ 5](#_Toc26693405)

[1. Giới thiệu phần mềm 5](#_Toc26693406)

[2. Dự liệu thông tin đầu vào 5](#_Toc26693407)

[3. Tình huống sử dụng 5](#_Toc26693408)

[4. Giao diện dự kiến 6](#_Toc26693409)

[5. Tính năng của ứng dụng 7](#_Toc26693410)

[5.1. Vẽ hình: 7](#_Toc26693411)

[5.2. Tùy chỉnh thuộc tính vẽ: 7](#_Toc26693412)

[5.3. Chỉnh sửa hình: 7](#_Toc26693413)

[5.4. Chức năng về kết nối vẽ từ xa: 7](#_Toc26693414)

[5.5. Chức năng khác: 7](#_Toc26693415)

[6. Các Control được sử dụng trong chương trình 8](#_Toc26693416)

II. [PhÂn Công Công Việc](#_Toc26693405) 9

III. [Thiết Kế 10](#_Toc26693405)

[1. Diagram 10](#_Toc26693419)

[2. Thiết kế lớp 10](#_Toc26693420)

[2.1. Các lớp được sử dụng trong chương trình 10](#_Toc26693421)

[2.2. Các phương thức trong lớp 12](#_Toc26693422)

IV. [Cài ĐẶt Kiểm Thử 19](#_Toc26693405)

[1. Lập kế hoạch 19](#_Toc26693424)

[1.1. Mục tiêu 19](#_Toc26693425)

[1.2. Môi trường 19](#_Toc26693426)

[2. Thực thi 19](#_Toc26693427)

[2.1. Trường hợp 1: dùng các công cụ vẽ 19](#_Toc26693428)

[2.2. Trường hợp 2: dùng công cụ di chuyển 20](#_Toc26693429)

[2.3. Trường hợp 3: dùng công cụ đổi màu 21](#_Toc26693430)

[2.4. Trường hợp 4: thay đổi độ trong suốt 21](#_Toc26693431)

[2.5. Trường hợp 5: thay đổi độ dày nét vẽ 22](#_Toc26693432)

[2.6. Trường hợp 6: thay đổi màu theo hệ ARGB 22](#_Toc26693433)

[2.7. Trường hợp 7: thay đổi màu theo bảng màu mở rộng 23](#_Toc26693434)

[2.8. Trường hợp 8: dùng chức năng xoay hình (áp dụng đối với hình vuông) 23](#_Toc26693435)

[2.9. Trường hợp 9: xóa hình 24](#_Toc26693436)

[2.10. Trường hợp 10: chức năng undo 24](#_Toc26693437)

[2.11. Trường hợp 11: dùng chức năng save 25](#_Toc26693438)

[2.12. Trường hợp 12: dùng chức năng open 25](#_Toc26693439)

[2.13. Trường hợp 13: dùng chức năng xuất ra file .png 26](#_Toc26693440)

[2.14. Trường hợp 15: hoạt động với mode server - client 26](#_Toc26693441)

[2.15. Trường hợp 16: hoạt động với mode server 1-n client 27](#_Toc26693442)

[2.16. Trường hợp 17: đổi chế độ 27](#_Toc26693443)

[2.17. Trường hợp 18: máy hoạt động với mode client bị tắt 28](#_Toc26693444)

[2.18. Trường hợp 19: máy hoạt động với mode server bị tắt 28](#_Toc26693445)

[3. Đánh giá 28](#_Toc26693446)

[3.1. Mức độ đáp ứng 28](#_Toc26693447)

V. [KẾT LUẬN 28](#_Toc26693405)

[1. Kết quả đạt được 28](#_Toc26693449)

[2. Kết luận 28](#_Toc26693450)

[2.1. Ưu điểm 28](#_Toc26693451)

[2.2. Nhược điểm 28](#_Toc26693452)

[3. Đề xuất cải thiện 28](#_Toc26693453)

# TÀI LIỆU THAM KHẢO

<https://www.javatpoint.com/java-actionlistener>

<https://examples.javacodegeeks.com/desktop-java/awt/event/adjustmentlistener-example/>

<https://www.javatpoint.com/java-actionlistener>

<https://stackoverflow.com/questions/6714045/how-to-resize-jlabel-imageicon>

<http://www.java2s.com/Tutorial/Java/0240__Swing/Verticalmenubar.htm>

<https://docs.oracle.com/javase/tutorial/2d/images/saveimage.html>

<https://www.mkyong.com/swing/java-swing-jfilechooser-example/>

<https://stackoverflow.com/questions/26559981/jpanel-snapshot-and-save-the-image-in-a-file-it-shows-black-image-what-to-do>

<https://www.mkyong.com/swing/java-swing-jfilechooser-example/>

<https://www.ntu.edu.sg/home/ehchua/programming/java/J4b_CustomGraphics.html>

<https://www.geeksforgeeks.org/socket-programming-in-java/>

<https://www.geeksforgeeks.org/socket-programming-in-java/>

<https://www.geeksforgeeks.org/socket-programming-in-java/>

<https://www.geeksforgeeks.org/socket-programming-in-java/>

<https://www.geeksforgeeks.org/socket-programming-in-java/>

<https://www.ntu.edu.sg/home/ehchua/programming/java/J4b_CustomGraphics.html>

# DANH MỤC HÌNH ẢNH

*Hình 1: Giao diện dự kiến*

*Hình 2: Class Diagram*

*Hình 3: Vẽ line, pencil*

*Hình 4: Test các chế độ vẽ rectangle, circle, elipse, triangle, text.*

*Hình 5: Di chuyển các hình ảnh đã vẽ*

*Hình 6: Đổ màu lên hình*

*Hình 7: Thay đổi độ trong suốt*

*Hình 8: Thay đổi độ dày nét vẽ*

*Hình 9: Thay đổi màu theo hệ ARGB*

*Hình 10,11: Bảng màu mở rộng*

*Hình 12: Chức năng xoay hình (rotate)*

*Hình 13: Chức năng cục tẩy*

*Hình 14: Chức năng undo*

*Hình 15: Chức năng save*

*Hình 16: Chức năng open*

*Hình 17,18: Dùng chức năng xuất ra file .png*

*Hình 19: Hoạt động với mode server-client*

*Hình 20: Hoạt động với mode server 1-n client*

*Hình 21: Đổi chế độ*

# DANH MỤC BẢNG

*Bảng 1: Danh sách các Control được sử dụng*

*Bảng 2: Phân công công việc*

*Bảng 3: Các lớp được sử dụng trong chương trình, sinh viên phụ trách: Lê Huỳnh Giảng*

*Bảng 4: Các lớp được sử dụng trong chương trình, sinh viên phụ trách: Ngô Đặng Đồng Khởi*

*Bảng 5: Các phương thức trong project*

## Đặc tả

### Giới thiệu phần mềm

RemoteDraw (RD) là một phần mềm hỗ trợ vẽ các đối tượng hình học 2d trực tuyến trên giao diện đồ họa chỉ bằng những thao tác kéo thả chuột đơn giản. Ngoài việc có thể tạo các bản vẽ cá nhân RD còn cho phép chia sẻ bản vẽ với người user từ xa bao gồm trình chiếu trực tuyến thời gian thực và tương tác sửa chữa với mọi người.

### Dự liệu thông tin đầu vào

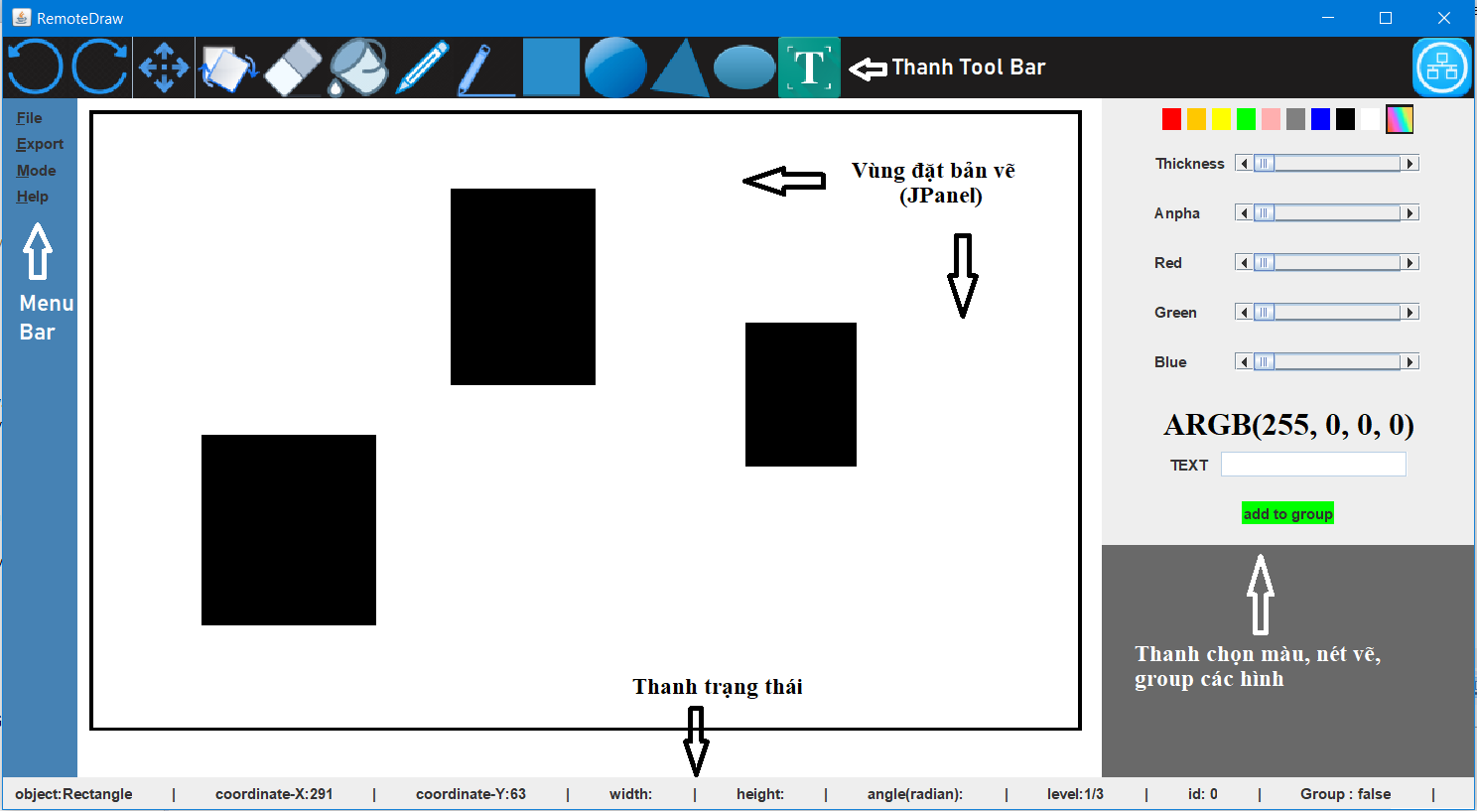
Phần mềm nhận các thông tin tương tác từ người dùng thông qua các thao tác với chuột trên cửa sổ làm viêc (click, press, drag, drop) cùng với nhận các gói tin được gởi từ xa qua mạng LAN.

### Tình huống sử dụng

Phần mềm được sử dụng khi người dùng muốn vẽ những đối tượng hình học cơ bản như hình tròn, hình vuông, tam giác, … chỉ bằng các thao tác đơn giản với chuột, cung cấp tính năng điều chỉnh màu sắc, độ dày nét vẽ, vị trí, kích cỡ, của các đối tượng hình học cùng với khả năng khôi phục lại các trạng thái, thao tác đã thực hiện trước đó.

Các tài liệu Remote Paint (RP) có thể chia sẻ và chỉnh sửa cùng lúc với nhiều người khác nhau, thông qua việc kích hoạt chế độ Share trên máy hiện tại (server), các user khác (client) trong mạng sẽ có thể tham gia cộng tác khi kích hoạt chế độ Join và nhập ip máy chủ. Toàn bộ những thành viên có tham gia vào môi trường cộng tác của RP được hiển thị tên trên giao diện cộng tác và có thể trao đổi thông tin qua hộp tin nhắn một cách trực tuyến và công khai. Mỗi lần nỗi dung thay đổi đều được tự động cập nhật đến tất cả thành viên trong nhóm cộng tác.

### Giao diện dự kiến



*Hình 1: Giao diện dự kiến*

### Tính năng của ứng dụng

#### Vẽ hình:

* Vẽ bất kì (nét vẽ không theo hình dạng cụ thể)
* Đoạn thẳng (Line)
* Hình chữ nhật (Rectangle)
* Đường tròn (Circle)
* Hình tam giác (Triangle)
* Hình đa giác (Polygon)

#### Tùy chỉnh thuộc tính vẽ:

* Màu cơ bản
* Màu theo hệ ARGB
* Độ dày nét
* Độ trong suốt

#### Chỉnh sửa hình:

* Đổi vị trí
* Đổi màu
* Xoay hình (chỉ áp dụng cho hình chữ nhật)
* Xóa hình

#### Chức năng về kết nối vẽ từ xa:

* Kết nối các máy trong cùng mạng Lan
* Đồng bộ bản vẽ đến mọi máy được kết nối
* Chọn Mode là Server hoặc Client tùy người dùng muốn

#### Chức năng khác:

* Undo, redo
* Save, Open, New File
* Xuất bản ra file png

### Các Control được sử dụng trong chương trình

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Control** | **Chức năng** |
| 1 | JPanel, JFrame | Tạo không gian làm việc chính trong chương trình |
| 2 | JButton, JRadioButton | Tạo các nút chức năng trong chương trình |
| 3 | MenuBar | Bao gồm những tùy chọn New, Save, Open, Exit, About… |
| 4 | JLabel | Tạo các dòng text |
| 5 | JTextField | Dùng để nhập lệnh |
| 6 | Color | Dùng để chọn màu |
| 7 | [JColorChooser](https://vietjack.com/java_swing/lop_jcolorchooser_trong_java_swing.jsp) | Cung cấp một pane gồm các control được thiết kế để cho phép một người dùng thao tác và lựa chọn màu |
| 8 | Graphics,Graphics2D | Cung cấp nhiều phương thức để lập trình đồ họa để vẽ |
| 9 | JFileChooser | Biểu diễn một dialog window từ đó người dùng có thể lựa chọn một file để Save, Open,.. |

*Bảng 1: Danh sách các Control được sử dụng*

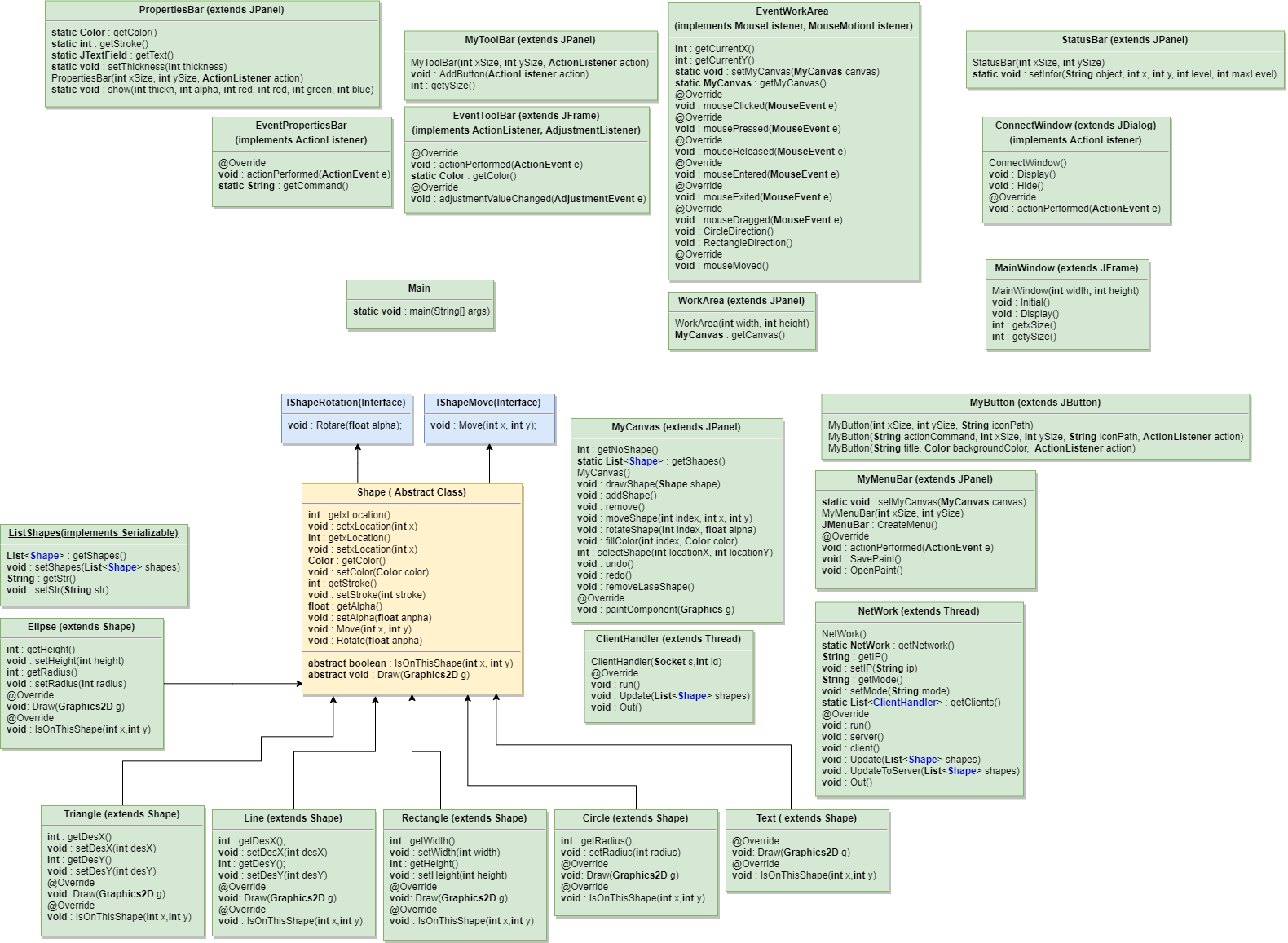
## PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên sinh viên** | **Công việc thực hiện** | **Phần trăm đóng góp** |
| 1 | Lê Huỳnh Giảng | Tìm hiểu đề tài  Thiết kế lớp  Thiết kế giao diện  Viết chức năng nhập lệnh  Kiểm thử phần mềm  Viết báo cáo | 50% |
| 2 | Ngô Đặng Đồng Khởi | Tìm hiểu đề tài  Thiết kế lớp  Thiết kế giao diện  Viết chức năng nhập lệnh  Kiểm thử phần mềm  Viết báo cáo | 50% |

*Bảng 2: Phân công công việc*

## Thiết kế

### Diagram



*Hình 2: Class Diagram*

### Thiết kế lớp

#### Các lớp được sử dụng trong chương trình

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên lớp** | **Mục đích** |
| 1 | Main | Chứ hàm main  Khai báo đối tượng |
| 2 | MainWindow  Kế thừa Jframe | Cửa số ứng dụng chính |
| 3 | ConnectWindow  Kế thừa Jdialog  Dẫn xuất Action Listener | Cửa sổ chọn chế độ (client/server) |
| 4 | MyButton  Kế thừa JButton | Tạo button theo kích cỡ, icon, màu, chức năng được chỉ định |
| 5 | MyCanvas  Kế thừa Jpanel | Vùng để vẽ |
| 6 | MyMenuBar  Kế thùa Jpanel  Dẫn xuất ActionListener | Tạo thanh menu dọc |
| 7 | MyToolBar  Kế thừa JPanel | Tạo thanh công cụ |
| 8 | PropertiesBar  Kế thừa JPanel | Tạo thanh thuộc tính vẽ |
| 9 | StatusBar  Kế thừa JPanel | Tạo thanh trạng thái |
| 10 | WorkArea  Kế thừa JPanel | Vùng đặt MyCanvas |
| 11 | NetWork  Kế thừa Thread | Tạo luồn Xử lý kết nối socket |
| 12 | ClientHandler  Kế thừa Thread | Tạo luồng Xử lý kết nối riêng cho mỗi Client |
| 13 | IShapeMove | Các đối tượng có khả năng duy chuyển dẫn xuất interface này |
| 14 | IShapeRotation | Các đối tượng có khả năng xoay dẫn xuất interface này |
| 15 | Shape  Dẫn xuất IShapeMove  Dẫn xuất IshapeRotation  Dẫn xuất Serializable | Các đối tượng là các hình học và có thể được lưu trữ thành chuỗi các byte kế thừa abstract này |

*Bảng 3: Các lớp được sử dụng trong chương trình, sinh viên phụ trách: Lê Huỳnh Giảng*

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên lớp** | **Mục đích** |
| 1 | EventPropertiesBar  Dẫn xuất ActionListener  Dẫn xuất AdjustmentListener | Bắt sự kiện button và scroll trên PropertiesBar |
| 2 | EventToolBar  Dẫn xuất ActionListener | Bắt sự kiện button trên MyToolBar |
| 3 | EventWorkArea  Dẫn xuất MouseListener  Dẫn xuất MouseMotionListener | Bắt sự kiện chuột và con trỏ duy chuyển trên MyCanvas |
| 4 | Circle  Kế thừa Shape | Đối tượng chứa thuộc tính hình tròn |
| 5 | Elipse  Kế thừa Shape | Đối tượng chứa thuộc tính hình elipse |
| 6 | Line  Kế thừa Shape | Đối tượng chứa thuộc tính đường thẳng |
| 7 | PolyLine  Kế thừa Shape | Đối tượng chứa thuộc tính của nét vẽ không theo hình dạng cụ thể |
| 8 | Rectangle  Kế thừa Shape | Đối tượng chứ thuộc tính hình chữ nhật |
| 9 | Triangle  Kế thừa Shape | Đối tượng chứ thuộc tính hình tam giác |
| 10 | Text  Kế thừa Shape | Đối tượng chứa thuộc tính ô văn bản |
| 11 | ListShape  Dẫn xuất Serializable | Đối tượng chứa danh sách các đối tượng Shape, có lưu đối tượng thành file hay gửi qua mạng |

*Bảng 4: Các lớp được sử dụng trong chương trình, sinh viên phụ trách: Ngô Đặng Đồng Khởi*

#### Các phương thức trong lớp

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Phương thức** | **Mục đích** | **Tên file, số thứ tự dòng chứa khai báo** |
| 1 | **actionPerformed(ActionEvent e)**  Input: e  Output: không có  <https://www.javatpoint.com/java-actionlistener> | Xử lý sự kiện nút chọn màu | EventPropertiesBar.java(20) |
| 2 | **adjustmentValueChanged(AdjustmentEvent e)**  Input: e  Output: không có  <https://examples.javacodegeeks.com/desktop-java/awt/event/adjustmentlistener-example/> | Xử lý sự kiện scroll chọn màu | EventPropertiesBar.java(77) |
| 3 | **actionPerformed(ActionEvent e)**  Input: e  Output: không có  <https://www.javatpoint.com/java-actionlistener> | Xử lý sự kiện nút chọn công cụ | EventToolBar.java(12) |
| 4 | **getCommand()**  Input: không có  Output: string | Trả về lệnh tương ứng với công cụ được chọn | EventToolBar.java(44) |
| 5 | **getCurrentX()**  Input: không có  Output: int | Trả về hoành độ con trỏ | EventWorkArea.java(27) |
| 6 | **getCurrentY()**  Input: không có  Output: int | Trả về tung độ con trỏ | EventWorkArea.java(30) |
| 7 | **setMyCanvas(MyCanvas canvas)**  Input: canvas  Output: không có | Đặt đối tượng MyCanvas | EventWorkArea.java(37) |
| 8 | **getShapeSlected()**  Input: không có  Output: int | Trả về index của đối tượng hình học | EventWorkArea.java(40) |
| 9 | **getMyCanvas()**  Input: không có  Output: MyCanvas | Trả về đối tượng MyCanvas | EventWorkArea.java(43) |
| 10 | **CicrleDirection()**  Input: không có  Output: MyCanvas | Tạo đối tượng hình tròn | EventWorkArea.java(253) |
| 11 | **RectangleDirection(int previousX,int previousY)**  Input: không có  Output: MyCanvas | Tạo đối tượng hình chữ nhật | EventWorkArea.java(265) |
| 12 | **Display(int lx, int ly)**  Input: lx,ly  Output: không có | Hiển thị cửa sổ chọn chế độ theo tọa độ | ConnectWindow.java(100) |
| 13 | **Initial()**  Input: không có  Output: không có | Khởi tạo cử sổ làm việc chính | MainWindow.java(30) |
| 14 | **Display()**  Input: không có  Output: không có | Hiển thị cửa sổ làm việc chính | MainWindow.java(38) |
| 15 | **ReSizeImg(String path, int newWidth, int newHeight)**  Input: đường dẫn, chiều rộng, chiều cao  Output: image  <https://stackoverflow.com/questions/6714045/how-to-resize-jlabel-imageicon> | Trả về image được điều chỉnh kích thước | MyButton.java(42) |
| 16 | **getNoShape()**  Input: không có  Output: int | Trả về số lượng đối tượng được vẽ | MyCanvas.java(29) |
| 17 | **getShapes()**  Input: không có  Output: List<Shape> | Trả về danh sách các đối tượng được vẽ | MyCanvas.java(33) |
| 19 | **setShapes(List<Shape> shapes)**  Input: shapes  Output: không có | Đặt danh sách các đối tượng | MyCanvas.java(37) |
| 20 | **drawShape(Shape shape)**  Input: shape  Output: không có | Tạo hiệu ứng vẽ, không lưu chính thức shape | MyCanvas.java(54) |
| 21 | **addShape()**  Input: không có  Output: không có | Bật cờ add, lưu chính thức shape | MyCanvas.java(62) |
| 22 | **remove()**  Input: không có  Output: không có | Xóa Shape mà được vẽ chỉ để tạo hiệu ứng | MyCanvas.java(75) |
| 23 | **moveShape(int index, int x, int y)**  Input: index, x, y  Output: không có | Thay đổi tọa độ của shape có index tương ứng trong list, theo trục ngang thêm x, theo trục đứng thêm y | MyCanvas.java(83) |
| 24 | **rotateShape(int index, float alpha)**  Input: index, alpha  Output: không có | Xoay shape có index tương ứng thêm góc quay alpha | MyCanvas.java(110) |
| 25 | **fillColor(int index, Color color)**  Input: index, color  Output: không có | Đổi màu shape có index tương ứng | MyCanvas.java(120) |
| 26 | **selectShape(int locationX, int locationY)**  Input: locationX, locationY  Output: index | Trả về index của shape mà chứa điểm có tọa độ truyền vào | MyCanvas.java(130) |
| 27 | **removeByIndex(int index)**  Input: index  Output: không có | Xóa shape có index tương ứng | MyCanvas.java(157) |
| 28 | **undo()**  Input: không có  Output: không có | Khôi phục trạng thái gần nhất trước đó của canvas | MyCanvas.java(170) |
| 29 | **redo()**  Input: không có  Output: không có | Khôi phục trạng thái gần nhất sau đó của canvas | MyCanvas.java(177) |
| 30 | **CreateMenu()**  Input: không có  Output: không có  <http://www.java2s.com/Tutorial/Java/0240__Swing/Verticalmenubar.htm> | Tạo thanh menu dọc | MyMenuBar.java(51) |
| 31 | **ExportPNG()**  Input: không có  Output: không có  <https://docs.oracle.com/javase/tutorial/2d/images/saveimage.html>  <https://www.mkyong.com/swing/java-swing-jfilechooser-example/> | Xuất ra file hình .png | MyMenuBar.java(110) |
| 32 | **SaveFile()**  Input: không có  Output: không có  <https://stackoverflow.com/questions/26559981/jpanel-snapshot-and-save-the-image-in-a-file-it-shows-black-image-what-to-do> | Lưu đối danh sách các đối tượng shape được vẽ thành file | MyMenuBar.java(126) |
| 33 | **OpenFile()**  Input: không có  Output: không có  <https://www.mkyong.com/swing/java-swing-jfilechooser-example/> | Mở file lưu danh sách các đối tượng shape | MyMenuBar.java(157) |
| 34 | **AddButton(ActionListener action)**  Input: action  Output: không có | Thêm button | MyToolBar.java(25) |
| 35 | **getColor()**  Input: không có  Output: color | Trả về màu được chọn | PropertiesBar.java(31) |
| 36 | **getStroke()**  Input: không có  Output: int | Trả về độ dày nét vẽ | PropertiesBar.java(34) |
| 37 | **getText()**  Input: không có  Output: string | Trả về văn bản được vẽ | PropertiesBar.java(37) |
| 38 | **show(int thickn, int anpha, int red, int green, int blue)**  Input: thickness, alpha, red, green, blue  Output: không có | Hiển thị các thuôc tính trên label mẫu | PropertiesBar.java(186) |
| 39 | **ShowGroup(boolean b)**  Input: boolean  Output: không có | Hiển thị button thêm shape vào group nếu giá trị truyền vào là true | PropertiesBar.java(191) |
| 40 | **setInfor(String object, int x, int y, int level, int maxLevel,int IDlabel,Boolean isgroup)**  Input: object, x, y, level, max level, id, isGroup  Output: không có | Đặt các thông tin được truyền vào cho các thuộc tính tương ứng của StatusBar | StatusBar.java(78) |
| 41 | **Move(int x, int y)**  Input: x,y  Output: không có | Đặt độ dời thêm x, y | IShapeMove.java(3) |
| 42 | **Rotate(float alpha)**  Input: alpha  Output: không có | Đặt góc xoay thêm alpha | IshapeRotation.java(3) |
| 43 | **getxLocation()**  Input: không có  Output: tọa độ x | Trả về hoành độ của shape | Shape.java(44) |
| 44 | **setxLocation(int xLocation)**  Input: x location  Output: không có | Đặt hoành độ cho shape | Shape.java(47) |
| 45 | **getyLocation()**  Input: không có  Output: tọa độ y | Trả về tung độ của shape | Shape.java(51) |
| 46 | **setyLocation(int yLocation)**  Input: đường dẫn, chiều rộng, chiều cao  Output: image | Đặt tung độ cho shape | Shape.java(54) |
| 47 | **getColor()**  Input: không có  Output: color | Trả về màu của shape | Shape.java(58) |
| 48 | **setColor(Color color)**  Input: color  Output: không có | Đặt màu của shape | Shape.java(61) |
| 49 | **getStroke()**  Input: không có  Output: stroke | Trả về độ dày nét vẽ shape | Shape.java(64) |
| 50 | **setStroke(int stroke)**  Input: stroke  Output: không có | Đặt độ dày nét vẽ shape | Shape.java(67) |
| 51 | **Draw(Graphics2D g)**  Input: graphic2d  Output: không có  <https://www.ntu.edu.sg/home/ehchua/programming/java/J4b_CustomGraphics.html> | Vẽ đối tượng | Shape.java(79) |
| 52 | **Server()**  Input: không có  Output: không có  <https://www.geeksforgeeks.org/socket-programming-in-java/> | Xử lí chế độ server | NetWork.java(84) |
| 53 | **Client()**  Input: không có  Output: không có  <https://www.geeksforgeeks.org/socket-programming-in-java/> | Xử lý chế độ client | NetWork.java(113) |
| 54 | **Update(List<Shape> shapes)**  Input: list shape  Output: không có  <https://www.geeksforgeeks.org/socket-programming-in-java/> | Đồng bộ bản vẽ đến nhóm client | NetWork.java(151) |
| 55 | **UpdateToServer(List<Shape> shapes)**  Input: list shape  Output: không có  <https://www.geeksforgeeks.org/socket-programming-in-java/> | Đồng bộ bản vẽ đến server | NetWork.java(171) |
| 56 | **Update(List<Shape> shapes)**  Input: list shape  Output: không có  <https://www.geeksforgeeks.org/socket-programming-in-java/> | Đồng bộ bản vẽ đến mỗi client | ClientHandler.java(78) |
| 57 | **IsOnShape(int x, int y)**  Input: x, y  Output: true or false  Pseudo code:  If d((x,y) , (I))<=r then true  Else fale | Kiểm tra tọa độ truyền vào có nằm trên hình tròn | Circle.java(17) |
| 58 | **Draw(Graphics2D g)**  Input: g  Output: không có  <https://www.ntu.edu.sg/home/ehchua/programming/java/J4b_CustomGraphics.html> | Vẽ hình tròn | Circle.java(23) |

*Bảng 5: Các phương thức trong project*

## Cài đặt kiểm thử

### Lập kế hoạch

#### Mục tiêu

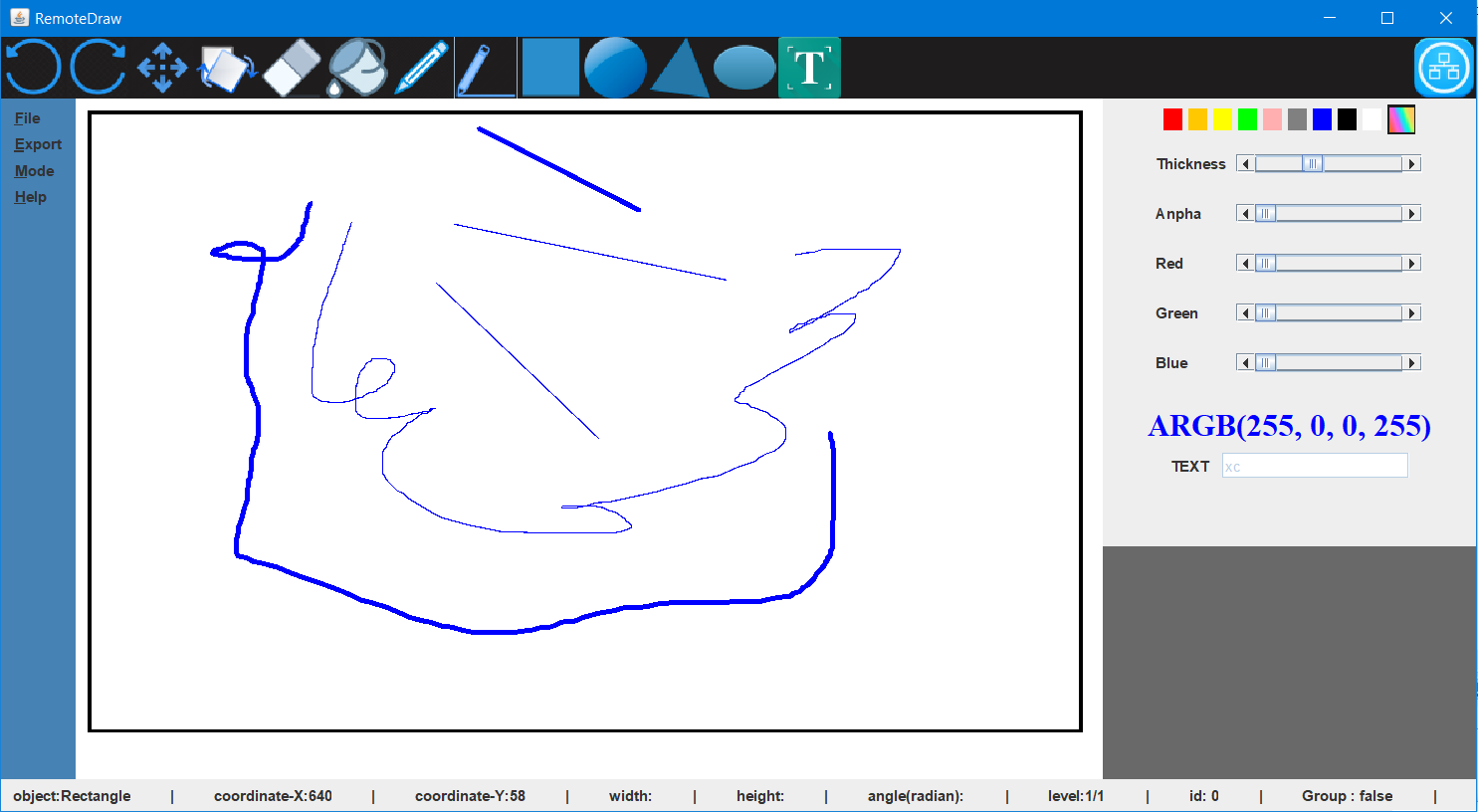
Đáp ứng được các tính năng đã xây dựng

#### Môi trường

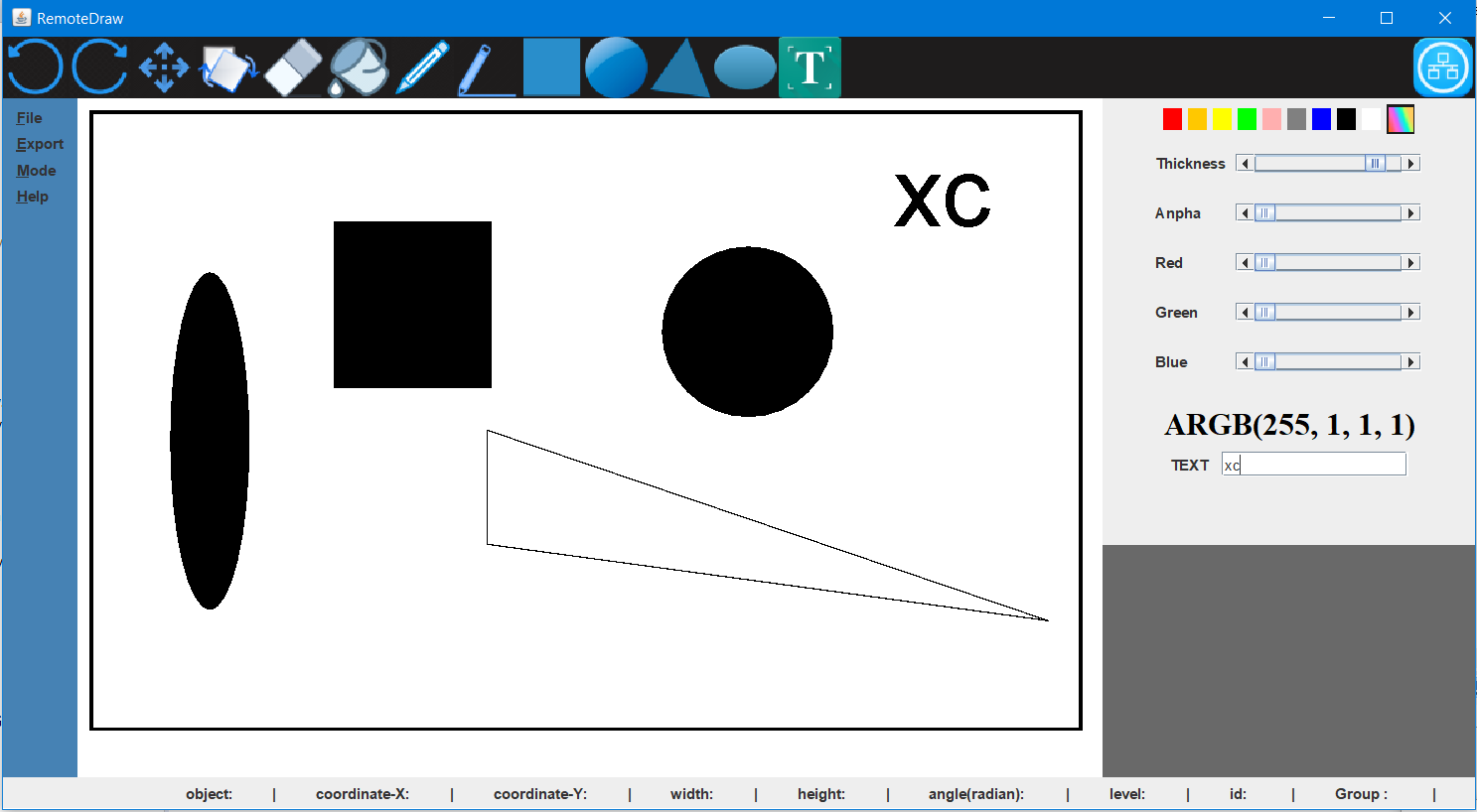
Hệ điều hành Windows

### Thực thi

#### Trường hợp 1: dùng các công cụ vẽ

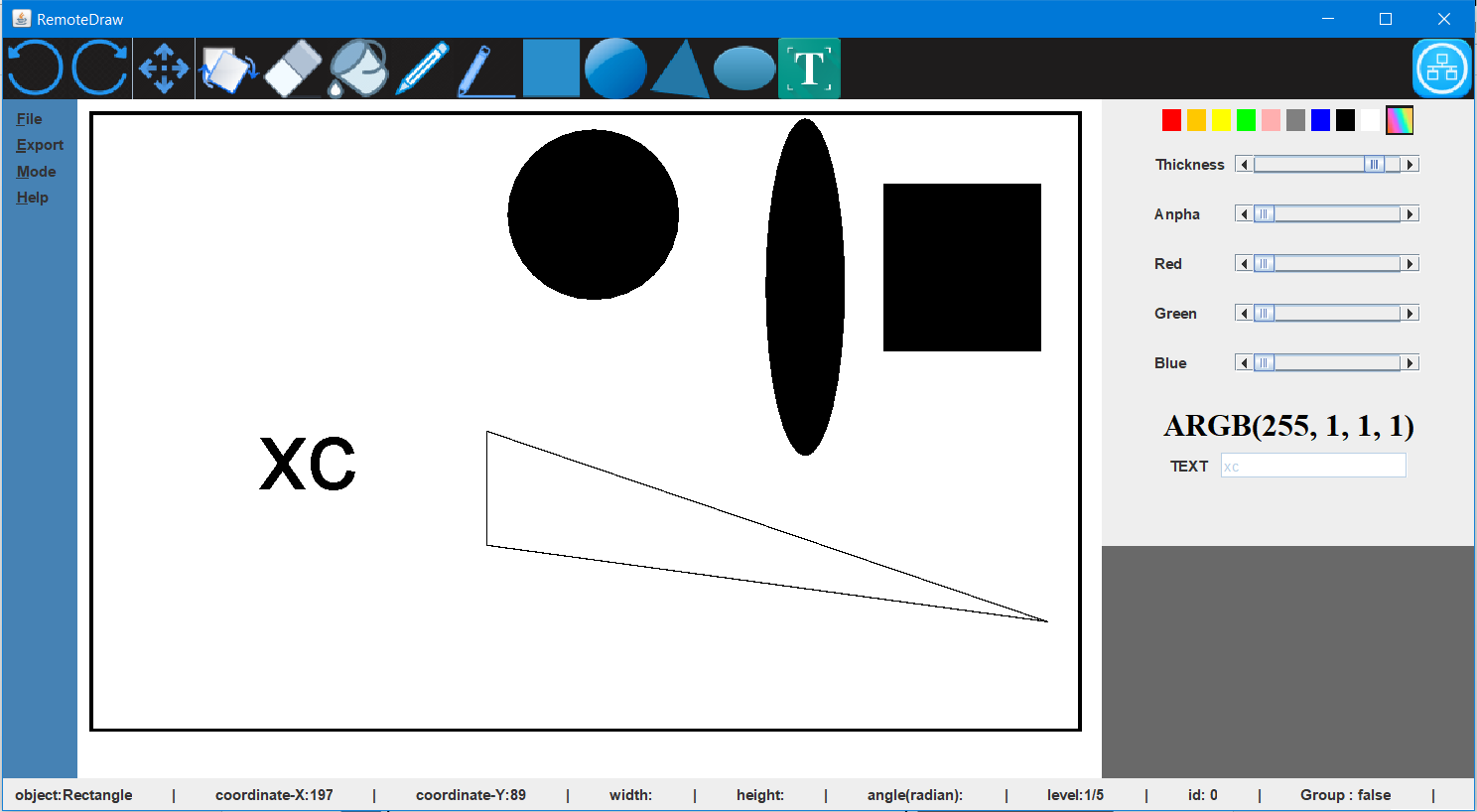
**

*Hình 3: Vẽ line, pencil*

**

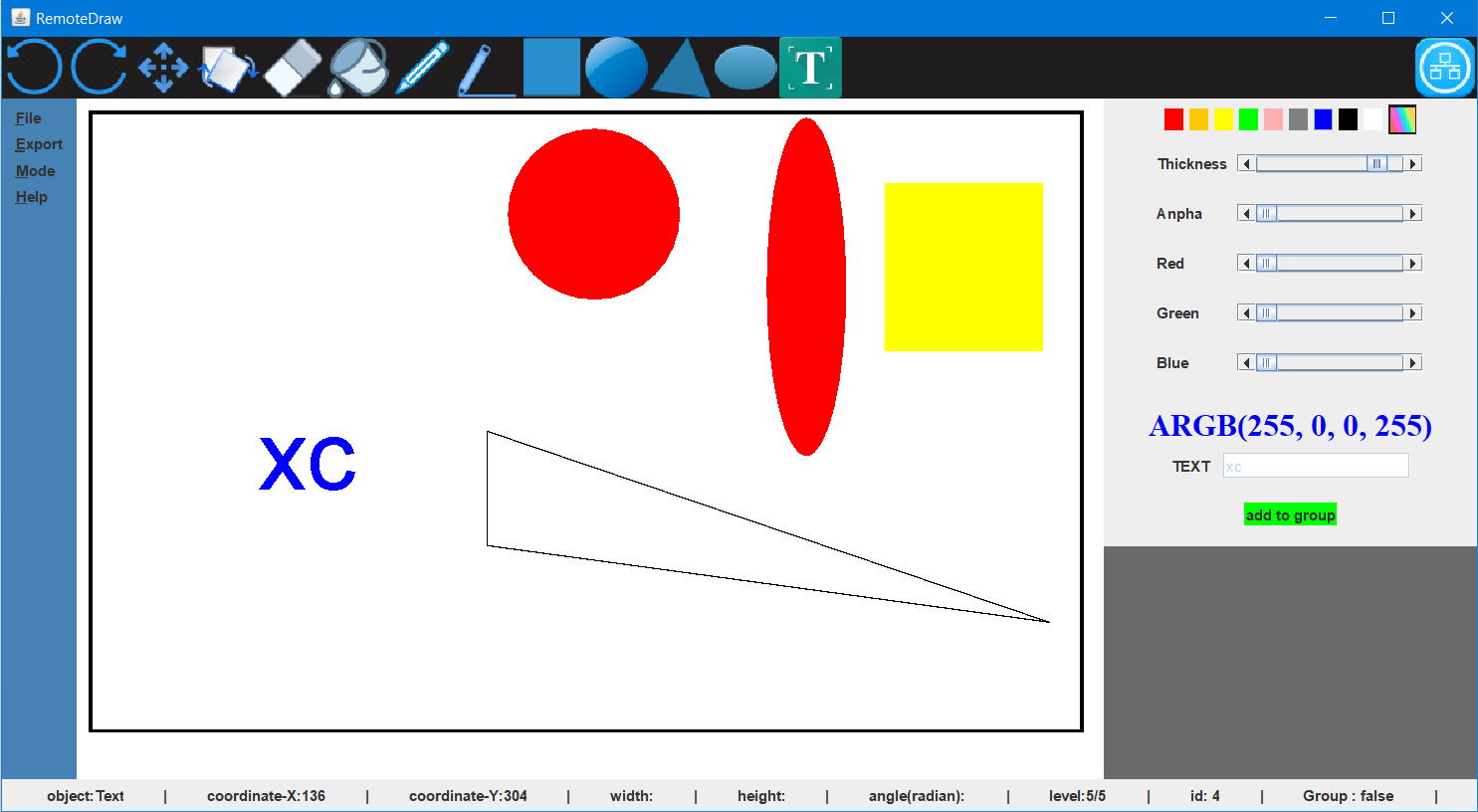
*Hình 4: Test các chế độ vẽ rectangle, circle, elipse, triangle, text.*

#### Trường hợp 2: dùng công cụ di chuyển

**

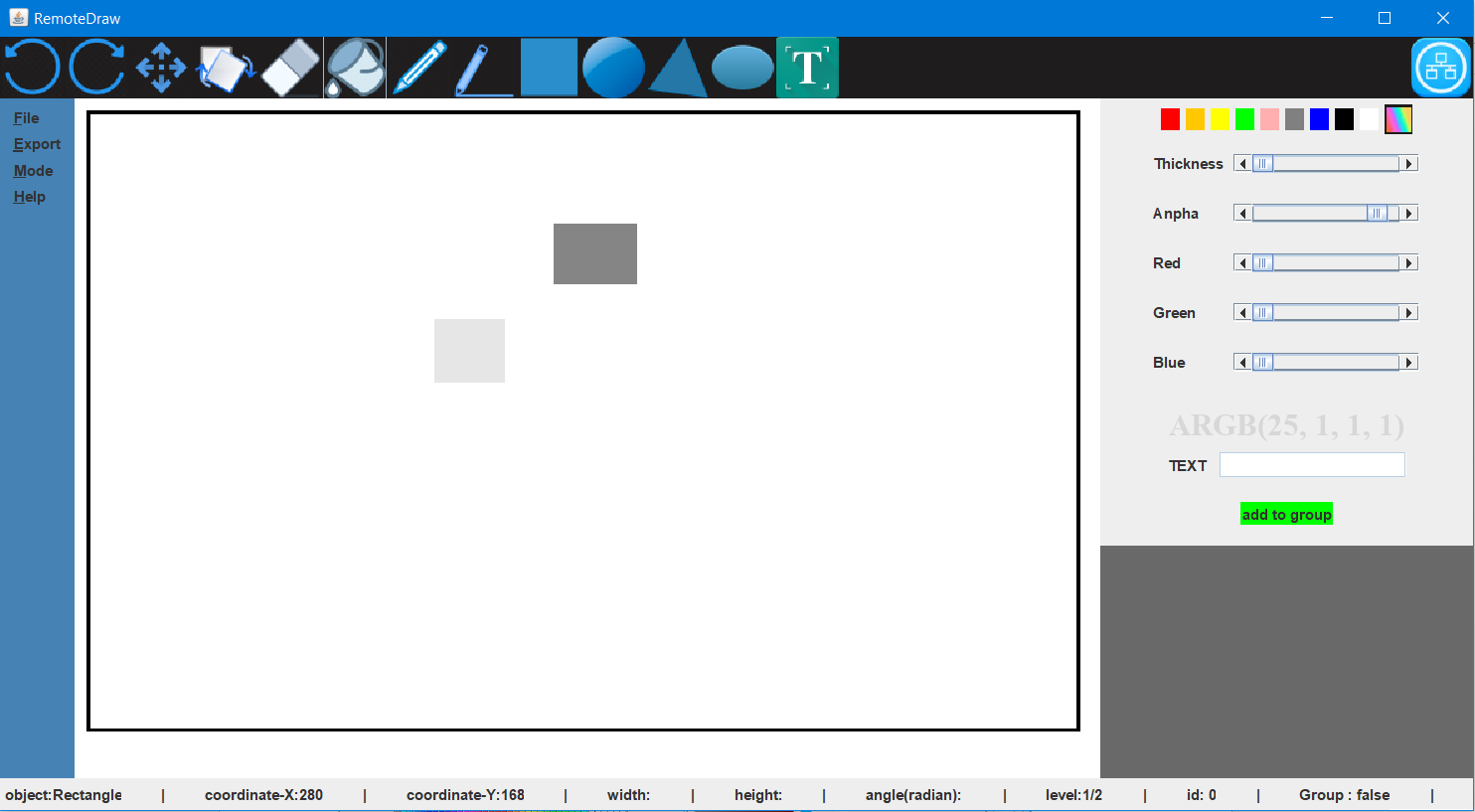
*Hình 5: Di chuyển các hình ảnh đã vẽ*

#### Trường hợp 3: dùng công cụ đổi màu

**

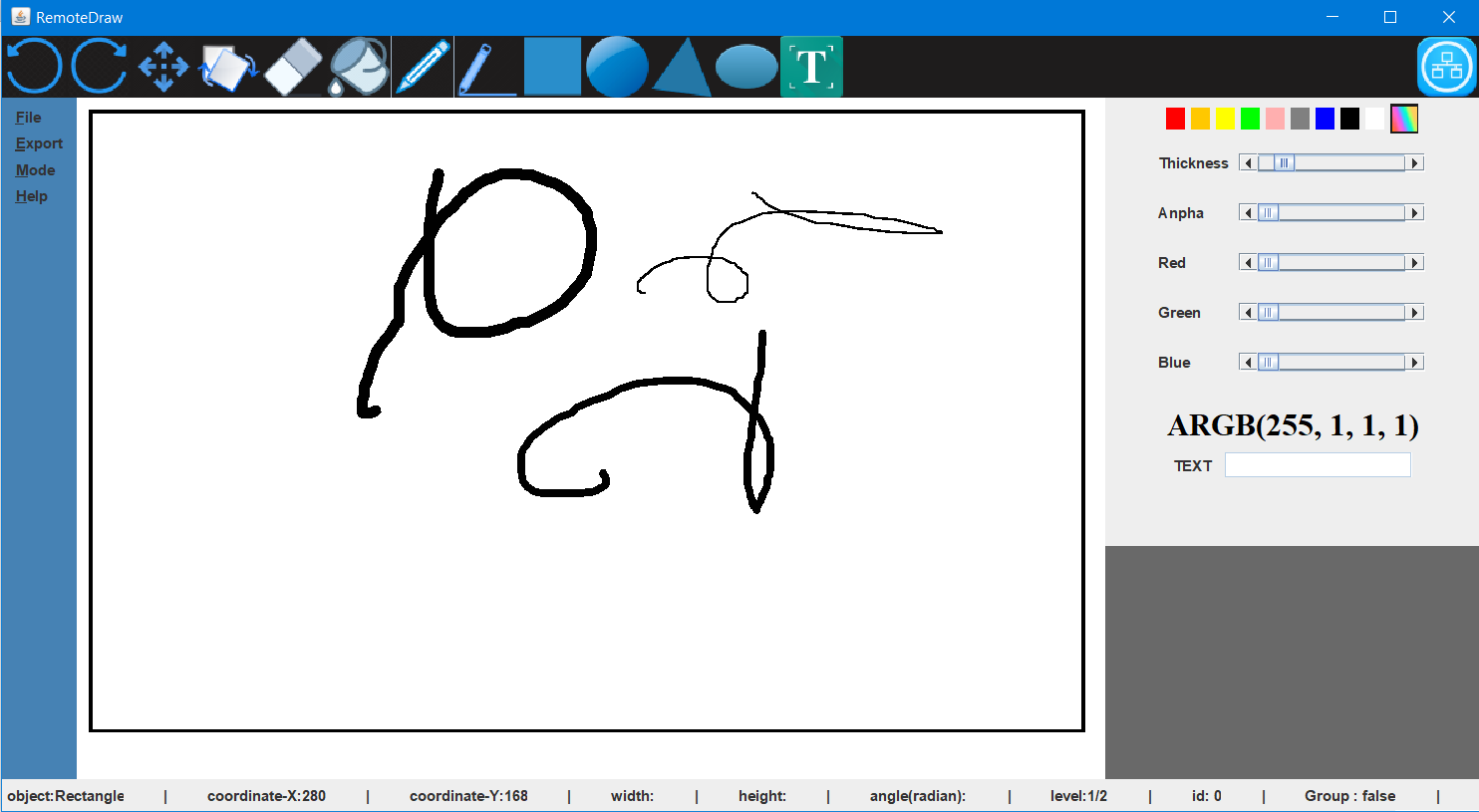
*Hình 6: Đổ màu lên hình*

#### Trường hợp 4: thay đổi độ trong suốt



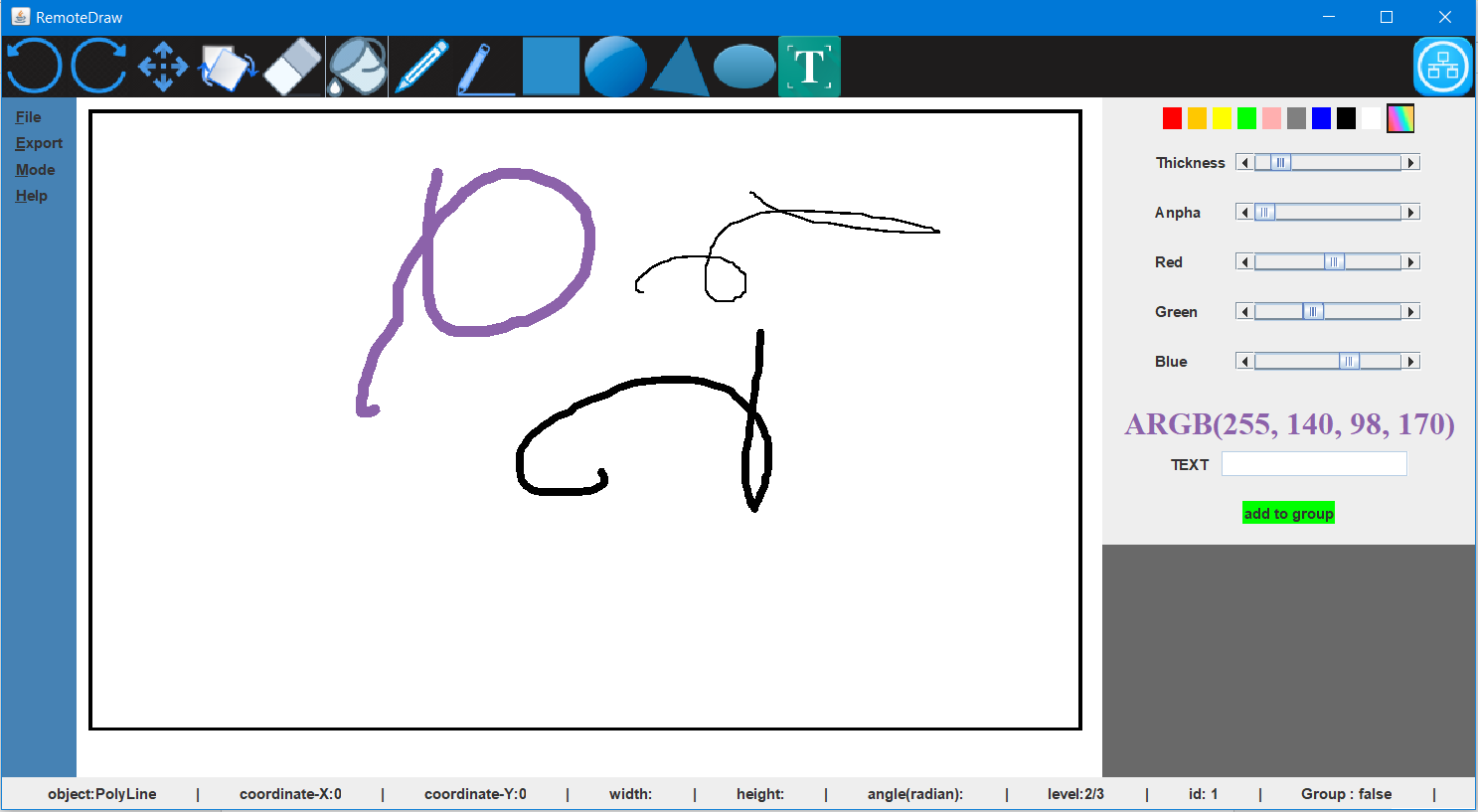
*Hình 7: Thay đổi độ trong suốt*

#### Trường hợp 5: thay đổi độ dày nét vẽ



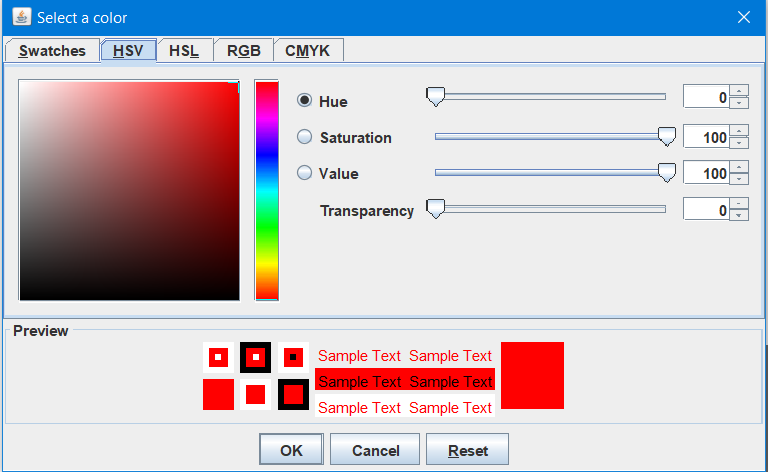
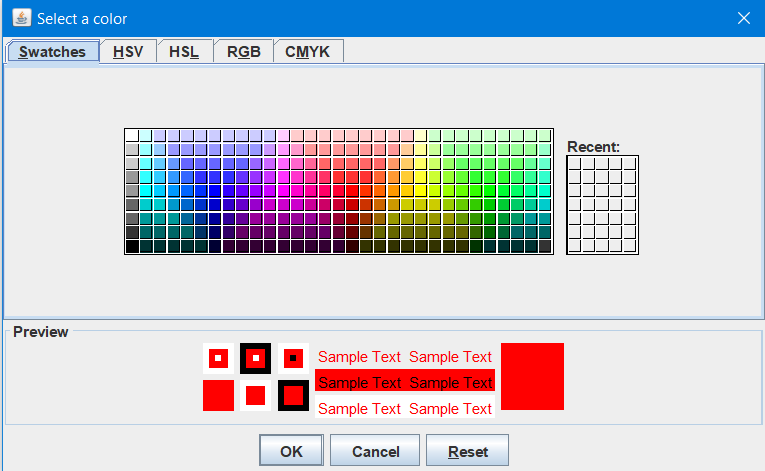
*Hình 8: Thay đổi độ dày nét vẽ*

#### Trường hợp 6: thay đổi màu theo hệ ARGB



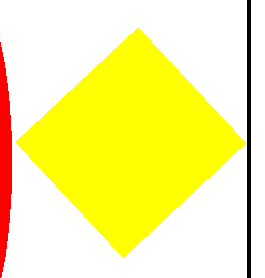
*Hình 9: Thay đổi màu theo hệ ARGB*

#### Trường hợp 7: thay đổi màu theo bảng màu mở rộng

**

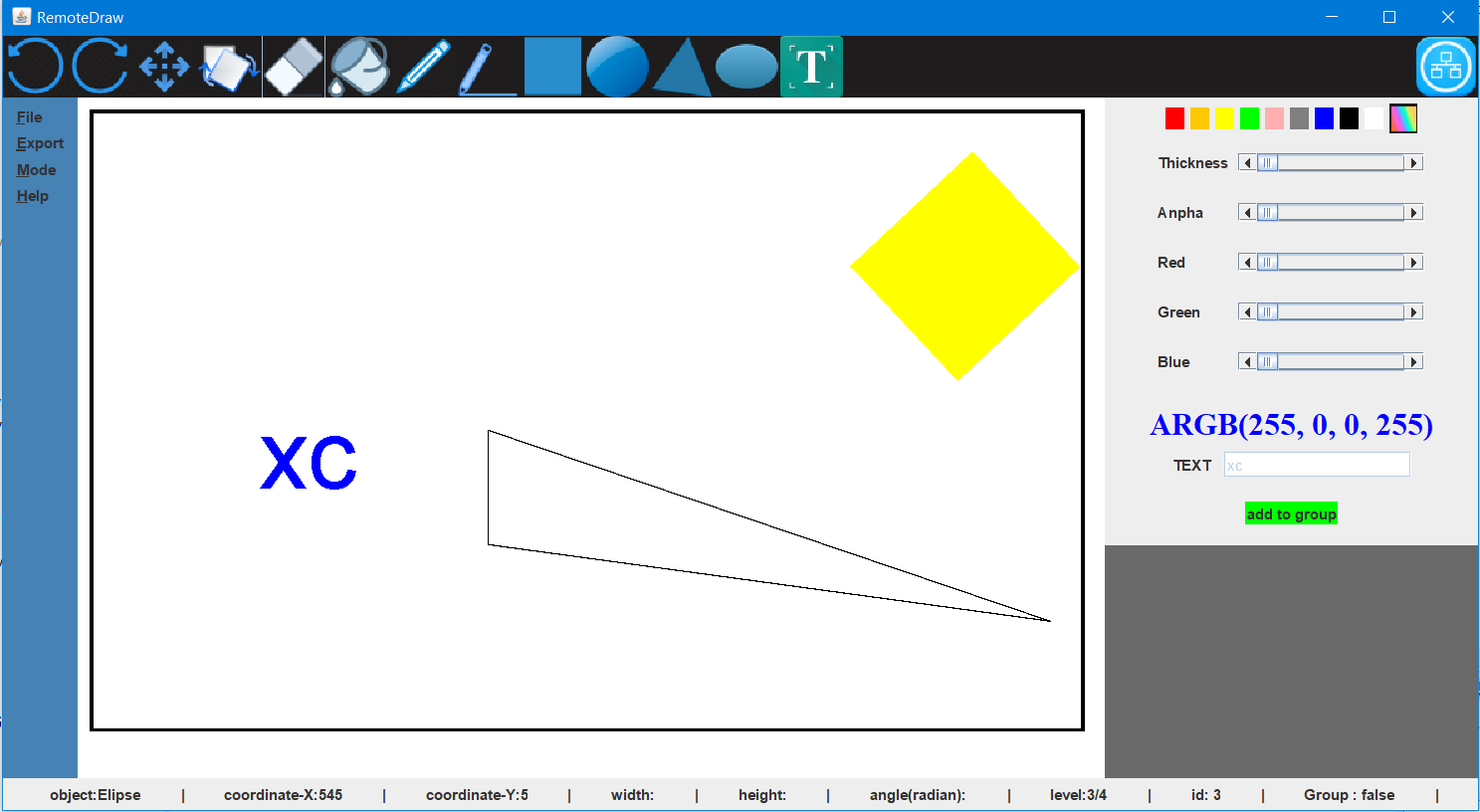
*Hình 10,11: Bảng màu mở rộng*

#### Trường hợp 8: dùng chức năng xoay hình (chỉ áp dụng đối với hình vuông)

**

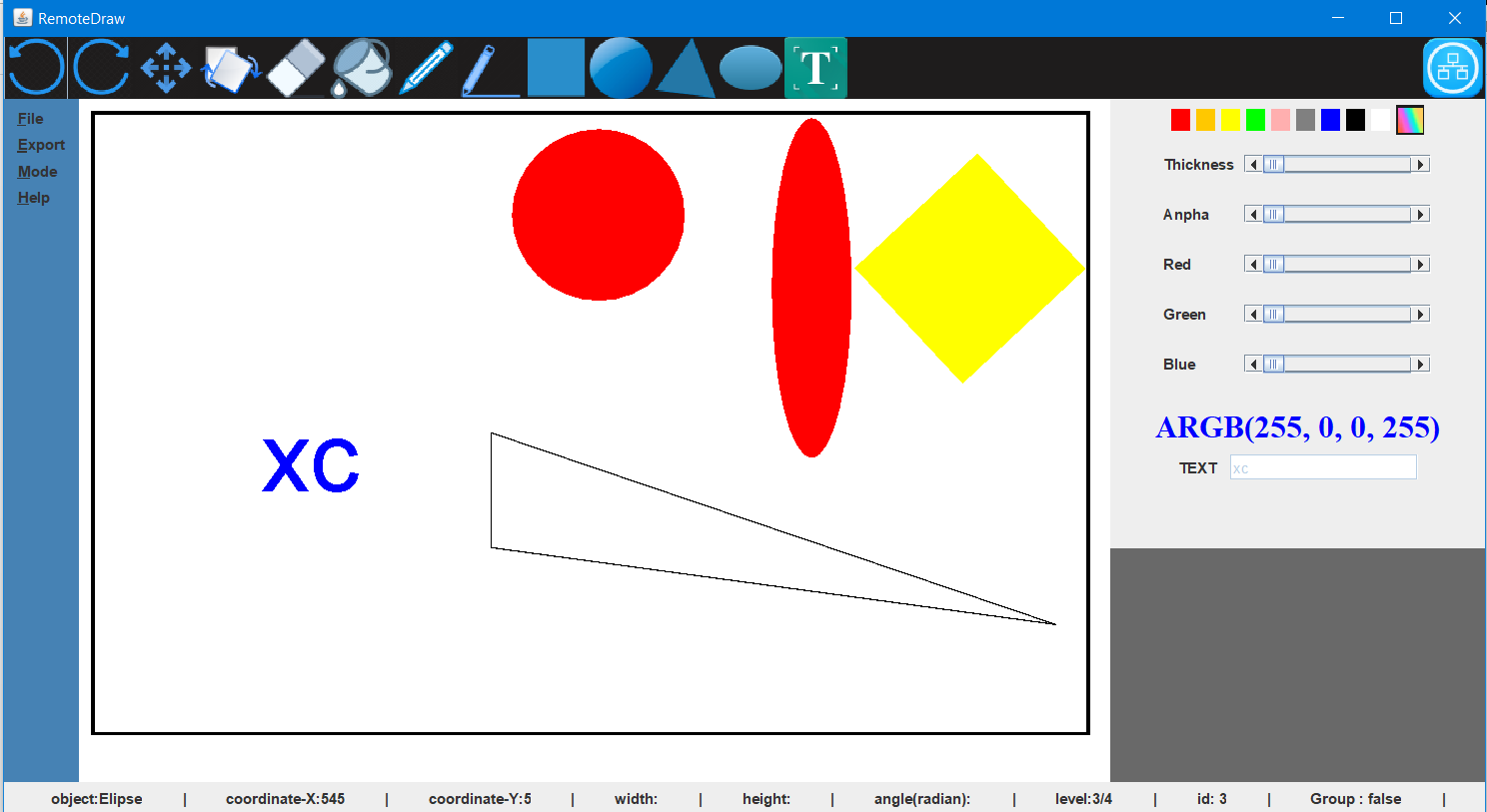
*Hình 12: Chức năng xoay hình (rotate)*

#### Trường hợp 9: xóa hình

**

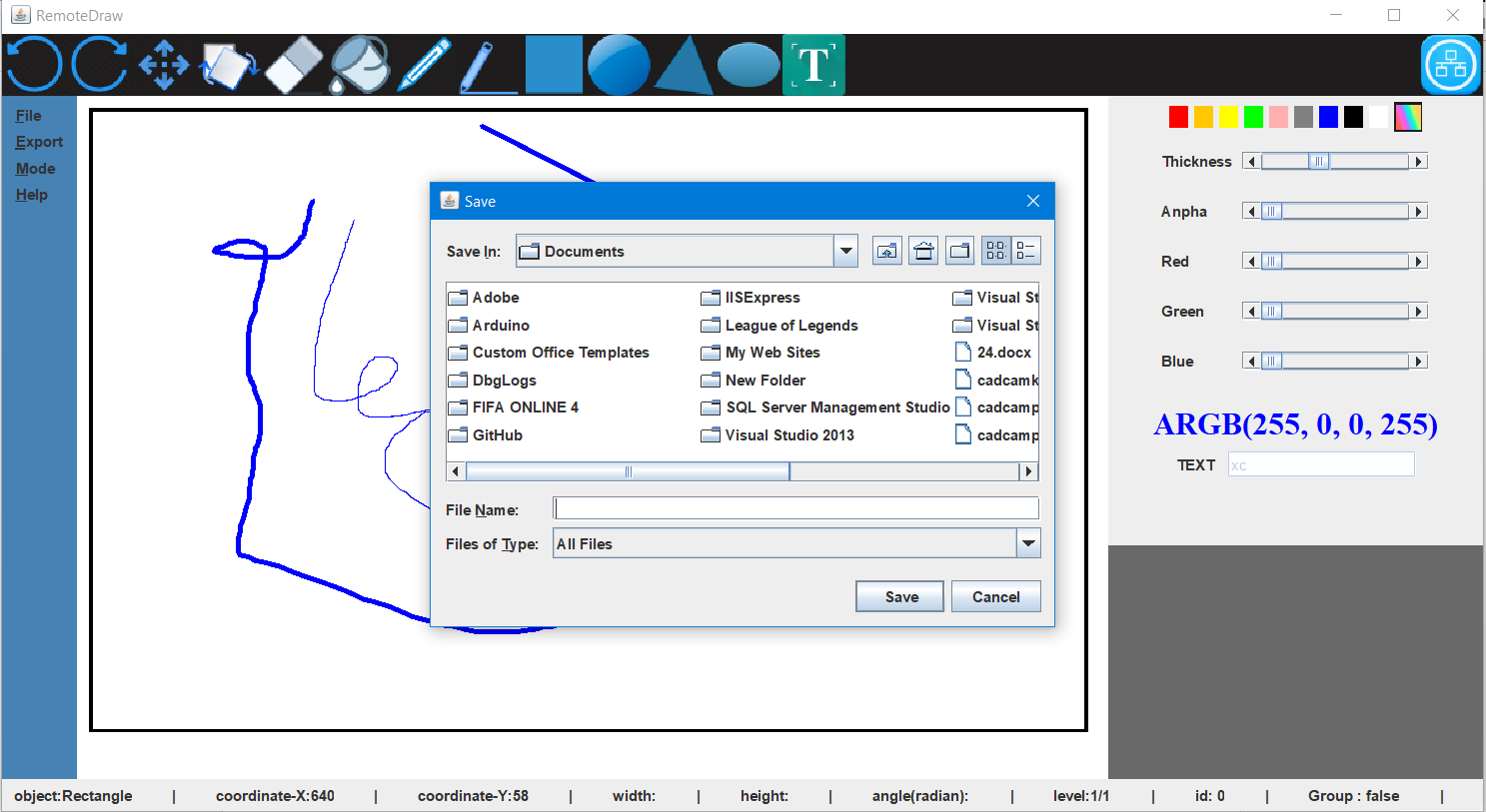
*Hình 13: Chức năng cục tẩy*

#### Trường hợp 10: chức năng undo

**

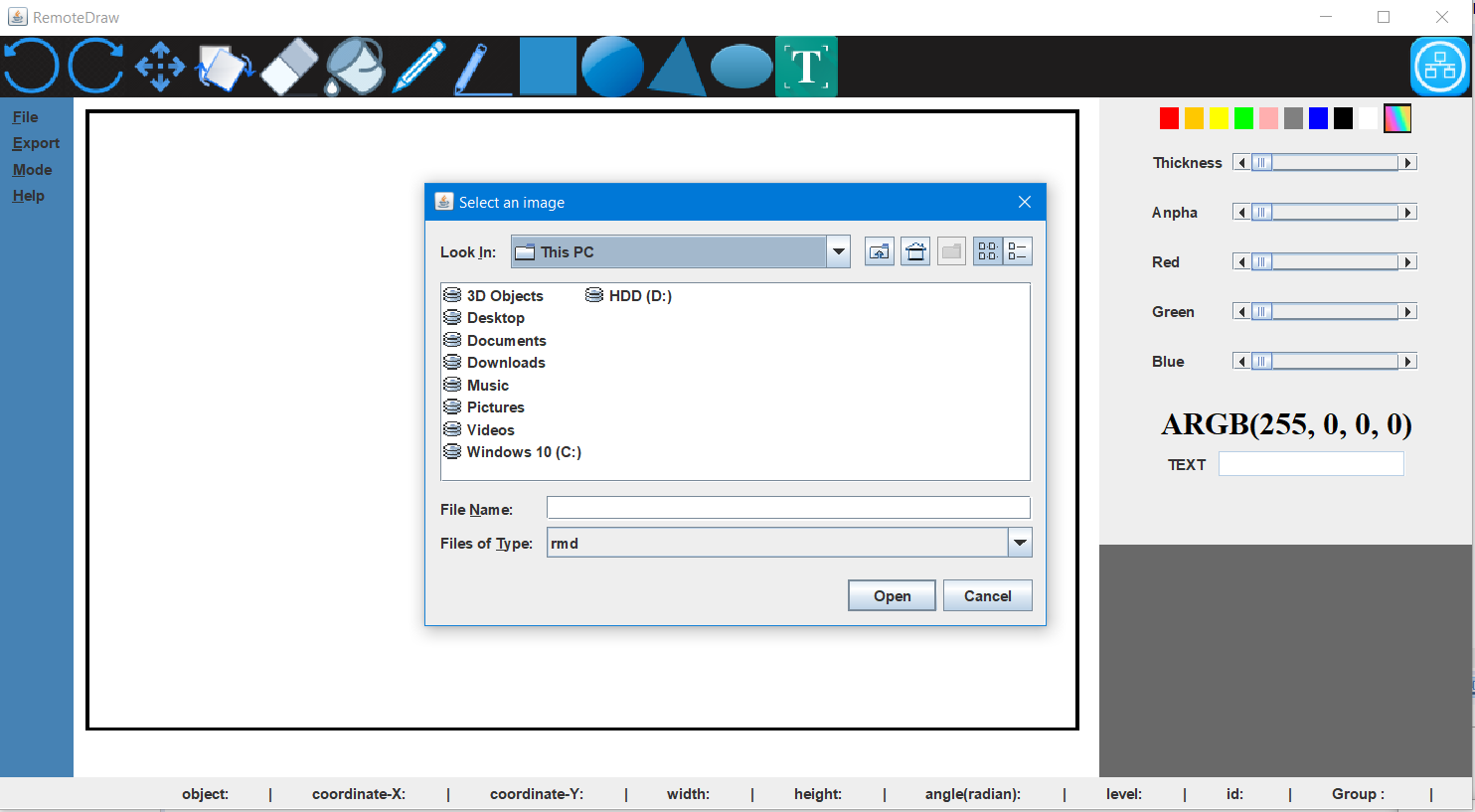
*Hình 14: Chức năng undo*

#### Trường hợp 11: dùng chức năng save



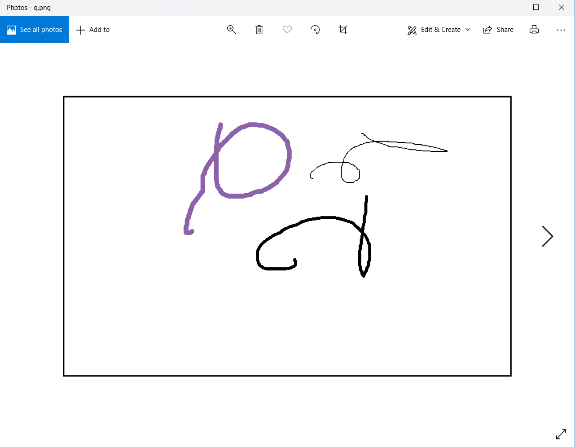
*Hình 15: Chức năng save*

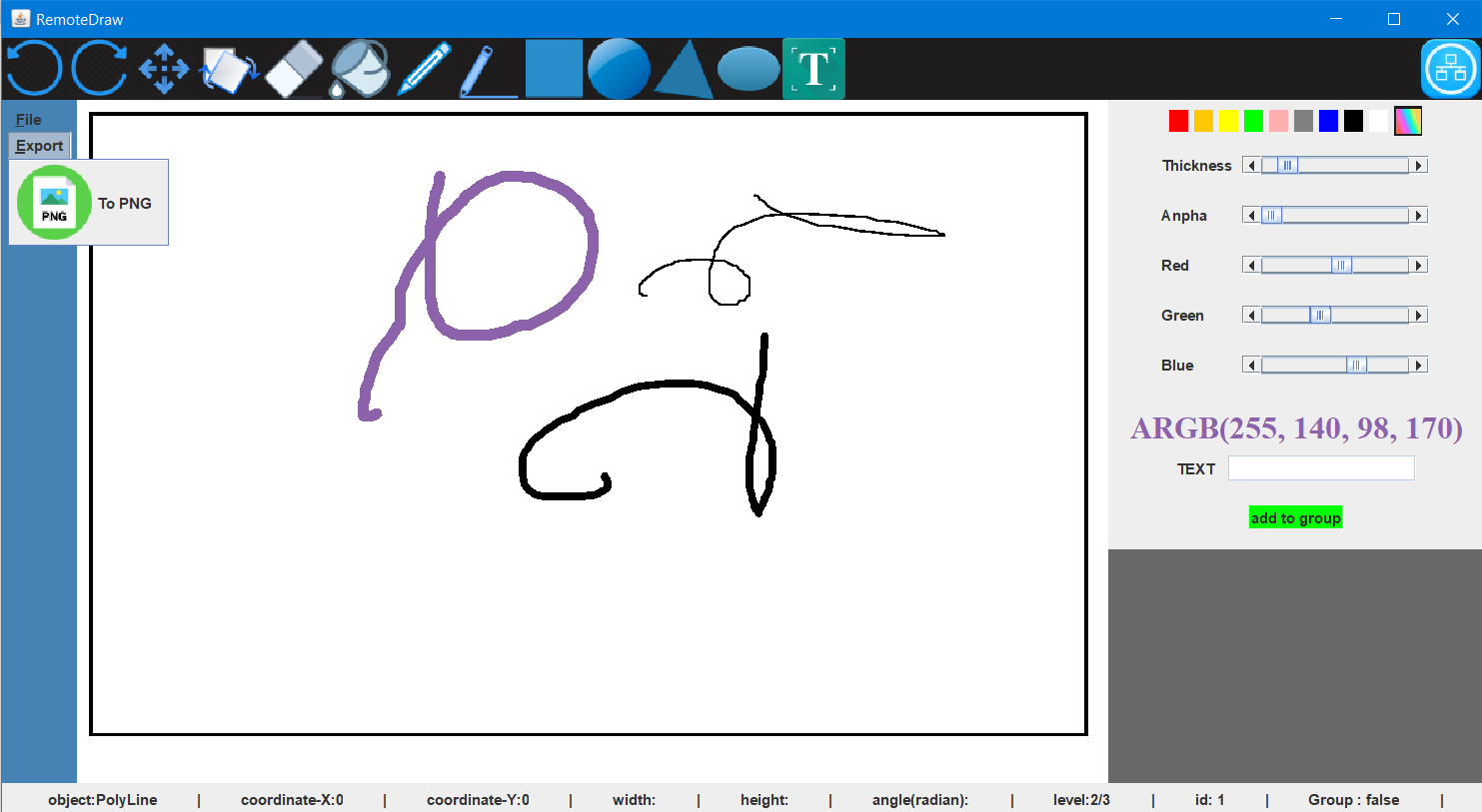
#### Trường hợp 12: dùng chức năng open



*Hình 16: Chức năng open*

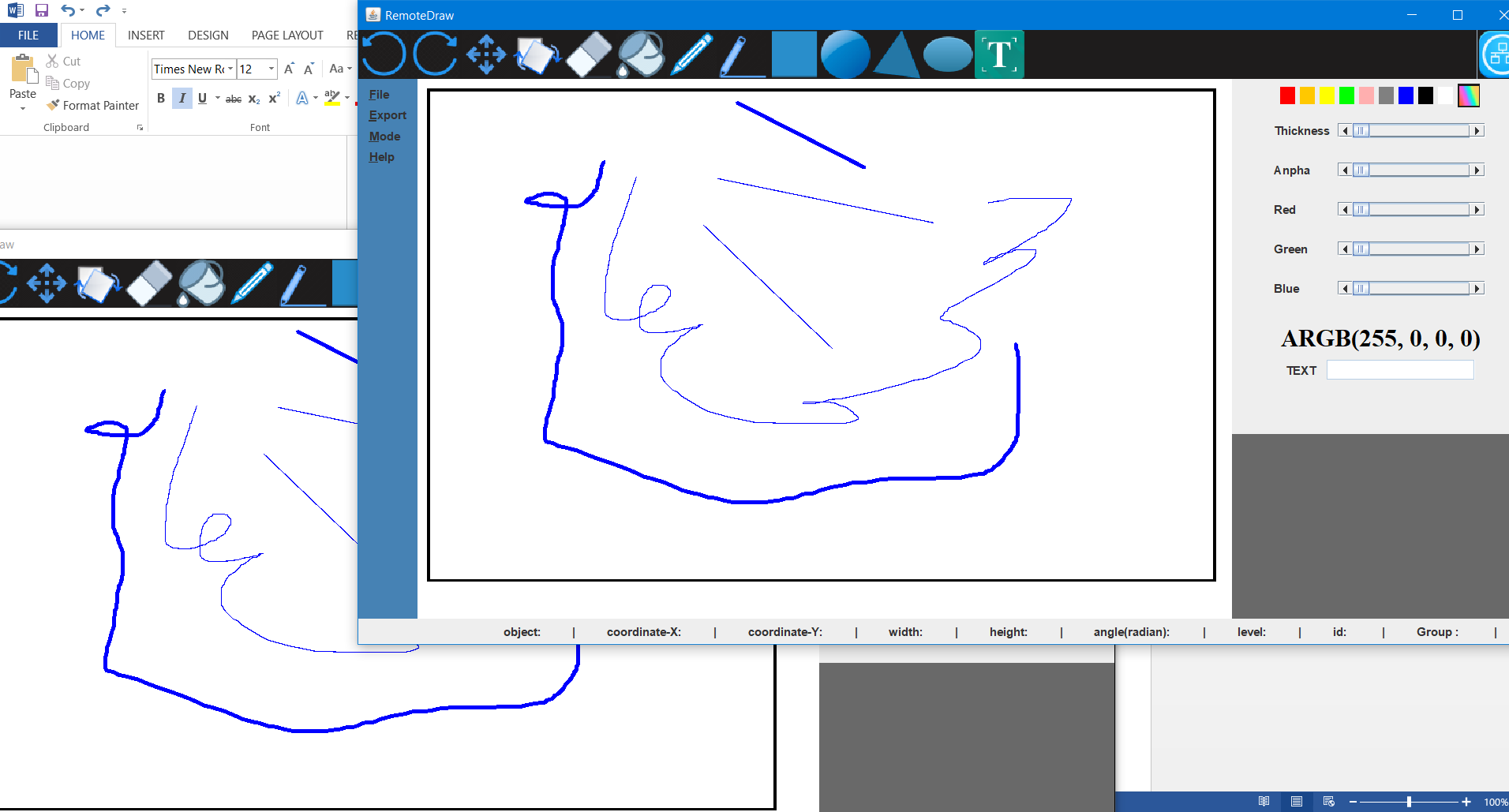
#### Trường hợp 13: dùng chức năng xuất ra file .png





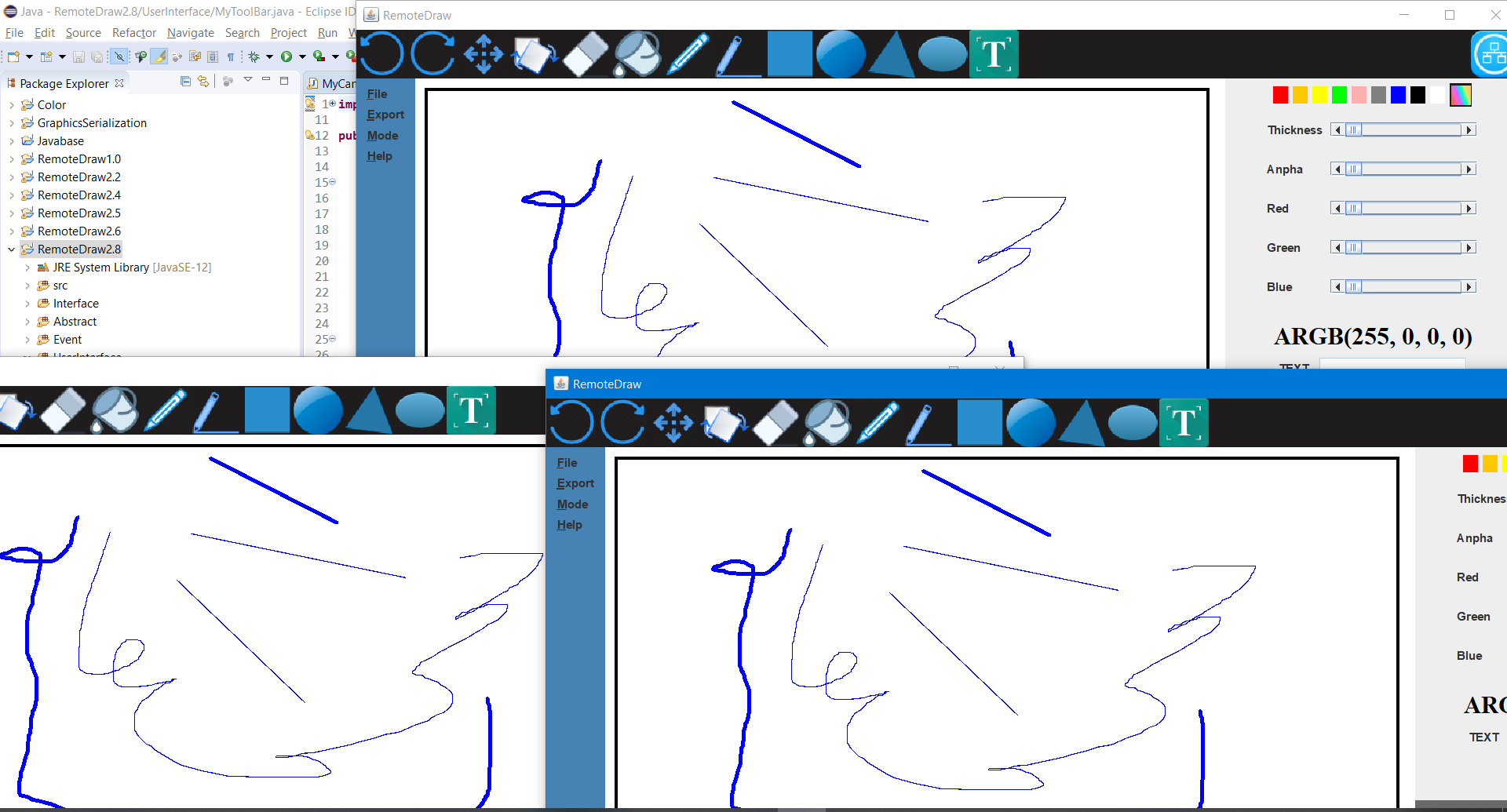
*Hình 17,18: Dùng chức năng xuất ra file .png*

#### Trường hợp 14: hoạt động với mode server - client



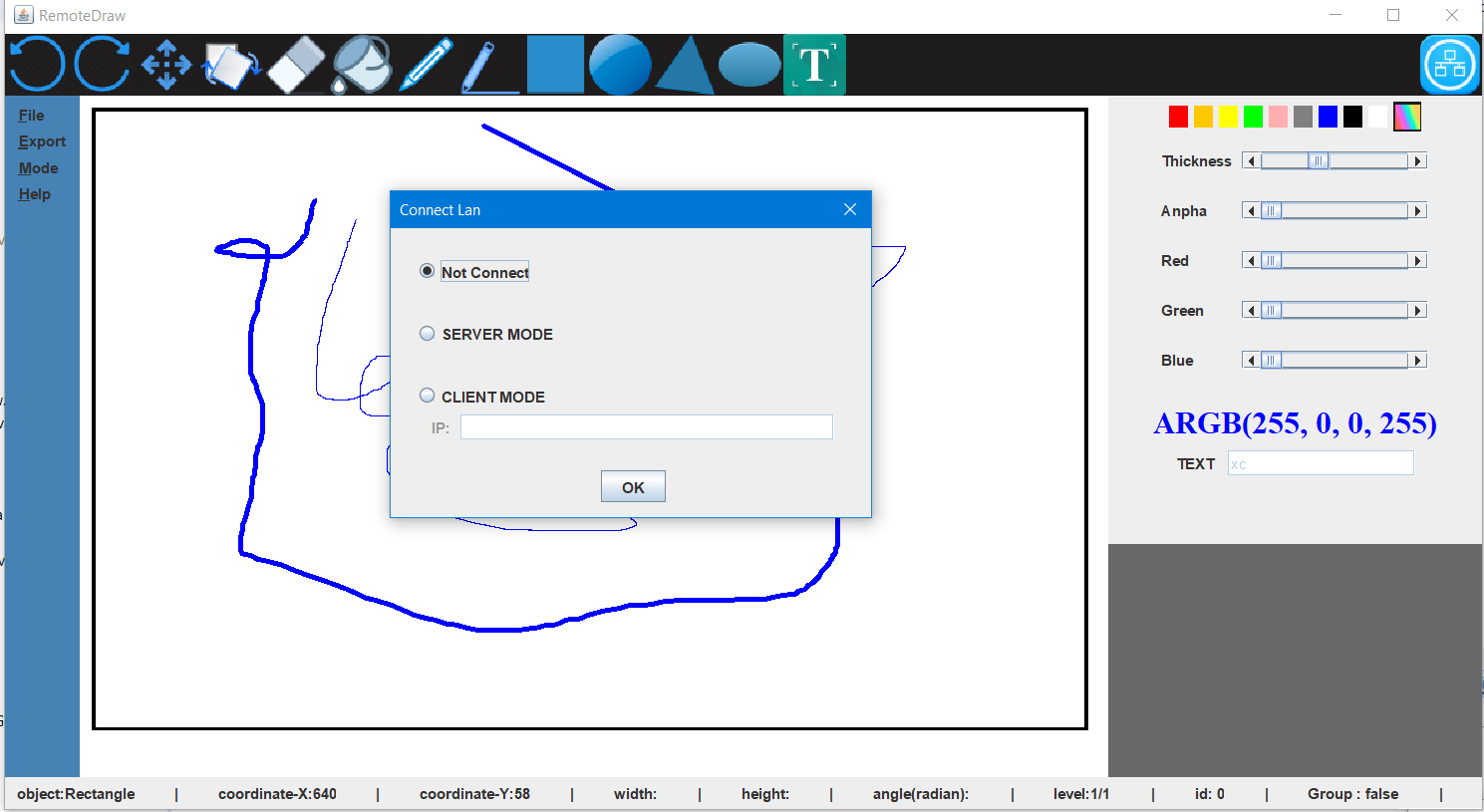
*Hình 19: Hoạt động với mode server-client*

#### Trường hợp 15: hoạt động với mode server 1– n client



*Hình 20: Hoạt động với mode server 1-n client*

#### Trường hợp 16: đổi chế độ



*Hình 21: Đổi chế độ*

#### Trường hợp 17: máy hoạt động với mode client bị tắt

#### Trường hợp 18: máy hoạt động với mode server bị tắt

### Đánh gía

#### Mức độ đáp ứng

Số trường hợp đáp ứng là 16 (trường hợp 1 đến 16) trong tổng số 18 trường hợp kiểm thử, đáp ứng 89%

Số trường hợp chưa đáp ứng là 3 (trường hợp 17, 18) trong tổng số 18 trường hợp kiểm thử, chưa đáp ứng 11%

## Kết luận

### Kết quả đạt được

Kết quả đạt được dựa trên các mục tiêu ban đầu đề ra, sau khi thực hiện là một chương trình vẽ được những hình học cơ bản và gửi chúng qua lại giữa mạng LAN với nhau. Hiểu rõ được các tình kế thừa, abstract, interface, network trong lập trình hướng đối tượng và sử dụng được ngôn ngữ java.

### Kết luận

#### Ưu điểm

- Hoàn thành được quá trình liên kết giữa các User với nhau và vẽ cùng nhau trên mạng LAN.

- Giao diện đơn giản, rõ ràng, dễ sử dụng.

#### Nhược điểm

- Chỉ vẽ được những hình cơ bản, không add picture từ bên ngoài vào được.

- Vẽ 1 số hình không hoàn toàn chính xác bị lệch 1 phần nhỏ tọa độ.

- Chưa thể đổi chế độ từ server sang client và ngược lại.

### Đề xuất cải thiện

- Xây dựng thêm tính năng như zoom hình, select nhiều hình trong 1 lần, brushes, add picture,..

- Thêm chức năng chatbox giữa các user đang làm việc với nhau.

- Tối ưu tính chính xác trong việc vẽ hình.

- Kiểm soát kết nối mạng linh hoạt hơn (có thể chuyển đổi chế độ)

- Chỉnh sửa lỗi khi 1 máy chuyển chế độ kết nối.

- Chỉnh sửa lỗi trên các máy chạy chế độ client khi máy server bị tắt