

人工智能

第二讲

图形、字符串、列表与文件

杜小勤

武汉纺织大学数学与计算机学院

2018/09/09

图形

底层的图形库：Tkinter；

封装文件：graphics.py

主要内容

- 绘制简单的图元；
- 坐标系统；
- 交互式图形；
 - 鼠标；
 - 键盘；
- 图形模块；
 - GraphWin 类；
 - 基本图元类；
 - Entry 类；
 - Image 类；
 - 颜色；
 - 显示的更新；

图形窗口与基本图元

- 阅读并运行程序 `simple1.py` 和 `simple2.py`;
- 掌握各类及主要方法的使用;

图形窗口与基本图元

类:

- GraphWin;
- Point;
- Circle;
- Text;
- Rectangle;
- Line;
- Oval;
- Polygon;

图形窗口与基本图元

方法：

- `close;`
- `draw;`
- `setOutline;`
- `setFill;`
- `setWidth;`
- ...

注意事项

- Python 中的别名问题;
- 要使用 clone 方法来解决图元复制问题;

年利率程序的图形化

- 年利率程序的图形化：futval_graph.py；
- 该程序存在的问题——图形系统没有抽象化，导致绘制过程繁琐；

图形系统的抽象与优化——坐标系

■ 优化与抽象前

程序员需要将实际问题的数据坐标系转换到图形系统的坐标系，例如
`futval_graph.py`;

■ 优化与抽象后

程序员直接使用实际问题中的数据坐标系进行绘制——但是，需要添加一个坐标系，它的任务是将实际数据坐标转换成图形系统的坐标。优化后的程序 `futval_graph2.py`;

如何优化与抽象？

- 使用线性变换在两者之间建立直接的缩放比例关系；
- graphics 图形库提供了坐标系统的转换函数：setCoords；

TicTacToe 棋盘的绘制

- `setCoords(0.0, 0.0, 3.0, 3.0)`
左下角坐标 (0.0,0.0), 右上角坐标 (3.0,3.0);
- 依据实际问题的数据坐标进行绘制
`Line(Point(1,0),Point(1,3)).draw(win)`
`Line(Point(2,0),Point(2,3)).draw(win)`

坐标系统的引入, 屏蔽了具体的图形窗口与坐标, 使得程序员可以专注于问题本身;

年利率的图形化改进版本

- 阅读并运行程序 `futval_graph2.py`;
- 认真体会改进后带来的好处;

交互式图形的设计

■ 鼠标;

■ `getMouse`

等待用户点击鼠标左键。运行程序：`click.py`、`triangle.py`;

■ `checkMouse`

功能类似，但是不会等待用户按下鼠标左键 (如果没有按下鼠标左键，返回 `None`);

■ 键盘;

■ `getKey`

等待用户按键，返回按键代表的字符串。运行程序：`clickntype.py`;

■ `checkKey`

功能类似，但是不会等待用户按键 (如果没有按键，返回空字符串 `""`);

交互式图形的设计

- 简单的文本输入框：Entry 类
阅读并运行程序 `convert_gui.py`;

graphics 模块

- GraphWin 类;
- 基本图元类;
- Entry 类;
- Image 类;
- 颜色;
- 显示的更新;

详见 5.6 节;

字符串、列表与文件

- 字符串；
 - 索引与切片；
 - 简单连接与重复连接；
 - 迭代；
- 列表；
 - 基本用法；
 - 列表解析式；
 - 列表生成器；
- 字符编码；
- 格式化输出；
- 其它方法
- 文件；

详见第 6 章；

参考文献

1. 杜小勤。《人工智能》课程系列，Part I: Python 程序设计基础，2018/06/13。
2. 杜小勤。《人工智能》课程系列，Part II: Python 算法基础，2018/07/31。