

# Đồ họa trong J2ME

GV: ThS. Phan Nguyệt Minh  
[minhpn@uit.edu.vn](mailto:minhpn@uit.edu.vn)

<http://courses.uit.edu.vn>

# Đồ họa cấp thấp

- Giao diện cấp thấp ( low-level interfaces) : dùng cho các ứng dụng game, được hỗ trợ qua 2 lớp **Canvas** và **Graphics**.
- Lớp **Canvas** có dạng một phong cố định, những gì được vẽ lên phong sẽ được hiển thị cho người dùng, hỗ trợ các phương thức bất sự kiện cấp thấp
- Lớp **Graphics** cung cấp đối tượng để vẽ lên Canvas, hỗ trợ vẽ những đường thẳng, đường tròn... cũng như font, màu chữ.

# Canvas

- ▶ Lớp **Canvas**: cung cấp một khung vẽ tạo giao diện tùy biến người dùng. Đa số các phương thức trong lớp này để xử lý sự kiện, vẽ ảnh và chuỗi lên thiết bị hiển thị.
  - ▶ Hệ thống tọa độ
  - ▶ Tạo đối tượng Canvas
  - ▶ Vẽ lên trên đối tượng Canvas
  - ▶ Xử lý các sự kiện hành động
  - ▶ Xử lý các sự kiện phím nhấn
  - ▶ Xử lý sự kiện hành động của Game
  - ▶ Xử lý sự kiện con trỏ

# Hệ trục tọa độ

- ▶ Hệ tọa độ cho lớp Canvas: tâm tọa độ là điểm trái trên của thiết bị. Trị x tăng dần về phải, trị y tăng dần khi xuống dưới. Độ dày bút vẽ là một điểm ảnh.
- ▶ Các phương thức sau đây sẽ giúp xác định chiều rộng và chiều cao của canvas:
  - ▶ `int getWidth()`: xác định chiều rộng của canvas
  - ▶ `int getHeight ()`: xác định chiều cao của canvas

# Tạo đối tượng Canvas

Đầu tiên tạo ra một lớp thừa kế từ lớp Canvas

class TestCanvas extends Canvas implements  
CommandListener

```
{  
    private Command cmdExit;  
    display = Display.getDisplay(this);  
    cmdExit = new Command("Exit", Command.EXIT, 1);  
    addCommand(cmdExit);  
    setCommandListener(this);  
}
```

# Tạo đối tượng Canvas

```
protected void paint(Graphics g)
{
    // Draw onto the canvas
    // Set background color to white
    g.setColor(255, 255, 255);
    // Fill the entire canvas
    g.fillRect(0, 0, getWidth(), getHeight ());
}
}
TestCanvas canvas = new TestCanvas(this);
```

Phương thức paint của lớp Canvas cho phép bạn vẽ các hình dạng, vẽ ảnh, xuất chuỗi

# Sự kiện hành động

- ▶ Hệ tọa độ cho lớp Canvas: tâm tọa độ là điểm. Một **Canvas** có thể xử lý các Command.
- ▶ Mã phím: trường hợp xử lý các hành động của các phím mềm, một Canvas có thể truy cập đến 12 mã phím. Những mã này được đảm bảo luôn luôn có trên bất kỳ các thiết bị MIDP nào  
**KEY\_NUM0, KEY\_NUM1, KEY\_NUM2 ...**

# Sự kiện hành động

- Các phương thức để xử lý các mã phím là:  
`void keyPressed(int keyCode);`  
`void keyReleased(int keyCode);`  
`void keyRepeat(int keyCode);`  
`String getKeyName(int keyCode);`



# Các hành động xử lý trong game

- ▶ MIDP thường được sử dụng để tạo các trò chơi trên nền Java. Các hằng số sau đã được định nghĩa để xử lý các sự kiện có liên quan đến trò chơi trong MIDP  
UP, DOWN, LEFT, RIGHT, FIRE, GAME\_A, GAME\_B, GAME\_C, GAME\_D

# Các hành động xử lý trong game

- ▶ Đơn giản thì các giá trị này được ánh xạ thành các phím mũi tên chỉ hướng của thiết bị, nhưng không phải tất cả các thiết bị di động đều có những giá trị này. Nếu một thiết bị di động thiếu các phím mũi tên thì các hành động của trò chơi sẽ được ánh xạ vào các nút bấm, ví dụ phím trái được ánh xạ vào phím số 2, phím phải được ánh xạ vào phím số 5, và cứ tiếp tục như thế.

# Cách xác định hành động của game

```
protected void keyPressed(int keyCode) {  
    int gameAction = getGameAction(keyCode);  
    switch(gameAction) {  
        case UP:      mMessage = "UP";  
                        break;  
        case DOWN:    mMessage = "DOWN";  
                        break;  
        case LEFT:    mMessage = "LEFT";  
                        break;  
        case RIGHT:   mMessage = "RIGHT";  
                        break;  
        case FIRE:    mMessage = "FIRE";  
                        break;  
        ....  
        default:      mMessage = "";  
                        break;  
    }  
}
```

# Graphics

- ▶ Đối tượng **Graphics** được dùng để vẽ lên một Canvas.
  - ▶ `boolean isColor();` // thiết bị có hỗ trợ hiển thị màu không?
  - ▶ `int numColors();` // gọi để xác định số màu
    - ▶ Mặc định (`DefaultColorPhone`) là 4096 colors (0x1000)
    - ▶ `isColor=true; colorCount=0x1000;`

# Graphics

- Các phương thức lấy về màu và thiết lập màu:

*//* Đặt màu hiện thời qua giá trị RGB

**void setColor(int RGB);**

*//* đặt màu, các giá trị red, green.. từ 0-255

**void setColor(int red, int green, int blue);**

*//* trả về màu hiện thời

**int getColor();**

# Graphics

// trả về thành phần màu xanh da trời của màu hiện thời 0-255

int getBlueComponent();

// trả về thành phần màu lục của màu hiện thời 0-255

int getGreenComponent();

// trả về thành phần màu đỏ của màu hiện thời 0-255

int getRedComponent();

# Graphics

// đặt mức xám

void setGrayScale(int value);

// trả về giá trị xám từ 0-255

int getGrayScale();

Ví dụ: BLACK = 0; WHITE = 0xffffffff; RED = 0xf96868;  
GREY = 0xc6c6c6; LT\_GREY = 0xe5e3e3;

Hay int red = 0, green = 128, blue = 255;

Sau đó đặt màu: *g.setColor(WHITE);* hoặc  
*g.setColor(red, green, blue);*

# Nét vẽ

- ▶ Chọn nét khi vẽ đường thẳng, cung và hình chữ nhật.

`int getStrokeStyle();` // trả về kiểu nét vẽ

`void setStrokeStyle(int style);` // đặt kiểu nét vẽ

- ▶ Hai kiểu nét vẽ được định nghĩa trong lớp Graphics là:

- ▶ Nét chấm : `g.setStrokeStyle(Graphics.DOTTED);`

- ▶ Nét liền: `g.setStrokeStyle(Graphics.SOLID);`



# Vẽ cung

► Vẽ cung:

```
void drawArc(int x, int y, int width, int height, int  
    startAngle, int arcAngle);
```

```
void fillArc(int x, int y, int width, int height, int startAngle,  
    int arcAngle);
```

```
Ví Dụ: g.drawArc(10, 10, 100, 100, 0, 150);
```

# Vẽ hình chữ nhật

- ▶ Vẽ hình chữ nhật

```
void drawRect(int x, int y, int width, int height); //
```

```
void drawRoundRect(int x, int y, int width, int height, int arcWidth, int  
arcHeight);
```

```
void fillRect(int x, int y, int width, int height);
```

```
void fillRoundRect(int x, int y, int width, int height, int arcWidth, int  
arcHeight);
```

# Vẽ Font chữ

- ▶ Các phương thức dựng của lớp Font:
  - ▶ Font `getFont(int face, int style, int size);`
  - ▶ Font `getFont(int fontSpecifier);`
  - ▶ Font `getDefaultFont();`
- ▶ Một số thuộc tính của lớp Font
  - ▶ FACE\_SYSTEM  
FACE\_MONOSPACE  
FACE\_PROPORTIONAL
  - ▶ STYLE\_PLAIN  
STYLE\_BOLD  
STYLE\_ITALIC  
STYLE\_UNDERLINED
  - ▶ SIZE\_SMALL  
SIZE\_MEDIUM  
SIZE\_LARGE

# Vẽ Font chữ

- ▶ Các tham số kiểu dáng có thể được kết hợp thông qua toán tử hay (|)
- ▶ Ví dụ

```
Font font = Font.getFont(Font.FACE_PROPORTIONAL, Font.STYLE_BOLD  
    || Font.STYLE_ITALIC, Font.SIZE_MEDIUM);
```

# Vẽ chuỗi ký tự

```
void drawChar(char character, int x, int y, int anchor);
```

```
void drawChars(char[] data, int offset, int length, int x,  
    int y, int anchor);
```

```
void drawString(String str, int x, int y, int anchor);
```

```
protected void paint(Graphics g) {  
    int xcenter = getWidth() / 2,  
        ycenter = getHeight() / 2;  
    g.setFont(Font.getFont(Font.FACE_SYSTEM,  
        Font.STYLE_ITALICS, Font.SIZE_MEDIUM));  
    g.drawString("developerWorks", xcenter, ycenter,  
        Graphics.BASELINE | Graphics.HCENTER);  
}
```

# Vẽ đối tượng hình ảnh

- ▶ Lớp Graphics hỗ trợ một phương thức duy nhất để vẽ lên màn hình đó là :

`void DrawImage(Image img,int x, int y, int anchor)`

# Vẽ đối tượng hình ảnh

- ▶ Đối tượng image gồm 2 loại:
  - ▶ Hình ảnh có thể thay đổi (Mutable Image)
  - ▶ Hình ảnh cố định (Immutable Image)

# Vẽ đối tượng hình ảnh

## Immutable Image

- ▶ Tạo đối tượng hình ảnh( thường là load từ file)

```
Image img =Image.CreateImage("\\test.png");
```

- ▶ Hiển thị đối tượng lên màn hình.

```
protected void paint (Graphics g)  
{ ....  
g.drawImage(img,10,10,  
             Graphics.LEFT|Graphics.TOP);  
}
```



# Vẽ đối tượng hình ảnh

## Mutable Image

- ▶ Tạo đối tượng hình ảnh.

```
Image img =Image.CreateImage(80,100);
```

- ▶ Hiển thị đối tượng lên màn hình

```
g=img.getGraphics();
```

```
g.fillRoundRect(0,0,50,50,20,20);
```

```
protected void paint (Graphics g)
```

```
{ ....
```

```
g.drawImage(img,10,10,Graphics.VCENTER|Graphics.HCENTER); }
```

# Tài liệu tham khảo

- ▶ Kim Topley, *J2ME in a Nutshell*, O'Reilly, 2002
- ▶ Roger Riggs, *Programming Wireless Devices with the Java™ 2 Platform Micro Edition Second Edition*, Addison Wesley, 2003
- ▶ <http://java.sun.com>
- ▶ <http://www.javavietnam.org>

# Q/A