Draw a circle

Example Introduction

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

API Documentation

`draw_circle`

```
image.draw_circle(x, y, radius[, color[, thickness=1[, fill=False]]])
```

Draw a circle on the image. Parameters can be passed in `x, y, radius` separately or as a tuple `(x, y, radius)`.

- **color**: RGB888 tuple representing the color, suitable for grayscale or RGB565 images, default is 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 circle border, default is 1.
- **fill**: When set to `True`, the inside of the circle will be filled, the default is `False`.

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

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:
        # Main outline
        # 主轮廓
        img.draw_circle(320, 240, 150, color=(50, 50, 50), thickness=8)  # 外圈 Outer circle
        img.draw_circle(320, 240, 130, color=(80, 80, 80), thickness=5)  # 内圈 Inner circle

        # Center Hub
        # 中心轮毂
        img.draw_circle(320, 240, 40, color=(100, 100, 100), fill=True)  # 填充 Fill
        img.draw_circle(320, 240, 40, color=(50, 50, 50), thickness=3)   # 轮毂边框 wheel border
        img.draw_circle(320, 240, 15, color=(30, 30, 30), fill=True)     # 轮毂中心 Center of the wheel

        # Spokes
        # 辐条
        for i in range(8):
            angle = i * (360 / 8)
            x_outer = int(320 + 130 * math.cos(math.radians(angle)))
            y_outer = int(240 + 130 * math.sin(math.radians(angle)))
            x_inner = int(320 + 40 * math.cos(math.radians(angle)))
            y_inner = int(240 + 40 * math.sin(math.radians(angle)))

            # Main spokes
            # 主辐条
            img.draw_circle(x_outer, y_outer, 10, color=(70, 70, 70), fill=True)
            img.draw_circle(x_inner, y_inner, 8, color=(70, 70, 70), fill=True)

        # Decorative Bolts
        # 装饰性螺栓
        for i in range(16):
            angle = i * (360 / 16)
            x = int(320 + 140 * math.cos(math.radians(angle)))
            y = int(240 + 140 * math.sin(math.radians(angle)))

            img.draw_circle(x, y, 5, color=(40, 40, 40), fill=True)
        # 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()
```

Example running effect

You can see that we draw a circle in the center of the screen that looks like a "mechanical bearing" or "wheel".