Draw the arrow

Example Introduction

In this section, we introduce the draw_arrow() method for drawing lines

API Documentation

```
image.draw_arrow(x0, y0, x1, y1[, color[, thickness=1]])
```

Draw an arrow from `(x0, y0)` to `(x1, y1)` on the image. Parameters can be passed in `x0, y0, x1, y1` separately or as a tuple `(x0, y0, x1, y1)`.

- **color**: An RGB888 tuple representing a color, suitable for grayscale or RGB565 images, defaulting to white. For grayscale images, you can also pass pixel values (range 0-255); for RGB565 images, you can pass byte-flipped RGB565 values.
- **thickness**: controls the pixel width of the arrow line, default is 1.

This method returns an image object, allowing other methods to be called through chaining.

Compressed images and Bayer format images are not supported.

Sample code

```
# Import required modules
# 导入所需的模块
import time, os, urandom, sys, math

# Import display and media related modules
# 导入显示和媒体相关模块
from media.display import *
from media.media import *

# Define display resolution constants
# 定义显示分辨率常量
DISPLAY_WIDTH = 640
DISPLAY_HEIGHT = 480

def display_test():
    """
    Function to test display functionality
    测试显示功能的函数
    """

    # Create main background image with white color
```

```
    # 创建白色背景的主图像
    img = image.Image(DISPLAY_WIDTH, DISPLAY_HEIGHT, image.ARGB8888)
    img.clear()
    img.draw_rectangle(0, 0, DISPLAY_WIDTH, DISPLAY_HEIGHT,color=
(255,255,255),fill=True)

    # Initialize display with ST7701 driver
    # 使用ST7701驱动初始化显示器
    Display.init(Display.ST7701, width = DISPLAY_WIDTH, height = DISPLAY_HEIGHT,
to_ide = True)
    # Initialize media manager
    # 初始化媒体管理器
    MediaManager.init()

    try:
        # The central main arrow symbolizes the direction of travel
        img . draw_arrow ( 320 , 200 , 400 , 200 , color =( 0 , 191 , 255 ),
thickness = 5 )   # Standard sky blue

        # Auxiliary arrows to increase the sense of hierarchy
        img . draw_arrow ( 300 , 180 , 380 , 180 , color =( 135 , 206 , 235 ),
thickness = 3 )   # Light sky blue
        img . draw_arrow ( 340 , 220 , 420 , 220 , color =( 135 , 206 , 235 ),
thickness = 3 )   # Light sky blue

        # Diagonal arrows to add dynamism
        img . draw_arrow ( 250 , 150 , 350 , 250 , color =( 0 , 191 , 255 ),
thickness = 3 )   # Standard sky blue
        img . draw_arrow ( 350 , 150 , 450 , 250 , color =( 0 , 191 , 255 ),
thickness = 3 )   # Standard sky blue

        # Reverse arrows to increase contrast
        img . draw_arrow ( 400 , 200 , 320 , 200 , color =( 173 , 216 , 230 ),
thickness = 3 )   # Lightest sky blue
        img . draw_arrow ( 380 , 180 , 300 , 180 , color =( 173 , 216 , 230 ),
thickness = 2 )   # Lightest sky blue
        img . draw_arrow ( 420 , 220 , 340 , 220 , color =( 173 , 216 , 230 ),
thickness = 2 )   # Lightest sky blue

        # Vertical arrows to increase the three-dimensional effect
        img . draw_arrow ( 320 , 150 , 320 , 250 , color =( 0 , 191 , 255 ),
thickness = 3 )   # Standard sky blue
        img . draw_arrow ( 400 , 150 , 400 , 250 , color =( 0 , 191 , 255 ),
thickness = 3 )   # Standard sky blue

        # Dotted arrows to add details
        img . draw_arrow ( 300 , 220 , 310 , 230 , color =( 135 , 206 , 235 ),
thickness = 2 )   # Light sky blue
        img . draw_arrow ( 330 , 170 , 340 , 180 , color =( 135 , 206 , 235 ),
thickness = 2 )   # Light sky blue
        # Update display with background image
        # 更新显示背景图像
        Display.show_image(img)
        while True:
            time.sleep(2)

    except KeyboardInterrupt as e:
        print("user stop: ", e)
```

```
    except BaseException as e:
        print(f"Exception {e}")

    # Cleanup and deinitialize display
    # 清理并反初始化显示器
    Display.deinit()
    os.exitpoint(os.EXITPOINT_ENABLE_SLEEP)
    time.sleep_ms(100)
    # Release media resources
    # 释放媒体资源
    MediaManager.deinit()

if __name__ == "__main__":
    # Enable exit points and run display test
    # 启用退出点并运行显示测试
    os.exitpoint(os.EXITPOINT_ENABLE)
    display_test()
```

Routine running effect