****

**本科生毕业论文（设计）**

(\*论文题目应以简短、明确的词语恰当概括整个论文的核心内容，避免使用不常见的缩略词、缩写字。读者通过标题可大致了解毕业设计（论文）的内容、专业的特点和科学的范畴。中文题目一般不宜超过24个字，必要时可增加副标题。外文题目一般不宜超过12个实词。\*)

题 目："Dyscrabble"动态拼字游戏的设计与实现

院 系： 软件学院

专 业： 软件工程（嵌入式软件与系统）

学生姓名： 赖卓航

学 号： 11331157

指导教师： 刘宁 （副教授）

二〇一 五 年 三 月

**摘 要**

拼字游戏是一款起源于西方国家的文字类益智游戏，深受许多玩家喜爱。现有的大多数拼字游戏都是从指定词典取词后生成拼字地图并给予玩家少许提示以完成拼图。本次设计主要针对传统的拼字游戏进行形式上的创新，增加了从网络中下载实时新闻、学术类文章，让玩家阅读后从文章中选词填入等功能。选取的均为词频较低或较为生僻的词汇。

论文首先介绍拼字游戏的发展历程。再介绍游戏界面的设计与制作过程。然后介绍游戏各个模块的设计过程，包括通过网络爬虫技术设计模块从指定网站上获取英文文章，通过词频数据库分析文章词汇并得出词频最低的若干词汇，以及最后基于选取的词汇通过拼字地图生成算法生成地图等等。接着介绍游戏中使用到了算法以及优化策略。最后将展示这款动态拼字游戏，从画面中感受其带来的游戏效果。

**(\* 中文摘要不少于300字。**

摘要、关键词和设计说明

**1.中文摘要和中文关键词**

摘要内容应概括地反映出本论文的主要内容，主要说明本论文的研究目的、内容、方法、成果和结论。要突出本论文的创造性成果或新见解，不要与引言相混淆。语言力求精练、准确，以300—500字为宜。

在摘要的下方另起一行，注明本文的关键词（3—5个）。关键词是供检索用的主题词条，应采用能覆盖论文主要内容的通用技术词条(参照相应的技术术语标准)。按词条的外延层次排列（外延大的排在前面）。摘要与关键词应在同一页。

**2.英文摘要和英文关键词**

英文摘要内容与中文摘要相同，以250—400个实词为宜。摘要下方另起一行注明英文关键词（Keywords3—5个）。\*)

**关键词： 网络爬虫 词频统计 地图 优化** (\*关键词1；关键词2；……

关键词一般为论文中出现的重要的、描述主题的词；

应包含技术领域和问题领域两大类关键字；

尽量不要使用组合词；

中英文关键词不少于3个，不多于5个。\*)

拼字游戏是一款起源于西方国家的文字类益智游戏，深受许多玩家。现有的大多数拼字游戏都是从指定词典取词后生成拼字地图并给予玩家少许提示以完成拼图。本次设计主要针对传统的拼字游戏进行创新，增加了从网络中下载实时新闻、学术类文章，让玩家阅读后从文章中选词填入等功能。选取的均为词频较低或较为生僻的词汇。

通过网络爬虫技术设计模块从指定网站上获取英文文章，通过词频数据库分析文章词汇并得出词频最低的若干词汇。最后基于选取的词汇通过拼字地图生成算法生成地图。同时，分别通过设置缓存、多次生成并评分的优化策略，词频数据库的性能和拼字地图的质量得到了有效的提升。这款动态拼字游戏，会给玩家带来不一样的游戏体验。

ABSTRACT

Scrabble game is a type of popular literal puzzle game originated from the west. Currently most of the scrabble games pick up words from a specified dictionary, use them to generate a scrabble map for players and provide with several tips. Based on this traditional design model, I would like to make some innovations, such as downloading news or academic articles automatically from the websites, words picking from these articles according to their statistic frequencies. Also for players, their task is not remembering the words in the dictionary, but reading and paying attention to those long and uncommon words. Thanks to the advanced and intelligent web crawler technique, lots of plain texts can be scratched back with high efficiency for analysis.

(\* 英文摘要不少于500单词，严格使用英文标点符号。中、英文摘要意思要基本相同。在撰写完英文摘要后，请执行一次自动拼写检查，以减少英文拼写错误的可能性。\*)

**Keywords: web crawler, frequency statistic, map, optimization**  (\* Keywords 1; Keywords 2; ... ...

中英文关键词要严格对译。\*)

目 录

(\* 目录放在正文之前，中英文摘要之后。论文目录是论文的提纲，也是论文各章节组成部分的小标题。要求标题层次清晰，目录中的标题要与正文中的标题一致。

目录至少有二级目录，即包含正文的章和节及其页码；最好有三级目录，即包含正文的章、节和小节及其页码。目录还要包含参考文献及其页码。中英文摘要及目录本身及其页码可不出现在目录中。请使用由Word等排版软件自动生成目录，不要自己手工做目录。\*)

第一章 概述/引言 3

1.1 XXX问题的背景和意义 4

1.2 XXX问题的描述 4

1.3 本文的工作 4

1.4 论文结构简介 4

第二章 XXX综述 5

2.1 ××× 5

2.2 ××××××× 6

2.3 ××× 6

2.3.1 ××× 6

第三章 提出的×××方法 7

3.1 ×××问题描述 7

3.1.1 ×××××× 7

3.1.2 ×××××× 7

3.2 ××××方法 7

3.2.1 ×××××× 7

3.2.2 ×××××× 7

第四章 ×××算法 8

4.1 ×××××× 8

4.1.1 ××× 8

第五章 仿真/实验结果与分析 9

5.1 ×××××× 9

5.1.1 ××× 9

第六章 总结与展望 10

6.1 ×××××× 10

6.1.1 ××× 10

参考文献 11

相关的科研成果目录 13

致 谢 14

附 录 15

# 概述/引言

(\*正文是毕业论文的主体和核心部分，不同学科专业和不同的选题可以有不同的写作方式。正文一般包括以下几个方面：

**1.引言或背景**

引言是论文正文的开端，应包括毕业论文选题的背景、目的和意义；对国内外研究现状和相关领域中已有的研究成果的简要评述；介绍本项研究工作研究设想、研究方法或实验设计、理论依据或实验基础；涉及范围和预期结果等。要求言简意赅，注意不要与摘要雷同或成为摘要的注解。

**2.主体**

论文主体是毕业论文的主要部分，必须言之成理，论据可靠，严格遵循本学科国际通行的学术规范。在写作上要注意结构合理、层次分明、重点突出，章节标题、公式图表符号必须规范统一。论文主体的内容根据不同学科有不同的特点，一般应包括以下几个方面：

（1）毕业论文（设计）总体方案或选题的论证；

（2）毕业论文（设计）各部分的设计实现，包括实验数据的获取、数据可行性及有效性的处理与分析、各部分的设计计算等；

（3）对研究内容及成果的客观阐述，包括理论依据、创新见解、创造性成果及其改进与实际应用价值等；

（4）论文主体的所有数据必须真实可靠，凡引用他人观点、方案、资料、数据等，无论曾否发表，无论是纸质或电子版，均应详加注释。自然科学论文应推理正确、结论清晰；人文和社会学科的论文应把握论点正确、论证充分、论据可靠，恰当运用系统分析和比较研究的方法进行模型或方案设计，注重实证研究和案例分析，根据分析结果提出建议和改进措施等。

**3.结论**

结论是毕业论文的总结，是整篇论文的归宿，应精炼、准确、完整。结论应着重阐述自己的创造性成果及其在本研究领域中的意义、作用，还可进一步提出需要讨论的问题和建议。

正文不少于10千字；或使用小四字体、1.5倍行距、A4纸版式排版时不少于10页纸。正文须有页码，从第1页开始编页码。正文采用章、节、小节组织。章的标题使用“第一章”等字样开头，节的标题采用“1.1”等字样开头，表示第一章的第一节，小节的标题采用“1.1.1”等字样开头，表示第一章的第1.1小节。正文章、节、小节标题与正文段落使用不同的字体，并且之间有适当的间距。正文段落要统一缩进两个汉字。

行文时注意语句通顺，条理清晰；每章节开头部分需要有承上启下描述，先简要介绍本章节内容，再展开详细描述。

第一章作为概述，也是完整的短文，体现全文的内容。

\*)

## 拼字游戏创新的背景和意义

随着计算机技术的不断发展，计算机俨然已成为人们生活中不可或缺的元素。现今人们在精神上的追求越来越高，对游戏的种类和玩法的创新需求呼之欲出。依靠计算机硬件和软件的迅猛发展，计算机平台上的游戏的高速发展也正迎合了人们追求新颖、刺激、流畅的游戏效果的心态。具体地说，当前计算机平台上的游戏种类多样化程度越来越高，游戏的效果愈加炫丽多彩，游戏操作的流畅性也得到了大幅度的提高。许多历史悠久的游戏通过计算机平台，由实体型转变成了虚拟型。这些游戏的基本形式得到更加生动的表示，并且随着时间的迁移，人们不断在已有的游戏基础形式上派生出其他的新颖的形式与玩法，使得这些古老的游戏能够历久弥新，继续传承下去。

拼字游戏在1938年由建筑师Alfred Mosher Butts设计出来，是一款文字型益智游戏。根据拼字游戏最初的设计，玩家需要从指定的字典中选取符合要求的单词填入拼字地图格中并满足一定要求。一直以来，拼字游戏以其对玩家英语技能的锻炼以及其趣味性广受人们的好评。具体而言，传统的拼字游戏对玩家的单词记忆、联想能力进行了考验。而现代的创新意识驱使着人们对这类传统文字游戏进行改进和创新，使其在游戏流畅性、趣味性、综合性上日渐提高。而本次设计，即针对传统的拼字游戏在玩法上进行创新，融入了即时文章源下载的思想，不仅锻炼了玩家的联想能力，也锻炼了玩家的阅读能力，可谓一石二鸟。

由于计算机平台的多样性，大部分计算机语言只能通过在不同的平台上使用不同的代码来设计游戏。这使得一个计算机游戏拥有跨平台性的成本大大增加，对游戏的多平台化发展造成了影响。而Java语言与生俱来的跨平台性却很好地解决了这个问题。Java语言从1991年创建到现在，已经逐渐变得成熟，在越来越多的软件设计需求中发挥其优势。Java的Swing套件使用纯Java语言编写，在Java虚拟机中运行效率高，已经成为许多中小型游戏开发者的选择。虽然Java语言并不是为游戏编程而诞生，但是其跨平台、高性能等优点已经能在游戏编程中得到充分体现。

## 游戏设计难点描述

拼字游戏作为一种传统的文字类游戏，在玩法上已然被广大玩家所知晓。游戏的操作简单，然而在游戏的过程中，一些玩家由于自身水平的限制，常常受困于游戏难题中许久不能得到答案，使得传统拼字游戏的流畅性不如其他类型的游戏（如战略竞技类游戏）。如此，拼字游戏给玩家带来的能力的提升受到了限制，其游戏体验也并不能吸引到大量的玩家。因此，本次设计主要针对传统拼字游戏玩法较为古老且流畅性低的特点，希望对其进行玩法上的改进和创新。在对玩家英语技能的锻炼方面，希望能增加英语快速阅读与记忆的能力，并减少漫无目的的单词联想行为所占的比例，提升玩家的答题速度，进而增加游戏的流畅性和紧张性。

在游戏形式上，本次设计的动态拼字游戏"Dyscrabble"将从网络中提取实时性文章，分析文章的词汇并筛选出符合要求的若干词汇。进而针对这些词汇为玩家玩成待填的拼字地图。玩家需要通过阅读文章，对文章中较为生僻的词汇进行记忆，以便在填字流程中快速地填入词汇。

在技术实现上，可以将Dyscrabble分为三部分：在线新闻类文章的抓取与下载、文章的词汇词频分析与筛选、填字地图的生成。对于文章的抓取，如今网络上有许多的成型的网络爬虫可供参考。也有专门的网络爬虫生成框架，可以快速地为开发者定制针对特定网站的网络爬虫。本次设计的网络爬虫部分采用Python语言实现，未采用任何已有的框架。在网络爬虫的开发过程中，如何为特定网站定制爬虫对该网站的URL进行提取和筛选以及对HTML文档中有用信息的筛选成为难点。这需要设计者充分了解该网站的服务器中网页文件的组织形式，以及这些网页文件内容的构成与排序。

对于词汇词频的分析筛选，现有的许多功能强大的单机数据库依靠其高度的存取优化技术为分析筛选操作的高效性提供了有力的支持。然而在实际开发中设计者意识到，由于文章中词汇的数量较大，且不少高频词汇在文章中频频出现，频繁的数据库存取操作使得操作性能不高。因此该部分的难点在于如何设计一个合理的缓存机制，减少系统访问数据库的次数，提高词频分析的效率。

而在地图生成过程中，系统需要合理地规划地图中每个单词的存放位置，使得地图中的单词个数最多，生成的地图复杂程度最高。同时为了增加游戏的不确定性，生成的地图需要具有一定的随机性，使得一篇文章能对应生成多种不同的地图，解决从网络中获取文章资源的不足问题。具体地说，如何在词汇选取、词汇排放的策略中融入随机化的思想也是设计地图生成算法时需要思考的问题。

## 本文的工作

(\*阐述本文的主要工作，即简要描述提出的方法、创新点、结果。\*)

针对1.1中对Dyscrabble动态拼字游戏设计目的的描述以及1.2中对设计过程中对实现难点的分析，本文主要描述在Dyscrabble的整个设计开发过程中使用到的方法，实现技术的创新点以及创新的结果。

对于网络爬虫模块，设计中会使用Python语言进行爬虫开发。特别地，Python中的网络访问动态库对网络的访问速度、解码速度与准确性进行了提升，如urllib2、HTMLParser等。在网页文件的分析与有用信息的筛选中，Python的正则表达式模块re起到关键作用。re模块通过订制需要获取的信息的正则表达式形式，快速地在HTML文件中寻找符合正则表达式形式的内容。同时，正则表达式对已获取信息的进一步加工与筛选也起到了很大的作用。在解码方面，Python的decode、encode函数能够方便高效地解码网络内容。网络中许多英文网站为了兼容英语以及一些西语字符，多采用iso-8859字符集。但由于网站开发者的目的与水平并不一致，因此并不是每个英文网站都采用同种编码。甚至有的网站网页文件采用的编码字符集与网页属性栏中显示的字符集不一致，由此造成解码的困难。而HTMLParser模块在HTML文本信息的提取、解码方面首屈一指，能有效解决这种问题。

在词频统计模块中，设计者采用了轻便小巧的单机型数据库sqlite3进行词汇词频信息的存放。通过在数据堂中下载拥有20万

## 论文结构简介

(\*简单介绍论文后面章节的安排和主要内容。\*)

# XXX综述

(\* 第二章综述与本文提出方法相关的工作，要分类别给出相关工作的分析评价、缺点在什么地方，而本文工作改进在什么地方，是怎样解决这些缺点的。

每一章的开头要另起一页。

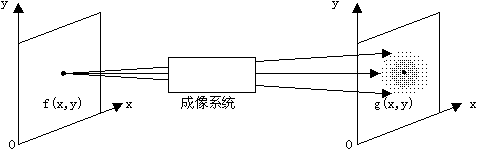
注意：第一、二章是引用文献最集中的章节，但篇幅不要超过全文的 1/3，后续章节尽量不要再出现大段引用的现象。

\*)

## ×××

(\* 正文中有图、表和公式时必须对图、表和公式按章进行编号。例如，下图编号“图2-1”表示第二章第1个图；或者，在整篇正文中，图和表按出现的顺序依次编号，如“图1”、“图2”、“表1”、“表2”等。

图和表必须有标题（如图2-1中的“成像系统原理”），图的编号和标题放在图的下方，表的编号和标题放在表的上方。所有的图和表都必须在正文中引用，而且引用时不得采用“如下图”或“如下表”等字样，而要采用“如图2-1”或“如表2-1”等字样。排版时，可根据排版的美观性适当调整图和表在正文中的放置位置。



**图2-1：成像系统原理**

**表2-1： 表的样例**

|  |  |  |
| --- | --- | --- |
| xxx | yyy | zzz |
| aaa | 1234 | 5678 |

## ×××××××

×××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××**[1]**，此蓝色方括号上标是参考文献的引用，方括号内的数字指明前面这段文字引自第几个参考文献。

××××：

列举的公式也必须按章进行编号，例如公式2.1表示第二章第1个公式。

×××××××[2]。（引用参考文献2）

××××××××[[1]](#footnote-1)。（注释）

## ×××

### ×××

#### 列表项一

#### 列表项二

#### 列表项三

# 提出的×××方法

(\*

论文一般从本章开始介绍自己的工作。这是论文最主要的部分，在这一部分表达你自己在论文部分的工作，而不像前一章综述那样主要介绍别人的工作。千万不要大量列举你的源代码（部分核心算法可使用伪码或流程图等形式表示），否则给人的感觉是你的论文没有内容而靠这些来充数。

\*)

## ×××问题描述

(\*

本小节可以更详细的描述本文研究的问题。

\*)

### ××××××

×××××××[5] (\* 引用参考文献5 \*)

××××××××[[2]](#footnote-2)。

### ××××××

## ××××方法

(\*

本小节可以详细的描述本文提出的方法原理。

\*)

### ××××××

×××××××[5] (\* 引用参考文献5 \*)

××××××××[[3]](#footnote-3)。

### ××××××

# ×××算法

(\*

如果本文针对所提的方法设计了相关的算法、协议，可以在本章详细描述。同时要从理论上证明算法的正确性，分析算法的复杂度，注意要与其他算法比较。

\*)

## ××××××

### ×××

# 仿真/实验结果与分析

(\*

本章需描述针对本文所提的方法及其相关方法所做的仿真或实验，并给出结果比较与分析。具体要阐明仿真/实验环境及参数设置、评价指标、结果图表与具体分析，通过与其他相关方法的比较证明本文方法的正确性、有效性。

\*)

## ××××××

### ×××

# 总结与展望

(\* 论文第三部分主体：总结、比较与展望。

1. 给出本文的结论：总结自己的工作/贡献/创新点，指出自己工作的优点在哪里，比较你的工作与他人工作的优、劣。

2）分析系统的不足之处在哪里、指明以后改进或努力的方向；

3）也可以总结一下本论文工作开展的经验和教训。

\*)

## ××××××

### ×××

# 参考文献

(\* 列出你在完成论文过程主要参阅的论文与著作。其中的篇数不宜太少，否则让人感觉到你没有参考相关工作就动手做论文；一篇本科毕业论文的典型参考文献是15至30篇，如果你所列参考文献数目不足5篇，则肯定无法通过论文初审的形式检查。列举参考文献时，按论文中引用文献的先后顺序列于此处；注意正文中必须引用此处所列的全部参考文献，而且引用顺序就是参考文献的列举顺序。参考文献的著录应符合国家标准，参考文献的序号左顶格，并用数字加方括号表示，如“[1]”。每一条参考文献著录均以“.”结束。具体各类参考文献的编排格式如下：

1、文献是**期刊**时，书写格式为：

[序号] 作者. 文章题目[J]. 期刊名, 出版年份，卷号(期数):起止页码.

2、文献是**图书**时，书写格式为：

[序号] 作者. 书名[M]. 版次. 出版地：出版单位，出版年份：起止页码.

3、文献是**会议论文集**时，书写格式为：

[序号] 作者. 文章题目[A].主编.论文集名[C], 出版地：出版单位，出版年份:起止页码.

4、文献是**学位论文**时，书写格式为：

[序号] 作者. 论文题目[D].保存地：保存单位，年份.

5、文献是来自**报告**时，书写格式为：

[序号] 报告者. 报告题目[R].报告地：报告会主办单位，报告年份.

6、文献是来自**专利**时，书写格式为：

[序号] 专利所有者. 专利名称：专利国别，专利号[P].发布日期.

7、文献是来自**国际、国家标准**时，书写格式为：

[序号] 标准代号. 标准名称[S].出版地：出版单位，出版年份.

8、文献来自**报纸文章**时，书写格式为：

[序号] 作者. 文章题目[N].报纸名，出版日期（版次）.

9、文献来自**电子文献**时，书写格式为：

[序号] 作者.文献题目[电子文献及载体类型标识].电子文献的可获取地址，发表或更新日期/引用日期（可以只选择一项）.

电子参考文献建议标识：

**［DB/OL］**——联机网上数据库(database online)  
**［DB/MT］**——磁带数据库(database on magnetic tape)  
**［M/CD］** ——光盘图书(monograph on CD-ROM)  
**［CP/DK］**——磁盘软件(computer program on disk)  
**［J/OL］** ——网上期刊(serial online)  
**［EB/OL］**——网上电子公告(electronic bulletin board online)

\*)

1. 袁崇义. Petri网原理. 北京：电子工业出版社. 1998.
2. Bloch J.. Effective Java: Programming Language Guide. Addison Wesley. 2001.
3. 李建中, 李金宝, 石胜飞. 传感器网络及其数据管理的概念、问题与进展. 软件学报. 2003, 14(10): 1717-1727.
4. Goguen J. A.. Parameterized Programming. IEEE Transactions on Software Engineering. 1984, 10(5): 528-543.
5. Snyder A.. Encapsulation and Inheritance in Object-Oriented Programming Languages. In Proceedings of Annual ACM SIGPLAN Conferences on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA’86). 1986. 38-45.
6. … …

# 相关的科研成果目录

包括本科期间发表的与毕业论文相关的已发表论文或被鉴定的技术成果、发明专利等成果，应在成果目录中列出。此项不是必需项，空缺时可以略掉

# 致 谢

(\*谢辞应以简短的文字对课题研究与论文撰写过程中曾直接给予帮助的人员(例如指导教师、答疑教师及其他人员)表示对自己的谢意，这不仅是一种礼貌，也是对他人劳动的尊重，是治学者应当遵循的学术规范。内容限一页。\*)

# 附 录

(\*对于一些不宜放在正文中的重要支撑材料，可编入毕业论文的附录中。包括某些重要的原始数据、详细数学推导、程序全文及其说明、复杂的图表、设计图纸等一系列需要补充提供的说明材料。如果毕业设计(论文)中引用的实例、数据资料，实验结果等符号较多时，为了节约篇幅，便于读者查阅，可以编写一个符号说明，注明符号代表的意义。附录的篇幅不宜太多，一般不超过正文。

论文附录依次用大写字母“附录A、附录B、附录C……”表示，附录内的分级序号可采用“附A1、附A1.1、附A1.1.1”等表示，图、表、公式均依此类推为“图A1、表A1、式A1”等。

\*)

（\*

**声明**

**以上格式模板只作参照使用，自己验证是否都满足一下中山大学教务处的官方格式要求！！！！！！！！！**

**毕业论文的撰写格式要求**

（一）字数

除有特殊要求的专业外，毕业论文正文一般不少于5000字。各专业可根据需要确定具体的文字和字数要求，并报教务处备案。

（二）字体和字号

论文题目 三号宋体加粗

各部分标题 四号黑体

中文摘要、关键词标题 五号黑体并加方括号

中文摘要、关键词内容 五号楷体

英文摘要、关键词标题 小四号新罗马体（Time New Roman）加粗并加方括号

英文摘要内容 小四号新罗马体（Time New Roman）

目录标题 三号宋体加粗

目录内容中章的标题 四号黑体

目录中其他内容 小四号宋体

正文 小四号宋体

注释、参考文献标题 小五号黑体并加冒号

注释、参考文献内容 小五号宋体

致谢、附录标题 四号黑体

致谢、附录内容 小四号宋体

论文页码 页脚居中、阿拉伯数字（五号新罗马体）连续编码

（三）关键词

摘要正文下方另起一行顶格打印“关键词”款项，每个关键词之间用“；”分开，最后一个关键词不打标点符号。

（四）目录

目录应另起一页，包括论文中的各级标题，按照“一……”、“（一）……”或“1……”、“1.1……”格式编写。

（五）各级标题

正文各部分的标题应简明扼要，不使用标点符号。论文内文各大部分的标题用“一、二……（或1、2……）”，次级标题为“（一）、（二）……（或1.1、2.1……）”，三级标题用“1、2……（或1.1.1、2.1.1……）”，四级标题用“（1）、（2）……（或1.1.1.1、2.1.1.1……）”。不再使用五级以下标题。

（六）名词术语

1、科学技术名词术语尽量采用全国自然科学名词审定委员会公布的规范词或国家标准、部标准中规定的名称，尚未统一规定或叫法有争议的名词术语，可采用惯用的名称。

2、特定含义的名词术语或新名词、以及使用外文缩写代替某一名词术语时，首次出现时应在括号内注明其含义，如：OECD（Organisation for Economic Co-operation and Development） 代替经济合作发展组织。

3、外国人名一般采用英文原名，可不译成中文，英文人名按姓前名后的原则书写，如：CRAY P，不可将外国人姓名中的名部分漏写，例如：不能只写CRAY, 应写成CRAY P。一般很熟知的外国人名(如牛顿、爱因斯坦、达尔文、马克思等)可按通常标准译法写译名。

（七）物理量名称、符号与计量单位

1、论文中某一物理量的名称和符号应统一，一律采用国务院发布的《中华人民共和国法定计量单位》。单位名称和符号的书写方式，应采用国际通用符号。

2、在不涉及具体数据表达时允许使用中文计量单位如“千克”。

3、表达时刻应采用中文计量单位，如“下午3点10分”，不能写成“3h10min”，在表格中可以用“3:10PM”表示。

4、物理量符号、物理量常量、变量符号用斜体，计量单位符号均用正体。

（八）数字

1、无特别约定情况下，一般均采用阿拉伯数字表示。

2、年份一律使用4位数字表示。

3、小数的表示方法：一般情形下，小于1的数，需在小数点之前加0。但当某些特殊数字不可能大于1时（如相关系数、比率、概率值），小数点之前的0要去掉，如r=.26，p<.05。

4、统计符号的格式：一般除μ、α、β、λ、ε以及V等符号外，其余统计符号一律以斜体字呈现，如*ANCOVA，ANOVA，MANOVA，N，nl，M，SD，F，p，r*等。

（九）公式

1、公式应另起一行写在稿纸中央。一行写不完的长公式，最好在等号处转行，如做不到这一点，可在运算符号（如“﹢”、“﹣”号）处转行，等号或运算符号应在转行后的行首。

2、公式的编号用圆括号括起，放在公式右边行末，在公式和编号之间不加虚线。公式可按全文统编序号，也可按章独立序号，如（49）或（4.11）。采用哪一种序号应和图序、表序编法一致。不应出现某章里的公式编序号，有的则不编序号。子公式可不编序号，需要引用时可加编a、b、c……，重复引用的公式不得另编新序号。公式序号必须连续，不得重复或跳缺。

3、文中引用某一公式时，写成“由式（16.20）”。

（十）表格

1、表格必须与论文叙述有直接联系，不得出现与论文叙述脱节的表格。表格中的内容在技术上不得与正文矛盾。

2、每个表格都应有自己的标题和序号。标题应写在表格上方正中，不加标点，序号写在标题左方。

3、全文的表格可以统一编序，也可以逐章单独编序。采用哪一种方式应和插图、公式的编序方式统一。表序必须连续，不得跳缺。

4、表格允许下页接写，接写时标题省略，表头应重复书写，并在右上方写“续表××”。多项大表可以分割成块，多页书写，接口处必须注明“接下页”、“接上页”、“接第×页”字样。

5、表格应放在离正文首次出现处最近的地方，不应超前和过分拖后。

（十一）图

1、插图应与文字内容相符，技术内容正确。所有制图应符合国家标准和专业标准。对无规定符号的图形应采用该行业的常用画法。

2、每幅插图应有标题和序号，全文的插图可以统一编序，也可以逐章单独编序，如：图45或图6.8。采取哪一种方式应和表格、公式的编序方式统一。图序必须连续，不重复，不跳缺。

3、由若干分图组成的插图，分图用a、b、c……标序。分图的图名以及图中各种代号的意义，以图注形式写在图题下方，先写分图名，另起行写代号的意义。

4、图与图标题、图序号为一个整体，不得拆开排版为两页。当页空白不够排版该图整体时，可将其后文字部分提前，将图移至次页最前面。

5、对坐标轴必须进行文字标示，有数字标注的坐标图必须注明坐标单位。

（十二）注释

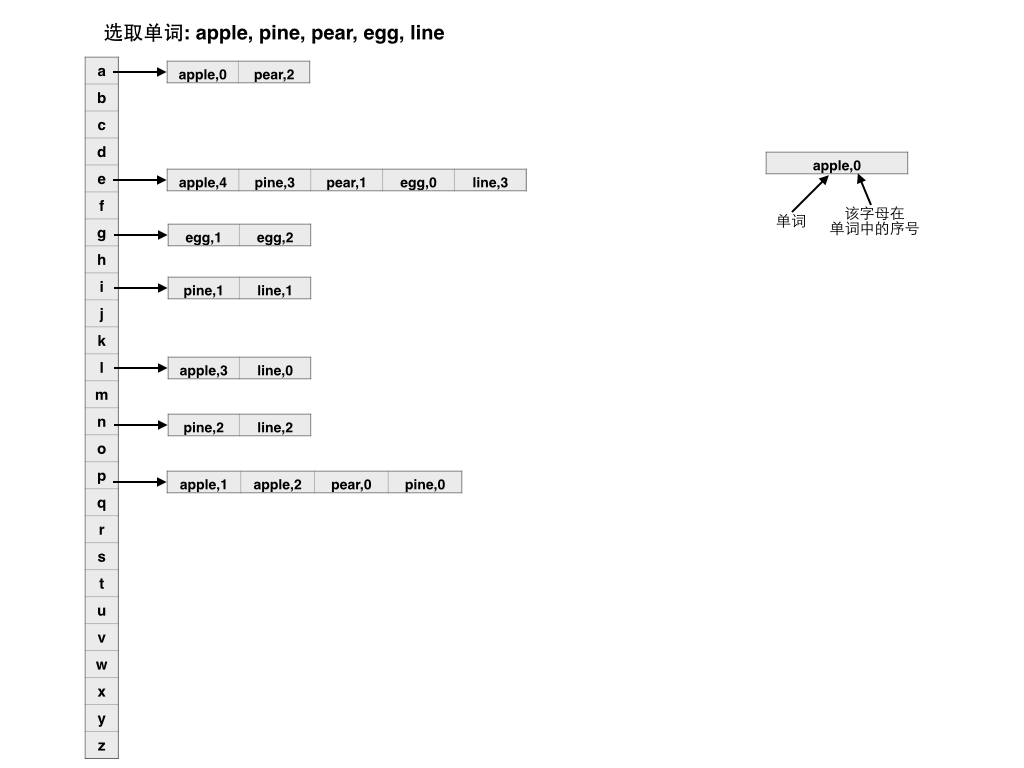
毕业设计(论文)中有个别名词或情况需要解释时，可加注说明。注释采用篇末注，应根据注释的先后顺序编排序号。注释序号以“①、②”等数字形式标示在被注释词条的右上角。篇末注释条目的序号应按照“①、②”等数字形式与被注释词条保持一致。 \*）

拼字地图的生成。

问题描述：

在地图生成模块中，系统需要根据选出的单词，布置它们在地图格中的位置，使得每个单词都能与另外至少一个单词在某个字母上进行横纵交汇，即共享某个格子中的字母。在地图的质量方面，若能填入地图的单词数目越多（最多不超过选出的单词数），地图中横纵交汇点越多，地图的总面积越小，则质量越高。

方法：

地图生成算法将从横纵交汇点入手。首先建立一个字母表索引表。对于每个单词，分别以其每个字母作为其索引存入索引表中。索引表的示意图如下：

索引表中每个索引项的类定义如下：

//database entry for processed data

**public** **class** StoreIndex {

**char** letter; //the indexed letter

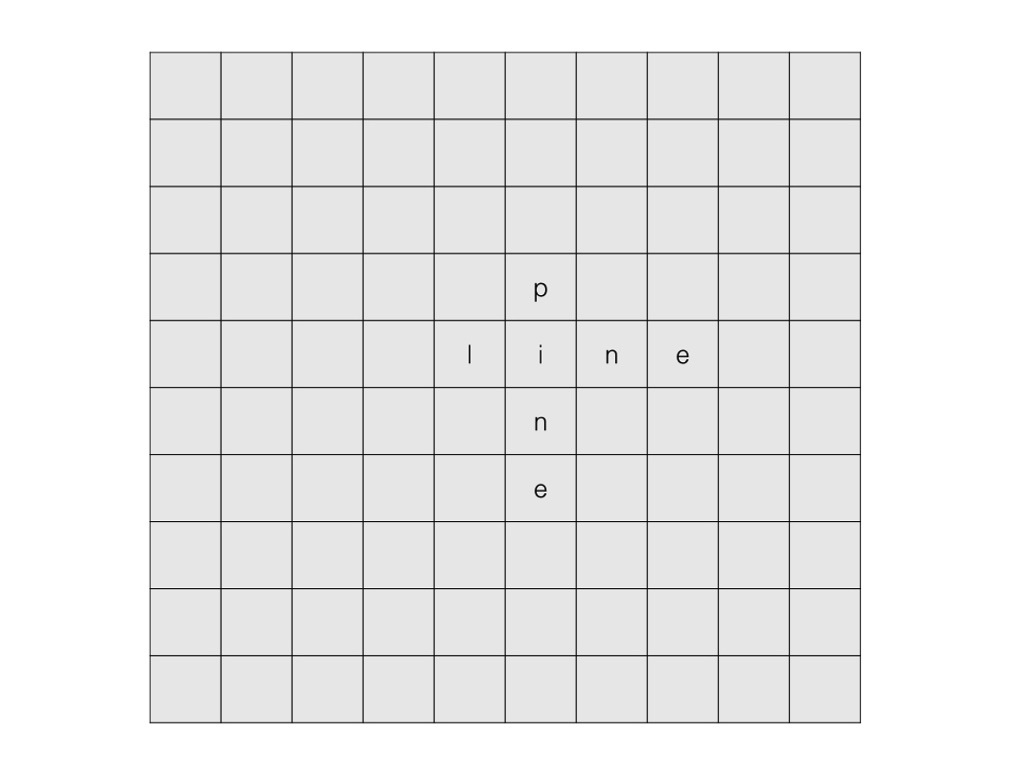
**int** wordIndex; //index of word in the original word list

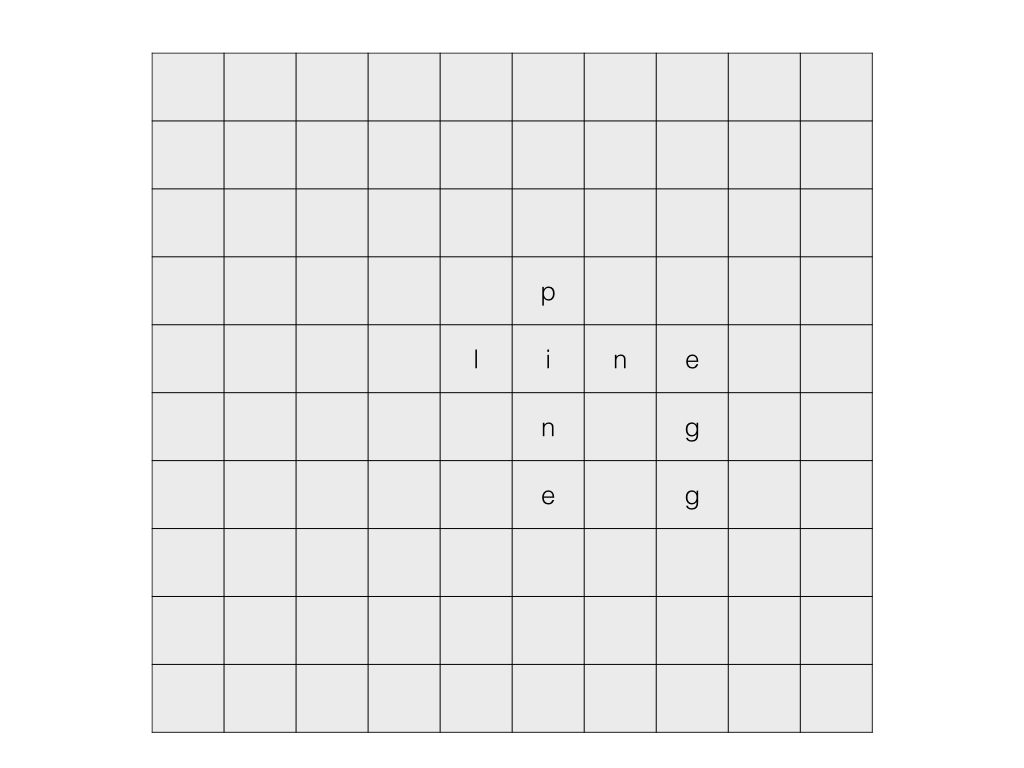
**int** alphIndex; //index of the letter in this word

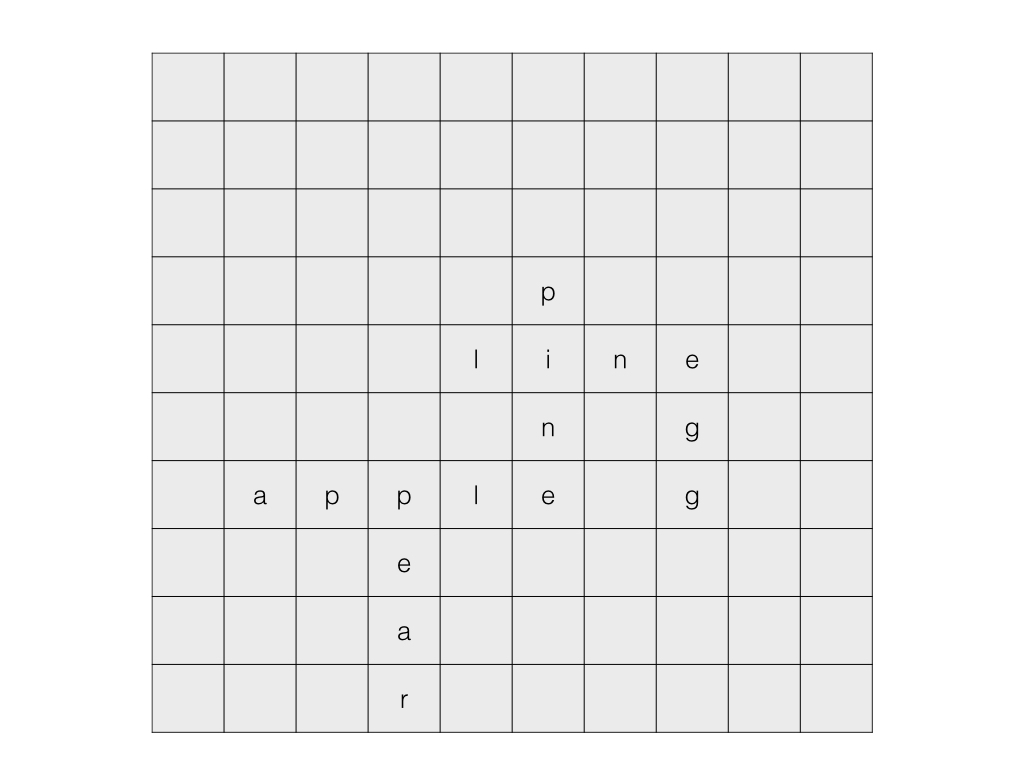
**int** length; //length of this word

}

即，每个索引项会存储该索引项的索引字母、索引单词在待填词汇表中的序号、该字母在该单词中的序号以及单词的长度。

首先，系统将从字母索引表中随机找出至少存储了两个不同单词的字母项，如示意图中的字母i。（字母g是不符合要求的）在该字母的表项中随机选出两个单词，以字母i为中心进行横纵交叠陈列在地图的中心位置，示意图如下所示：

然后以交汇点i为起点，通过广度优先算法，将其四周的4个格子加入到队列中。每次从队列中取出头元素，查找该元素所存字母所应的索引表中是否有可以在该处填入的单词。若有，则填入后将新的交汇点也加入到队列中，直到队列为空。如下图所示：

得到最终的地图：

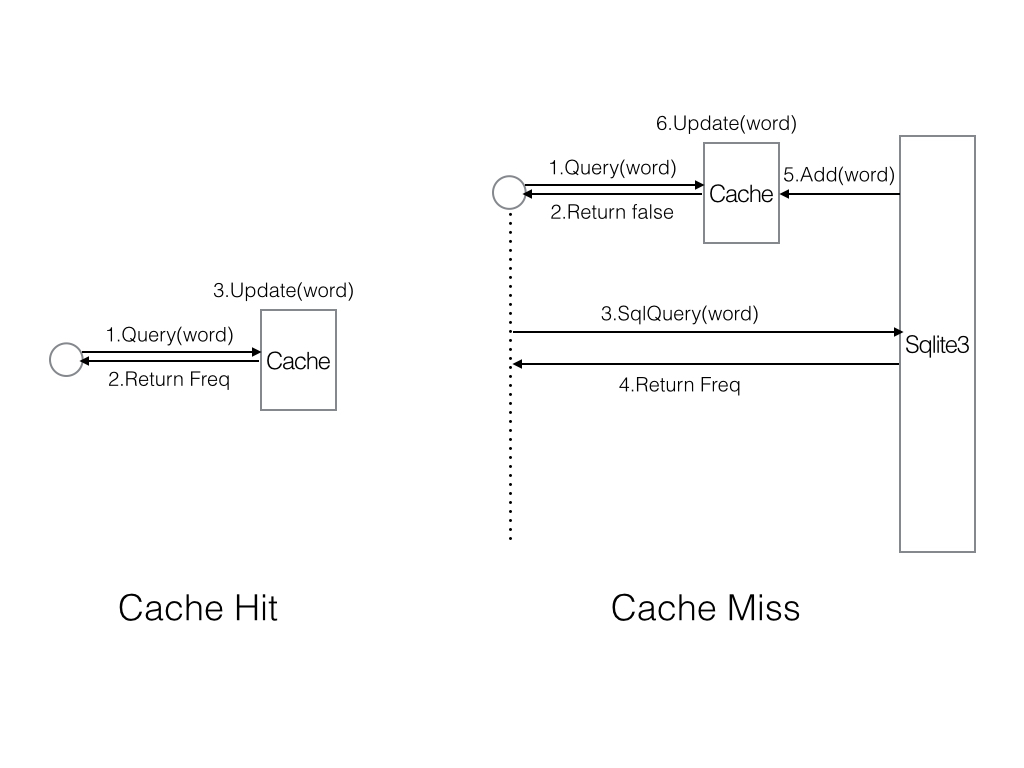
词汇处理与词频统计：

问题描述：

本模块需要将文章中的每个词进行还原处理，并在词频数据库中查找并更新处理后的单词词频。

方法：

词频统计：词频统计即通过对词频数据库的查询找出该单词的绝对词频，在处理完毕后同时更新该词频的操作。系统中使用了轻量级的单机数据库sqlite 3进行词频的统计。在程序安装时候会从初始词频文件Lex\_ratio.txt中读取原始词频信息并创建本地数据库。在之后的游戏过程中，数据库将会根据遇到的单词相应地更新词频。

由于文章中会存在大量重复的高频常用词（如the, this），重复地对这些词汇进行数据库查询操作非常浪费时间。因此设置一个词频缓存来存储已经遇到过的单词能为查询操作提高效率。当查询一个词汇时，系统会首先查询缓存中是否已经存在该单词。若存在则直接返回词频，否则再通过数据库进行查询，并将查询的结果加入到缓存中。查询的示意图如下：

查询的伪代码如下：

wordQuery(word)

if (cache.contain(word))

then

res <- cache.getValue(word) + 1

res <- cache.setValue(word)

else

res <- sqlQuery(word) + 1

cache.add(word,res)

return res

一篇文章的所有词汇以及其词频最终都会记录在一个HashMap中。该HashMap将用于用户填完答案后的校对检验工作。直接调用系统的sort函数并找出词频最低的若干个词汇。至此词频统计与词汇选取操作完成。

地图生成算法及其优化：

如上所述，地图的生成是通过BFS(广度优先搜索)来实现的。先在地图靠近中央的某一点填入两个横纵交汇的词汇，并以交汇点为搜索起点，将其四周的4个格子节点加入到队列中。循环开始，每次从队列中取出头元素，根据字母索引表判断其是否可作为与一个新单词的交汇点。若能则填入词汇并更新索引表。最后将其四周的节点加入队列。地图生成算法大致可分为以下子模块：搜索子模块、填词子模块。

在介绍各个子模块之前，需要先对地图节点的存储格式及其信息作介绍。

地图是一个的方阵。每个地图的格子称为一个节点。在图形上与一个节点共享一条边的节点称为该节点的邻节点。每个节点的类定义如下：

**public** **class** BasicNode {

**int** x; //x coordinate

**int** y; //y coordinate

**int** horizontal; //whether it is filled horizontally

**int** vertical; //whether it is filled vertically

**char** content; //the content of the node

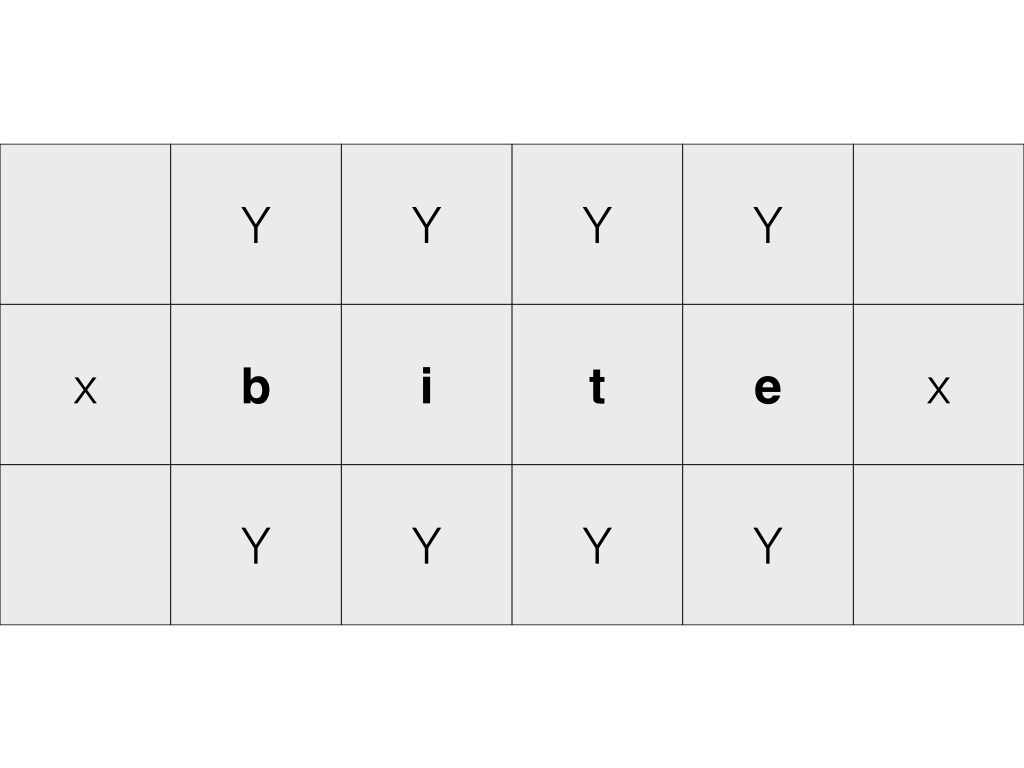
**boolean** consider; //will the node be considered by the waitQueue

}

每个节点存储该节点的横纵坐标、存储的字母、是否会被BFS考虑（后面将详细描述）、是否被横向、纵向填充（后面将详细描述）。

填词子模块：

该子模块主要尝试在某一区域填入某一个单词。造成某个单词无法填入地图主要有以下三个因素：

1. 空间不足

即给定的区域并不能将单词按字母顺序填入。有可能是待填区域中已存在字母，且该字母与待填单词填入该处的字母不一致；也有可能是由于边界限制，不能将所有字母填入。当出现空间不足时，填词失败。

2. 水平阻碍

如上图所示，待填入单词为"bite"，但是在标有“X”的区域存在已填入的字母。此时这些填入的字母会与将被填入的单词连接形成错误的单词，故在这种情况下填词失败。

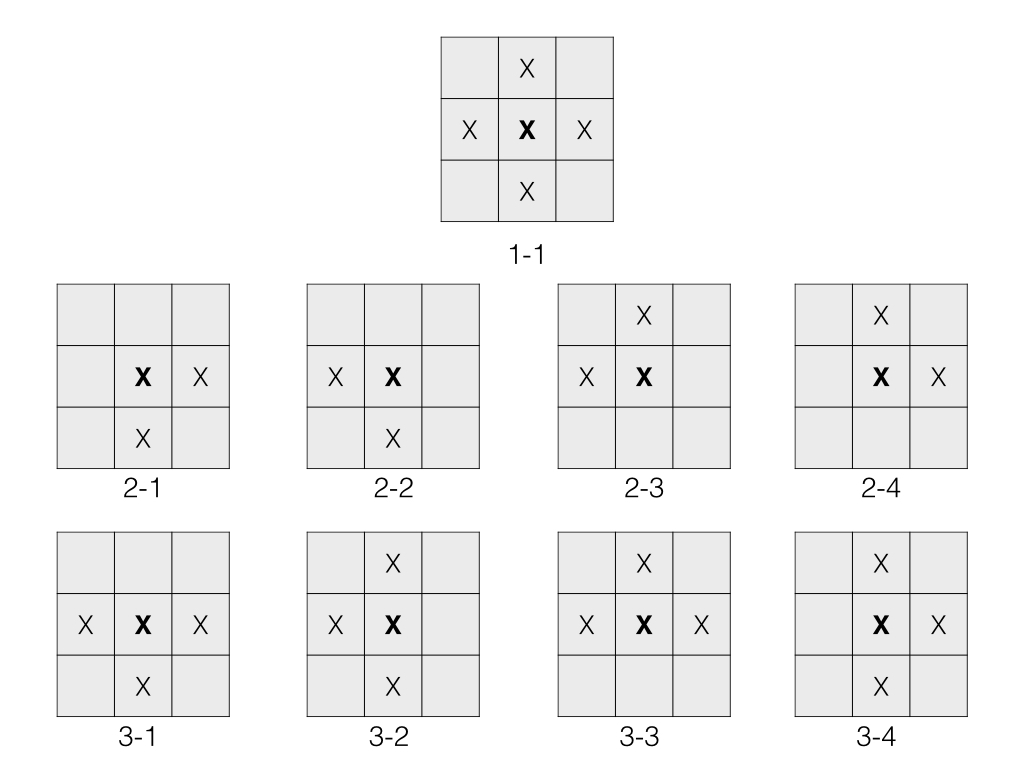
3. 垂直阻碍

如上图所示，待填入单词为"bite"，但是在标有“Y”的区域存在已填入的字母。在不存在空间不足的情况下，这些填入的字母会与将要填入的字母在垂直方向形成许多无意义的单词。故在这种情况下填词失败。

当模块判断某个单词可以正确填入某区域时，将词填入并相应设置占据的地图节点的填入字母、横（纵）向填入情况等属性。

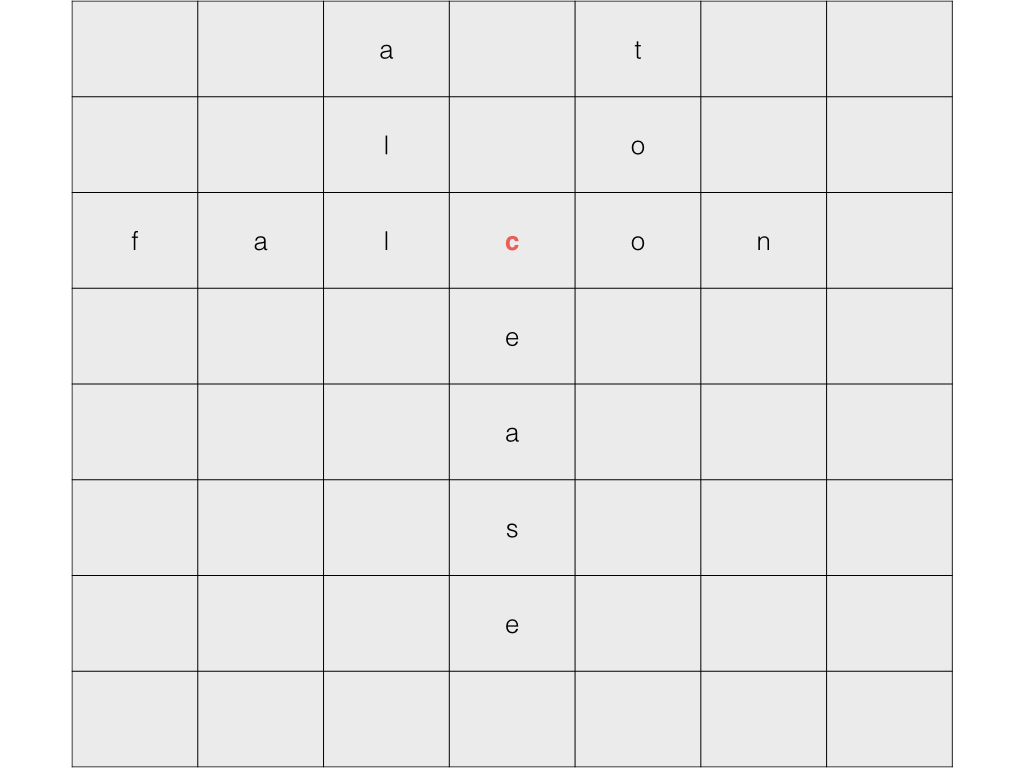
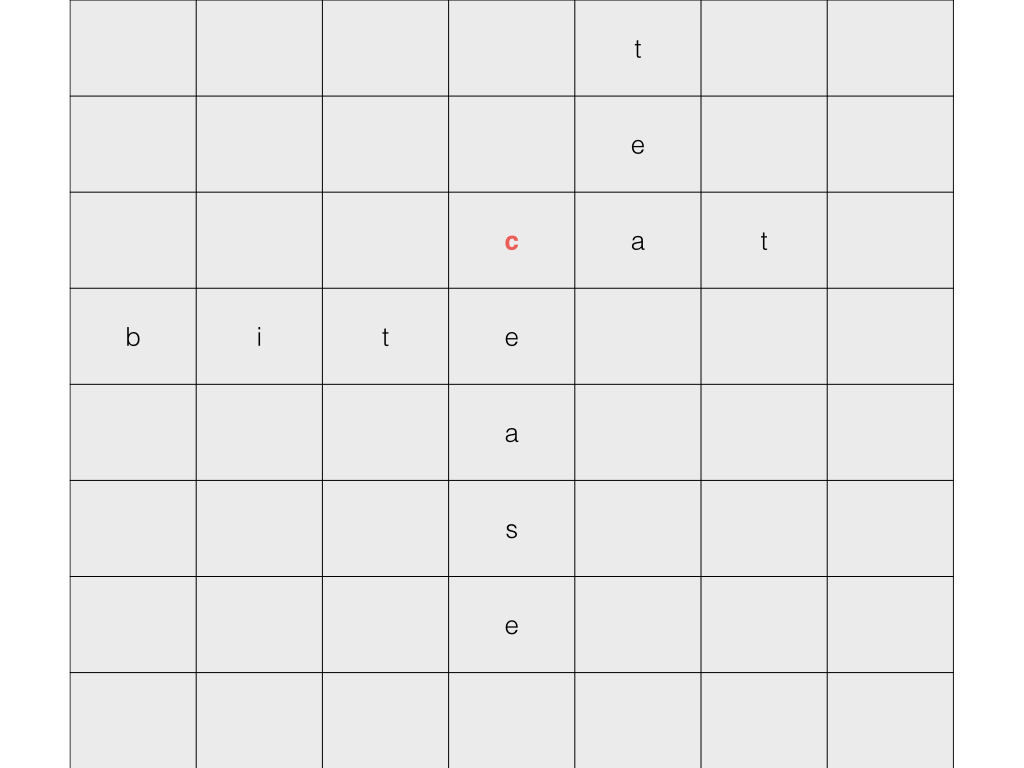
搜索子模块：

为了更好地阐述基于交汇点进行广度优先搜索的地图生成算法，首先对一个交汇点附近的4个节点可能出现的情况进行分类和定义。如下图所示：



图中粗体的“X”表示交汇点。可以看出，2-1、2-2、2-3、2-4是旋转对称的；3-1、3-2、3-3、3-4亦同理。由图易得，周围有2到3个字母的交汇点，必定是至少一个交汇单词的首（尾）字母。由于每个单词只有一个首（尾）字母，故这两种情况所占的比例不大。更多的情况是出现像图1-1的“十”字型。

由于系统规定选取的词汇长度不能小于3，故选取的词汇中不会出现长度为2的词汇，也即两个同方向（横向或纵向）的词汇不能粘在一起。例如，对于两个横向粘在一起（其所在行序号相差为1）的词汇来说，其纵向会产生若干个长度为2的短词，这些短词是很难组成文章中存在的词汇的。故对于一个交汇点来说，将与其距离为2的4个点加入队列中而不是距离为1的4个点能减少进入队列的无用节点数。（但是也有例外情况，将在后文进行介绍）而对于没能形成交汇点的节点，只需直接将其距离为1的四个点加入队列即可。

考虑以下两种情况：

如左图所示，若单词cat与cease的交汇点c为起点。此时与其距离为1的点e和a仍然可以作为交汇点加入单词；同理，右图中起点c的邻居点l和o也仍然能加入新的词汇。左图的情况对应交汇点图2-1到2-4；右图的情况则对应交汇点图3-1到3-4。且这些邻居点若是能够成为交汇点，也一定是新加词汇的首（尾）字母。

故每次向队列中加入新的邻节点时，首先判断交汇点情况为图中的哪一种。若是图1-1的情况（大多数情况），则直接将距离为2的点加入队列，否则，根据情况判断其邻节点是否能加入新的单词。若不能，则直接将距离为2的点加入队列。对于边界情况，可仍然视被边界截去的字母节点是存在的。在该处插入单词势必会无效，从而无需另外讨论。

这种方案能带来一定的性能提升。一般来说，“十”字型交汇点在所有的交汇点中大约占70%，对于每一个这样的交汇点，能减少4个无用的节点进入队列；对于剩下的30%情况，由于邻节点只能作为单词的首（尾）字母接入新单词，故只需判断未用的单词能否接入即可。

地图优化：

上述的地图生成算法在开始阶段选取第一个交汇点的操作是随机的；节点出队列后判断能接入哪些新词也是随机的。因此每个步骤的随机化使得每一步的策略并不一定是当前最优的。（而更可能趋近于平均情况）因此需要通过多次生成的方法，筛选出相对最优的地图。

方法：

要对地图进行多次生成筛选达到优化的目的，首先需要定义衡量地图优化程度的因素。

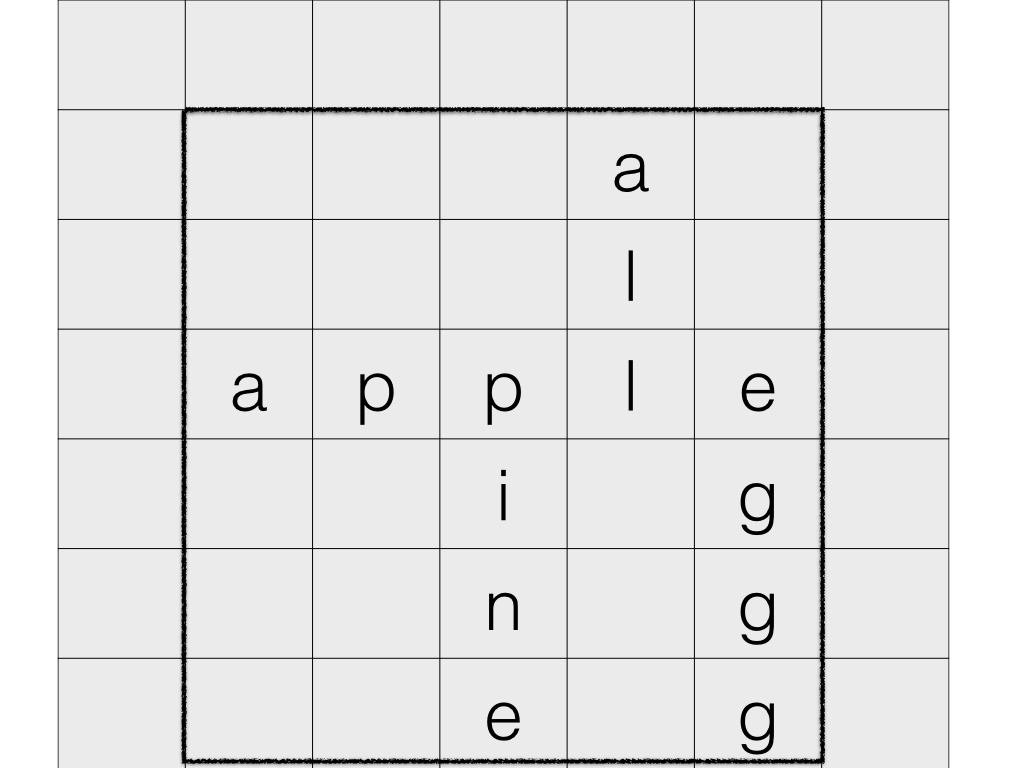
1. 填入词汇数：当队列为空时，地图生成算法便结束。故在一次生成过程中，可能由于某些步骤较为糟糕，（例如选择填入的词较长，占据了大量空间且没办法与剩下的词交汇，使得剩余的词汇不能填入到该区域）导致并不一定每个词都能填入到地图中。每填入一个单词，都会增加地图的复杂度，故填入词汇数是优化的重要因素。

2. 交汇点数：生成算法的设计是的每个填入的词至少会与另外一个词交汇。即若填入n个词，则交汇点数m满足：

交汇点上的字母需要满足横纵方向单词的正确性，那么当交汇点数越多，玩家需要顾及到的单词交汇情况则越多，游戏的可玩性才越强。因此地图优化应试图最大化k值。

3. 地图有效矩形面积：

地图有效矩形面积即能将所有填入的单词都包含进去的最小矩形面积。如下图所示，图中粗线黑框则为地图有效矩阵。



当填入的词汇数与交汇点数都一致时，若地图的有效矩形越小，表示地图越“密集”，在视觉上给予玩家的干扰效果也越好。因此地图有效矩形面积是次于交汇点数以及填入单词数的因素。

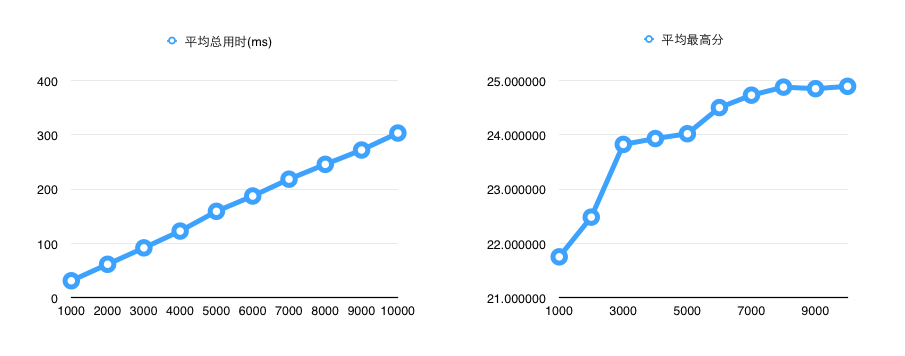
公式：

设填入单词数为n，形成的交汇点数为m，有效矩阵面积为S，拼字地图边长为w，则地图质量计算公式：

由上可见，单词数与交汇点的权重比为1:2，而由于必有,故，即有效矩阵因素只会在前两项均相等时才起作用。

地图生成次数：

为了选择比较适合的重复次数，我选择使用一篇较为典型的英语文章作为例子（文章见附录A）。该文章为一篇GRE写作范文，其难度与从网络中下载的新闻、学术类文章难度相近，具有代表性。经过对文章单词的解析和筛选，最后选出的15个单词分别为：congruity、convivial、impressing、sketching、enmity、stressful、contradicts、aggregates、sighted、logically、divergence、temper、quarrel、disparate和contradict。下图为通过使用Java的系统函数统计的不同的算法重复次数下总用时以及最高分。（实验数据收集时重复次数皆为100次，取平均数）

由左图可知，总用时与地图生成算法重复次数基本呈现线性正相关。而当重复次数达到8000次时，最高分已基本趋于稳定。因此地图生成算法的优化重复次数设置为8000次，用时平均为245ms。

1. (\* 注释，为你的某段文字提供解释 \*) [↑](#footnote-ref-1)
2. (\* 注释，为你的某段文字提供解释 \*) [↑](#footnote-ref-2)
3. (\* 注释，为你的某段文字提供解释 \*) [↑](#footnote-ref-3)