科研开源软件创意大赛

算法精灵软件项目规格书

参赛团队/个人名称

2019年6月26日

目录

[一．引言 3](#_Toc12213865)

[1.1背景 3](#_Toc12213866)

[1.2定义 4](#_Toc12213867)

[1.3参考资料 4](#_Toc12213868)

[二．任务概述 4](#_Toc12213869)

[2.1目标 4](#_Toc12213870)

[2.2创作思路 4](#_Toc12213871)

[2.4成功案例 7](#_Toc12213872)

[2.5需求规定 7](#_Toc12213873)

[2.6对性能规定 8](#_Toc12213874)

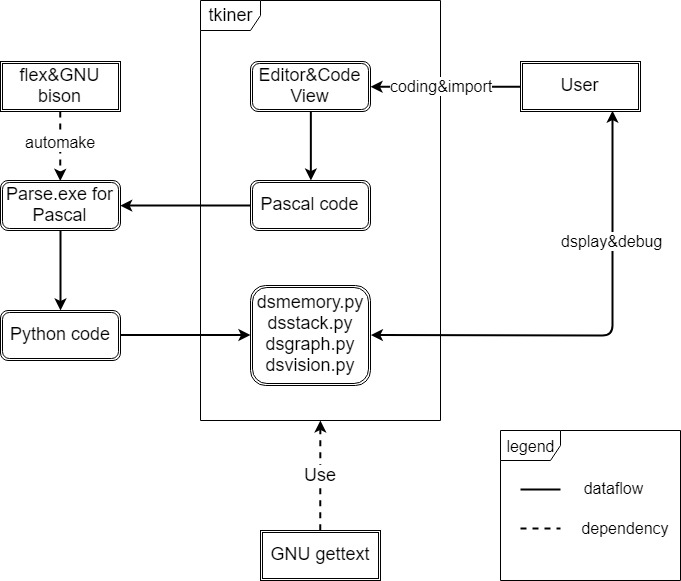
[2.7对其他的专门要求 8](#_Toc12213875)

# 一．引言

## 1.1背景

算法精灵是一个数据结构辅助教学的工具，同时又是一个帮助学习和理解算法的工具。面向广大学生和教师。本项目的设计者和开发者是德新软件科技。

算法精灵基于Python2.7构建，支持用户使用Pascal语言编写算法进行执行和演示，我们基于flex和GNU bison设计了Pascal-python的解析器将Pascal程序转换为Python源代码并执行演示，如下图。



算法精灵可以帮助你制作算法课件，也可以帮助你设计和调试算法。

算法精灵能够把算法的执行过程能够使用动画的方式把内存数据、堆栈数据等的变化直观的显示出来，可以帮助你理解算法执行过程和算法的实现原理。而这些动画的过程可以被提取出来作为设计课件的素材。

算法精灵还允许你输入自己设计的算法，然后运行该算法，通过图形化的方式来展示算法执行过程，从而发现算法设计中的问题和更好的改进算法，还可以在算法中增加代码断点和数据断点。

算法精灵内置了大部分的经典算法。每一个算法都可以作为一个游戏来启动。在游戏过程中，你需要正确的按照算法目前的状态正确的设定图片的属性，或者填写正确的数据值，才可以继续游戏或者得到积分（游戏功能在当前版本尚未实现）。

算法精灵使用pyinstaller进行打包，可运行在window和Linux等平台上。

## 1.2定义

Flex: 基于C和C++的快速词法分析器的生成器。

GNU bison: 属于GNU项目的一个语法分析器生成器。

Pascal: 由瑞士Niklaus Wirth教授于六十年代末设计并创立的一门结构化编程语言，常常被用作学习数据结构与算法的教学语言。

GNU gettext: 一种多国语言解决方案

## 1.3参考资料

Flex: <https://github.com/westes/flex>

GNU bison: <https://www.gnu.org/software/bison/manual/>

Python: <https://docs.python.org/2/>

Tcl/Tk: <https://core.tcl-lang.org/index.html>

# 二．任务概述

## 2.1目标

根据人类的学习知识、掌握知识和利用知识的过程，利用计算机技术设计教材和教案，从而更好的达到教书育人的目的。

## 2.2创作思路

知识就是外部世界规律的总和，获取知识的途径是创新和学习，而学习有两种模式，即言传身受和通过文字。下面让我们考察一下一个人是如何通过文字获得知识的：

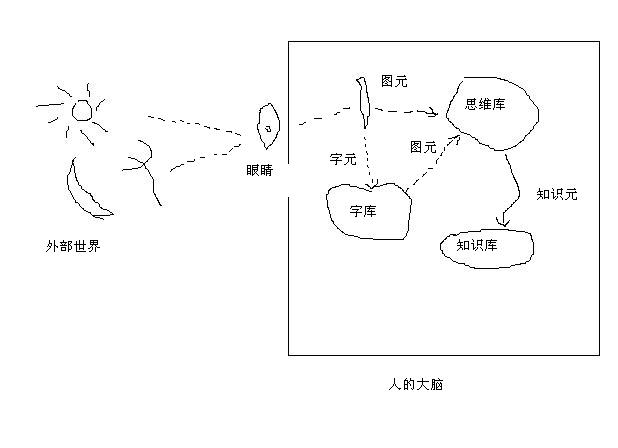
第一要在大脑记忆若干文字，也就是说，看到这个文字能过转换为对应世界的某事或某物。

第二是读某段文字，譬如“太阳有九大行星”。那么，人的大脑就会产生“太阳系”和“行星”的概念，然后又产生在太阳系里面有九大行星的图案，最后把这幅图像也存储在大脑中。这样，便是学习了一点新知识。

上面所描述的人的学习过程是一个简单的说明，我们要建立的模型是基于任何一个功能必然有对应的器官来实现这样一个物质至上的原则来实现。任何一个功能通过一个时序可以清楚地表达出来，这样的模型才可以将整个学习的过程详细的再现出来。当然，我们的模型不是生理模型，而是基于对功能的抽象模型，首先能够合理的描述了各种功能，我们便可以认为其是合理的，然后试图应用这个模型做一些推广和应用，如果在实现中发现了某种与现实存在的不一致，那么可以重新修改这个模型。

在这个意义上，知识就是文字所代表的物质世界映像，这个映像可能是一副图片，也可以是多幅图片，甚至是无穷多幅图片，还可以是一种时序，即多帧连续的图片，还可能包含多个时序。总而言之，知识就是文字所蕴含的现实世界的映像。能把文字转换成为某种现实世界的映像，那么，你就理解了这段文字。事实上，知识的发现者正式总结了和审视了所有的可能性，将有穷或者无穷多个物质世界映像凝结成一段文字，后人则正好相反，从文字中溯源或者说解读其所包含的图像。

这里我们提出一种人类的思维模型，如下图：



图中的各个概念如下：

眼睛： 用于观察客观世界。眼睛从客观世界得到光，输出图元或者字元。

图元、字元：就是客观世界经过眼睛之后在大脑中产生的一种物质。譬如桌子、太阳、月亮，眼睛看到这些之后在大脑中产生的就是图元，如果看到是文字，那么大脑中产生的是字元。

字库、思维库、知识库：分别对应大脑的不同区域。

字库：储存有字元和图元对应关系的仓库，一个字元可能对应多个图元。而针对不同的人，相同的字元夜可能对应不同的图元。

思维库：用于组合各种图元，形成新的知识的器官。

知识库：就是存储知识元的器官。

知识元：就是每一个知识所代表的所有图元的集合。

大脑：用于控制各种物质元流向的器官。

下面我们使用这个模型来说明人的学习过程：

A. 字库的建立。

人首先通过眼睛看字，然后在思维中把这个字元和图元对应起来，最后存放在字库中。

字元和那些图元建立关系，完全是后天的教育形成，所以不同的人面对相同的字元完全可能对应不同的图元。

第一种方式是字元和图元同时输入，譬如看一个“狗”字，然后在看看现实世界的狗，从而在两者之间建立一种关联，这是直接学习法。

第二种方式是通过文字或者语言（讲解）来建立关系。譬如在老师告诉你，或者你看到文字“狗就是四条腿，浑身长毛的动物”，从而在大脑中通过思维库建立“狗”的字元和图元之间的关系。

每一个人看书愈多，对同一个字形成的映像就愈多。从某一种意义上说，也就是知识也越多。譬如看到“伐”字，没有古文知识的人想到的只有“进攻，征伐”的意思，而熟悉古文的人自然会想到还有另外一个常用的意义“夸耀自己”。

B. 学习知识

眼睛看到文字，形成字元和图元。大脑把字元发送到字库，得到该字元对应的图元，然后把图元输入到思维库。所有的字的每一种意义（也就是图元）都会输入到思维库进行排列组合，最后得到一种有意义的输出，然后把这个图元组合形成知识元，放到知识库。这就是人类学习知识的基本过程。

这里我们可以看到，在这个过程中，字库和思维库起到了相当大的作用，字库越丰富，对同一个字存储的图元越多，那么组合的可能性就越多，就愈能够理解知识。而思维库组合的速度愈快，组合的方式愈多，则理解力越强。这也是区别人的智力的两大重要因素。实际上，大多数人的字库存储都局限于一定水平，并且思维库也只能达到一般水平。为了让大部分的人更好更快的学习和掌握知识，有一种可行的方式就是用外部的素材来代替内部的字库和思维库，从而使得不同的大脑能够很快的掌握知识。换言之，我们提供知识的时候不仅仅只是提供文字，而是同时提供文字所对应的各种图像和场景，即把字库和思维库的工作直接在外面的素材中展现出来，大脑需要做的工作相当于把这种知识直接放到知识库中，不需要思维库和字库的参与。这就是我们下面提到的教材设计方案的理论基础。实际上，借助于现代计算机技术，使得这一切正变得可能。

知识就是力量，在当代社会，如何快速传播和学习知识，则更是一个国家的立国之本。如何让全民素质迅速提高，如何把知识普及到最广泛的群体中去，而不是仅仅服务于知识精英，根据上文我们提出的理论，我们找到解决方案。

所有的文字化的知识我们都是用对应的图像以及时序（场景）的方式描述出来，通过不断的重复观察，从而使大多数人能够快速容易的掌握知识。

我们的第一步是定义各个文字所对应的图像或者时序，然后针对各门知识，定义每一个知识点对应的图像和时序。对于某一个知识点，可能需要无限多的图像，我们则抽取有代表性的图片即可。总之，我们的核心就是：利用现代的计算机技术，借助于各种媒体，对现有的文字知识进行全部的图像化处理，从而将知识所代表的内涵用实际的图像和场景展示出来。

这种方式不仅仅适用于现代各种科学技术方面的知识，对于中国古代传统的文化也同样适用。我始终相信中国传统文化的潜在价值与应用前景，但是，以其年代的久远以及文字的晦涩，让现代青少年望而却步。如果能够用这种方式重新展示在世人面前，我们一定会从其中汲取很多营养，尤其是关于价值观、道德观和社会文化等各个方面。

**2.3科研领域**

算法精灵是一个数据结构辅助教学的软件，适用于智能教育行业

## 2.4成功案例

*（如已有应用案例，提供软件使用情况和用户意见反馈等信息）*

## 2.5需求规定

该软件需要用户自己用Pascal语言来编写自己的算法，可以手动编写、读入源文件或者示例的经典算法。基于flex做词法分析，进一步利用bison做语法分析，将Pascal翻译成python源代码，该软件直接执行python源文件将算法的实现过程以动画的形式一步一步准确的展示给用户，包括代码执行位置、局部变量值、堆栈的变化情况以及程序或算法中的错误。

算法演示：

打开一个算法文件(.paf)或者通过editor编辑

查看 publish 目录下面是否存在编译好的文件，即对应的(.py)

文件，如果不存在提示需要编译；

装载代码到代码视图；

初始化 dscore.Driver，aftype.DATAPOOL, aftype.VOPTIONS；

初始化 参数列表，调用堆栈，控制台和数据视图；

装载算法模块；

这时候进入交互模块，等待用户进行操作，譬如可以修改算法参

数等，点击启动则开始执行算法对应的脚本文件。

## 2.6对性能规定

该软件需要用户自己用Pascal语言来编写自己的算法，可以手动编写、读入源文件或者示例的经典算法。注意文本使用的编码，一般为UTF-8。软件的可视化过程要求具有极高的执行效率，变量值和堆栈视图显示绝对准确，对用户的各种错误的输入都能够快速正确的响应，并且能使程序优雅的结束。

## 2.7对其他的专门要求

运行环境

Windows

Python2.7

Debian, Ubuntu

Python2.7 python-tk Tix

Red Hat

Python2.7