

# Readme for *Fake Logo Interpreter* *"GKD Mr.T"*

## 伪 Logo 语言解释器“龟龟快点爬呀”·说明文档

---

© ZENG Ziyu / WU Chenhao / NIE Dingyi, 2019

---

## 目录

- 0 分工情况
  - 1 工程文件说明
  - 2 操作说明
    - 2.1 交互界面
    - 2.2 输入和输出
  - 3 对伪 Logo 语法的支持
    - 3.1 文件头
    - 3.2 基础语法
    - 3.3 控制语法
    - 3.4 扩展语法
  - 4 ErrorCodes
- 

## 0 分工情况

- 聂鼎宜 (计96, 2019011346) 负责开荒, 完成了除 FUNC / LOOP 语法相关功能之外的基础功能, 并撰写了这份文档;
  - 曾子渝 (计96, 2019011343) 接手完成了 FUNC / LOOP 语法的解释, 并加入了画笔宽度和填色两项拓展功能;
  - 吴宸昊 (计96, 2019011338) 实现了灰度化和分形图形绘制两项拓展功能, 此外负责演示视频的制作。
- 

## 1 工程说明

- GKD Mr.T 能够根据您提供的伪 Logo 语言程序生成 .bmp 位图;

- 该工程的源代码仅由单个 C++ 源文件 (*main.cpp*) 组成，位于 **src** 目录下；
  - 代码由三位组员各自在 Xcode、VSCode 和 Dev C++ 环境下写成的相应部分汇总而来；
  - 在 Windows 上编译得到的 .exe 或在 Mac OS 上编译得到的 .exec 经测试均可完美运行，可以正常使用全部功能；
  - 为确保可读性，用注释标示的分割线将代码分成多个部分，分别是：
    - *//info*: 工程名、版权信息
    - *//headers*: 头文件、*namespace* 和对齐方式声明
    - *//definitions*: 给出所有工程涉及到的数据类型、数组、变量、结构体的定义
    - *//declarations*: 部分函数的预先声明
    - *//basic functions*: 基本函数，包括但不限于位图坐标转 *index*、*Deg* 转 *Rad*、画笔约束、变量/参量抓取的相关函数
    - *//functions*: 实现对应 Logo 语法解释的函数体
    - *//operations*: 解释过程的操作集，包括但不限于读取 .logo 文件、位图初始化、Logo 语句识别、保存位图文件的相关函数
    - *//main*: *main* 函数
    - *//errorcodes*: 错误代码表
  - 绘制 8 字形、渐变方框、奥运五环、五星红旗的输入 .logo 文件置于 **input** 目录下；此外，该目录中还包含有体现拓展功能的其他输入文件。
- 

## 2 操作说明

### 2.1 交互界面

- 运行 *main.cpp* 编译得到的可执行文件，我们可爱的龟龟 [*Triangle*] 便会出现向您问好 (此命名致敬了经典 LOGO 软体的主角——三角形龟龟)；
- [*Triangle*] 能向您报告 .logo 语言解释的进度。在打开文件、读取文件头、初始化、解释文件主体、保存位图五个步骤结束时，打印完成的状态。程序最终返回 0 代表全部进程顺利执行，并成功得到输出；
- 如果解释过程中出现错误，绝大多数情况是 .logo 文件的语法错误导致。此时，[*Triangle*] 将为您返回一个三位数的错误代码，您将可以据此对伪 Logo 语言程序进行调试。

### 2.2 输入和输出

- 输入文件：以 .logo 为扩展名的文本文档 (**不超过 1000 行**)，内含以本文档中定义的伪 Logo 语言写成的绘图程序；
  - 输入方式：在 [*Input File*] 提示处键入输入文件所在的路径 (以 *FILENAME.logo* 结尾)，回车，或者直接键入文件名回车 (当输入文件与可执行文件位于相同路径下时)；
  - 输出文件：24 位位图，.bmp 格式；
  - 输出路径：用户指定，键入于 [*Output File*] 提示处，以 *FILENAME.bmp* 结尾。
-

## 3 对伪 Logo 语法的支持

### 3.1 文件头

以下语句需要缺一不可地（但可以按任意顺序）包含在 .logo 输入文件的头部：

- **@SIZE [W][H]**
  - [W] 应替换为一正整数，表示图像宽度（单位：Px）；
  - [H] 应替换为一正整数，表示图像高度（单位：Px）。
- **@BACKGROUND [R][G][B]**
  - [R]、[G]、[B] 应替换为 [0, 255] 的整数，分别表示背景颜色的红、绿、蓝分量。
- **@POSITION [X][Y]**
  - [X] 应替换为 [0, W - 1] 的整数，[Y] 应替换为 [0, H - 1] 的整数，表示画笔初始坐标。在输出的位图中，极左下的像素点坐标为 (0, 0)，其右的像素有更大的 X 分量，其上的像素有更大的 Y 分量。

### 3.2 基础语法

以下语句是本工程支持的基础语句集，附有用法解释及效果说明：

- **DEF [NAME][VALUE]**
  - 用于定义变量的语句；
  - [NAME] 应替换为标示变量名的字符串。在原理上，我们允许使用任意字符进行命名，但为了避免降低伪 Logo 程序的可读性，建议您使用大小写字母进行命名；
  - [VALUE] 应替换为变量的值。可以为任何实数（但超过 float 型上下限的将被截断），也可以为已经定义的变量名，或者函数内部的形式参量；
  - 注意：如果您用已经存在的变量或参量的值来定义新的变量，则该值是以值传递的，而非以引用传递的；
  - 注意：函数内定义新变量，在其他环境下仍可以引用。即变量不分全局 / 局部。
  - 注意：允许定义的变量数量上限是 10 个。
- **ADD [NAME][VALUE]**
  - 用于修改变量值的语句；
  - [NAME] 应替换为已经定义的变量名；
  - [VALUE] 应替换为一实数，或者已经定义的变量名，或者函数内部的形式参量；
  - 该语句经解释器解释后，变量 [NAME] 将被赋予新值：[NAME] + [VALUE]；
  - 注意：您不可以对传入函数的参数使用该语句。
- **MOVE [STEP]**
  - 控制龟龟 *Triangle* 爬行的语句；
  - [STEP] 应替换为任意实数，或者已经定义的变量名，或者函数内部的形式参量，表示控制 *Triangle* 向当前方向前进 [STEP] 步长。步长可以是负数，这将控制 *Triangle* 向反方向前进，但并不改变其朝向；

- **TURN [ANGLE]**

- 控制 *Triangle* 转向的语句；
- 默认地，*Triangle* 经初始化时朝向 Y 的正方向（向上）；
- *[ANGLE]* 应替换为任意实数，或者已经定义的变量名，或者函数内部的形式参量，表示控制 *Triangle* 顺时针旋转 *[ANGLE]* 角度。角度可以是负数，这将控制 *Triangle* 向逆时针方向旋转；
- 注意：角度为**角度制**（单位：Deg）。

- **COLOR [R][G][B]**

- 控制换墨的语句；
- *[R]*、*[G]*、*[B]* 应替换为 [0, 255] 的整数，分别表示新墨颜色的红、绿、蓝分量；
- 我们允许您用变量表示颜色的三个分量，这样，您可以更轻松的用含循环结构的函数绘制渐变的线条；
- 注意：该语句经解释器解释时，自动将 *Triangle* 置于显形状态。

- **CLOAK**

- 该语句经解释器解释，*Triangle* 将在隐身 / 显形两个状态间切换；
- 隐身状态下，MOVE 语句将不再留下墨迹，但 *Triangle* 确实在移动。

### 3.3 控制语法

以下语句是本工程支持的特定结构集，附有功能说明：

- **插入一个循环**

```
1 | LOOP [VALUE]
2 | //insert your code here...
3 | END LOOP
```

- 用以插入循环体；
- *[VALUE]* 应替换为一**正整数**，或者已经定义的变量名，或者函数内部的形式参量，表示循环次数；
- 解释器将自动配对 *LOOP* 与 *END LOOP* 语句，将其间的程序循环解释 *[VALUE]* 次。

- **定义一个函数**

```
1 | FUNC [NAME]([F PARA1], [F PARA2], [F PARA3])
2 | //insert your code here...
3 | END FUNC
```

- 用以定义函数体；
- *[NAME]* 应替换为标示函数名的字符串。在原理上，我们允许使用更多的字符进行命名，但为了避免降低伪 Logo 程序的可读性，建议您使用**大小写字母**进行命名；

- `[F PARA1]`、`[F PARA2]` 和 `[F PARA3]` 应分别替换为标示形式参量 1、形式参量 2 和形式参量 3 的字符串。在原理上，我们允许使用更多的字符进行命名，但为了避免降低伪 Logo 程序的可读性，建议您使用**大小写字母**进行命名；
- 注意：您不必用完全部的三个形式参量，但**一个函数最多只能拥有 3 个形式参量**；
- 注意：**允许定义的函数数量上限是 10 个**。

### • 调用一个函数

```
1 | CALL [NAME]([PARA1], [PARA2], [PARA3])
```

- 用以调用函数；
- `[NAME]` 应替换为已经定义的函数名；
- `[PARA1]`、`[PARA2]` 和 `[PARA3]` 应分别替换为一数值，或者已经定义的变量名（如在函数内调用其他函数，还可以为当前函数环境下的形式参量），为实际参量；
- 注意：您不必用完全部的三个参量，但**实际参量与形式参量一一对应**；
- 注意：**您不可在函数内调用其自身**；

## 3.4 扩展语法

以下语句是本工程支持的扩展语句集，附有用法解释及效果说明：

### • **PENWIDTH [D]**

- *Triangle* 获得全新装备，笔触直径发生变化。若不使用该语句，笔触直径默认为 0；
- `[D]` 应替换为一正整数，或者已经定义的变量名，或者函数内部的形式参量，表示新的笔触直径。

### • **FILL [R][G][B]**

- *Triangle* 精神爆发，用指定颜色填充包含 *Triangle* 所在位置的整个同色区域；
- `[R]`、`[G]`、`[B]` 应替换为 `[0, 255]` 的整数，分别表示填充颜色的红、绿、蓝分量。

### • **GREY**

- *Triangle* 在爬行过程中不幸猝死，为了祭奠它，整张位图灰度化。

### • **RECRS [STEP][ANGLE][TIME]**

- *Triangle* 学会影分身之术，可以绘制分形图形；
- `[STEP]` 应替换为任意实数，或者已经定义的变量名，或者函数内部的形式参量；`[ANGLE]` 应替换为任意实数，或者已经定义的变量名，或者函数内部的形式参量；`[TIME]` 任意正整数，或者已经定义的变量名，或者函数内部的形式参量；
- 语句经解释器解释时，*Triangle* 每前进 `[STEP]` 步（单位：Px），便分裂成与原方向夹 `[ANGLE]` 角度（单位：Deg）的两支，并将该过程重复 `[TIME]` 次。

## 4 ErrorCodes

- 100 - 输入文件打开失败；
- 101 - 输出文件写入失败；
- 102 - 文件头缺失；
- 103 - 引用 / 修改未定义的变量；
- 104 - 变量定义过多；
- 105 - 函数体结构异常；
- 106 - 循环体结构异常；
- 107 - 重复定义函数；
- 108 - 调用未定义的函数；
- 109 - 形式参量过多；
- 110 - 函数定义过多；
- 111 - 伪 Logo 程序长度超限；
- 112 - 函数名过长；
- 113 - 形式参量名过长；