

分类号: _____

单位代码: _____ 10033

密 级: _____

学 号: _____ 114514

中国传媒大学

硕士学位论文



中文论文题目: 基于想不出名字所以随便装作轻小说标题的研究

英文论文题目: Research and Implementation of one Failed Love
Comedy based on My Poor Youth

申请人姓名: _____ 云游君

指导教师: _____ 比企谷八幡

专业名称: _____ 计算机应用技术

研究方向: _____ 数字娱乐与动画技术

所在学院: _____ 计算机与网络空间安全学院

论文提交日期 2022 年 2 月 21 日

中国传媒大学研究生学位论文独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 中国传媒大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名: 签字日期: 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解 中国传媒大学 有权保留并向国家有关部门或机构送交本论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 中国传媒大学 可以将学位论文的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后适用本授权书)

学位论文作者签名: 导师签名:

签字日期: 年 月 日 签字日期: 年 月 日

基于想不出名字所以随便装作轻小说标题的研究

摘 要

论文的摘要是对论文研究内容和成果的高度概括。摘要应对论文所研究的问题及其研究目的进行描述，对研究方法和过程进行简单介绍，对研究成果和所得结论进行概括。摘要应具有独立性和自明性，其内容应包含与论文全文同等量的主要信息。使读者即使不阅读全文，通过摘要就能了解论文的总体内容和主要成果。

论文摘要的书写应力求精确、简明。切忌写成对论文书写内容进行提要的形式，尤其要避免“第 1 章……；第 2 章……；……”这种或类似的陈述方式。

关键词是为了文献标引工作、用以表示全文主要内容信息的单词或术语。关键词不超过 5 个，每个关键词中间用分号分隔。

关键词：关键词 1；关键词 2；关键词 3；关键词 4；关键词 5

Research and Implementation of one Failed Love Comedy based on My Poor Youth

ABSTRACT

An abstract of a dissertation is a summary and extraction of research work and contributions. Included in an abstract should be description of research topic and research objective, brief introduction to methodology and research process, and summary of conclusion and contributions of the research. An abstract should be characterized by independence and clarity and carry identical information with the dissertation. It should be such that the general idea and major contributions of the dissertation are conveyed without reading the dissertation.

An abstract should be concise and to the point. It is a misunderstanding to make an abstract an outline of the dissertation and words “the first chapter”, “the second chapter” and the like should be avoided in the abstract.

Keywords are terms used in a dissertation for indexing, reflecting core information of the dissertation. An abstract may contain a maximum of 5 keywords, with semi-colons used in between to separate one another.

有时候英文摘要会超过一页，分两页也是可以的（我看师兄是这样）

KEYWORDS: keyword 1; keyword 2; keyword 3; keyword 4; keyword 5

目 录

摘 要	I
ABSTRACT	II
目 录	III
插图和附表清单	VI
1 绪论	1
1.1 研究背景及意义	1
1.1.1 研究背景	1
1.1.2 研究意义	2
1.2 国内外研究现状	2
1.2.1 论文主要部分的写法	2
1.2.2 论文的语言及表述	3
1.2.3 论文题目的写法	3
1.3 本文研究内容	3
1.4 本文组织结构	4
2 相关技术概述	5
2.1 西瓜种植	5
2.1.1 西瓜种子的最大公约数	5
2.2 鸡鸭养殖	5
2.2.1 循环生态的应用	5
2.2.2 确定有限状态机	5
2.3 虚拟世界	6
2.3.1 坐标系	6
2.4 数学公式	7
2.4.1 四元数的定义	7
2.4.2 四元数的几何意义	7

2.4.3 四元数的其他优势.....	8
2.4.4 贝塞尔曲线.....	9
2.5 本章小结	11
3 西瓜种植	12
3.1 传统种植方案	12
3.1.1 西瓜种植对比实验.....	12
3.2 西红柿炒鸡蛋	13
3.3 一些语法帮助	14
3.3.1 枚举示例.....	14
3.3.2 实验设计.....	14
3.3.3 实验结果及分析.....	15
3.4 本章小结	15
4 鸡鸭养殖	16
4.1 画图示例	16
4.1.1 公式示例.....	17
4.2 养殖算法	17
4.2.1 为什么呢.....	17
4.3 本章小结	19
5 系统设计与实现.....	20
5.1 设计架构	20
5.2 客户端开发	20
5.3 系统实现	20
5.4 系统展示	20
5.5 系统测试与评价	21
5.6 本章小结	21
6 总结与展望.....	23
6.1 总结	23
6.2 展望	23

参考文献.....	24
附 录	25
符号和缩略语说明.....	27
致 谢	28

插图和附表清单

图 1.1	2015-2024 年全球玩家数 ^[2]	1
图 2.1	FSM 逻辑示意图	6
图 2.2	FSM: 检测二进制数是否含有偶数个零	6
图 2.3	左手坐标系	7
图 2.4	右手坐标系	7
图 2.5	复数的几何意义	7
图 2.6	线性插值示意图	9
图 2.7	三次贝塞尔曲线	10
图 2.8	De Casteljau 算法绘制过程	10
图 3.1	JSON VS XML 读取解析性能对比	13
图 3.2	节点示意图	14
图 3.3	多节点语法树示意图	15
图 4.1	正向动力学	16
图 4.2	八叉树示例	18
图 4.3	小云立绘	19
图 5.1	Monorepo 中 Yarn/pnpm/npm Benchmark	20
图 5.2	模块关系图	21
图 5.3	测试	22
表 2.1	状态转移表	6
表 3.1	JSON VS XML 读取解析性能 (Chrome V8)	13
表 4.1	JsPerf (Chrome 84) 中循环性能对比	19

1 绪论

我习惯在每章节加一段引入的话，譬如面对 XXX 带来的挑战，本文提出了一种 XXX 方法。本章节就其研究背景、意义、现状以及本文研究内容与组织结构进行简要介绍。

1.1 研究背景及意义

1.1.1 研究背景

一堆乱七八糟的研究背景，加一点宏大叙事。简称懂的都懂。

中国互联网络信息中心（CNNIC）2021 年 8 月 27 日在京发布的第 48 次《中国互联网络发展状况统计报告》显示，截至 2021 年 6 月，我国网民规模达 10.11 亿，互联网普及率达 71.6%。十亿用户接入互联网，形成了全球最为庞大、生机勃勃的数字社会。^[1]其中我国网络游戏用户规模达 5.09 亿，占网民整体的 50.4%。

游戏产业的不断发展，使得游戏的经济属性和文化价值显现得愈加明显，越来越多的人接受了这种集休闲娱乐与文化传播为一体的新兴媒介。据 Newzoo 2021 全球游戏市场报告^[2]表明（如图 1.1），2021 年全球游戏市场收获近 1758 亿美元，全球游戏玩家数



图 1.1 2015-2024 年全球玩家数^[2]

量将持续增长，到 2023 年将超过三十亿，市场收入也在随之不断增长。全球三十亿玩

家中移动玩家数量约为 28 亿，PC 及主机玩家分别为 14 亿与 9 亿。

...

1.1.2 研究意义

写写研究意义。

关于这个事，我简单说两句，至于我的身份，你明白就行。总而言之，这个事呢，现在就是这个情况，具体的呢，大家也都看得到，我因为这个身份上的问题，也得出来说那么几句，可能，你听的不是很明白，但是意思就是那么个意思。我的身份呢，不知道的你也不用去猜，这种事情见得多了，我只想说懂得都懂，不懂的我也不多解释，毕竟自己知道就好，细细品吧。你们也别来问我怎么了，利益牵扯太大，说了对你我都没好处，当不知道就行了，其余的我只能说这里面水很深，牵扯到很多东西。详细情况你们自己是很难找的，网上大部分已经删除干净了，所以我只能说懂得都懂。懂的人已经基本都获利上岸什么的了，不懂的人永远不懂，关键懂的人都是自己悟的，你也不知道谁是懂的人也没法请教，大家都藏着掖着生怕别人知道自己懂事，懂了就能收割不懂的，你甚至都不知道自己不懂。只是在有些时候，某些人对某些事情不懂装懂，还以为别人不懂。其实自己才是不懂的，别人懂的够多了，不仅懂，还懂的超越了这个范围，但是某些不懂的人让这个懂的人完全教不懂，所以不懂的人永远不懂，只能不懂装懂。别人说懂的都懂，只要点点头就行了，其实你懂的我也懂，谁让我们都懂呢，不懂的话也没必要装懂，毕竟里面牵扣扯到很多懂不了的事。这种事懂的人也没必要访出来，不懂的人看见又来问七问八，最后跟他说了他也不一定能懂，就算懂了以后也对他不好，毕竟懂的太多了不是好事。所以大家最好是不懂就不要去了解，懂太多不好。

1.2 国内外研究现状

可以分几节来写。对了一般句子末尾最好不要单独一个字另起一行。

1.2.1 论文主要部分的写法

研究生学位论文撰写，除表达形式上需要符合一定的格式要求外，内容方面上也要遵循一些共性原则。

通常研究生学位论文只能有一个主题（不能是几块工作拼凑在一起），该主题应针

对某学科领域中的一个具体问题展开深入、系统的研究，并得出有价值的研究结论。学位论文的研究主题切忌过大，例如，“中国国有企业改制问题研究”这样的研究主题过大，因为“国企改革”涉及的问题范围太广，很难在一本研究生学位论文中完全研究透彻。

1.2.2 论文的语言及表述

除国际研究生外，学位论文一律须用汉语书写。学位论文应当用规范汉字进行撰写，除古汉语研究中涉及的古文字和参考文献中引用的外文文献之外，均采用简体汉字撰写。

国际研究生一般应以中文或英文书写学位论文，格式要求同上。论文须用中文封面。

研究生学位论文是学术作品，因此其表述要严谨简明，重点突出，专业常识应简写或不写，做到立论正确、数据可靠、说明透彻、推理严谨、文字凝练、层次分明，避免使用文学性质的或带感情色彩的非学术性语言。

论文中如出现一个非通用性的新名词、新术语或新概念，需随即解释清楚。

1.2.3 论文题目的写法

论文题目应简明扼要地反映论文工作的主要内容，力求精炼、准确，切忌笼统。论文题目是对研究对象的准确、具体描述，一般要在一定程度上体现研究结论，因此，论文题目不仅应告诉读者这本论文研究了什么问题，更要告诉读者这个研究得出的结论。例如：“在事实与虚构之间：梅乐、卡彭特、沃尔夫的新闻观”就比“三个美国作家的新闻观研究”更专业、更准确。

1.3 本文研究内容

总结本文研究内容。本文的主要目的是……

一篇学位论文的引言大致包含如下几个部分：1、问题的提出；2、选题背景及意义；3、文献综述；4、研究方法；5、论文结构安排。

- 问题的提出：要清晰地阐述所要研究的问题“是什么”。
- 选题背景及意义：论述清楚为什么选择这个题目来研究，即阐述该研究对学科发

展的贡献、对国计民生的理论与现实意义等。

- 文献综述：对本研究主题范围内的文献进行详尽的综合述评，“述”的同时一定要有“评”，指出现有研究状态，仍存在哪些尚待解决的问题，讲出自己的研究有哪些探索性内容。
- 研究方法：讲清论文所使用的学术研究方法。
- 论文结构安排：介绍本论文的写作结构安排。

1.4 本文组织结构

本文通过以下六个章节详细论述该研究课题，结构内容如下：

第一章：绪论。本章讨论了 XXX 以及它所代表的 XX 方式的意义与可在经济、社会价值上起到的作用，并回顾国内外现状，明确了本文研究内容与组织结构。

第二章：相关技术概述。本章对后续研究所要使用到的相关技术及概念进行了阐述。文中依次介绍了 XXX、XXX、XX 与 XX 等技术的基础原理与作用。...

第三章：XXX。...

第四章：XXX。...

第五章：系统设计与应用。本章从软件工程角度概述了系统的整体设计架构与设计方法，具体介绍了相关开发实现方案与特色功能，将三四章研究内容串联并应用以完成系统开发。最后对系统进行了展示与测试评估。

第六章：总结与展望。本文目标内容已基本顺利完成，但其研究成果仍然存在许多改进空间。该部分对研究内容与系统成品进行回顾，总结了主旨与存在的问题。通过思考相应改进措施，对未来研究内容与方向进行展望。

2 相关技术概述

...

本章主要对其中涉及的基本概念与相关技术进行介绍。

2.1 西瓜种植

... 虽然西瓜很好吃，但我不喜欢有很多种子的。

2.1.1 西瓜种子的最大公约数

所以只能边吃西瓜边数种子。展示一下可以写伪代码，说实话感觉格式优点麻烦。譬如辗转相除法伪代码示例如下：

算法 2.1 辗转相除法

```
while  $b \neq 0$  do
  if  $a \geq b$  then
     $a \leftarrow a - b$ 
  else
     $b \leftarrow b - a$ 
  end if
end while
```

2.2 鸡鸭养殖

鸡鸭适合养在树林里，这样既可以节约空间，鸡也可以找虫子吃，粪便还可作为肥料，旁边要是再有个小池塘就更好了。

2.2.1 循环生态的应用

FSM 逻辑如图 2.1 所示，它的输入和当前状态决定着下一个状态和输出。

2.2.2 确定有限状态机

通常使用没有起点的箭头表示开始状态，双圆圈表示接受状态。自动机中开始状态也可以是接受状态，此时表示其接受空字符串。

如图 2.2 所示，这是一台用于判断输入二进制字符串是否含有偶数个 0 的有限自动机的逻辑图。 S_1 为开始状态，并代表已经输入了偶数个 0，因此 S_1 也是接受状态。若

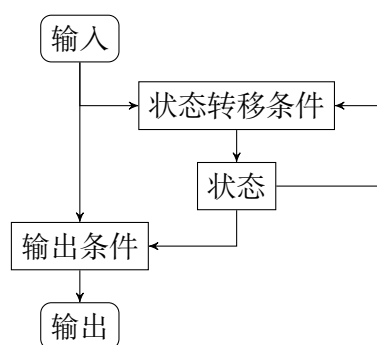


图 2.1 FSM 逻辑示意图

输入为不包含 0 或含有偶数个 0 的字符串（例如：1, 100, 01011 等），则自动机以接受状态结束。

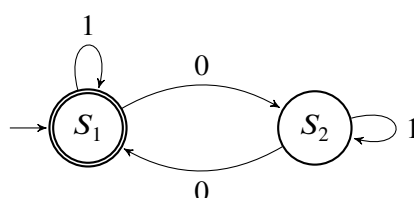


图 2.2 FSM：检测二进制数是否含有偶数个零

同时它也是一个确定有限状态机（DFA），对字母表中每个符号，确定有限自动机的状态都有且仅有一个转移。它对应的状态转移表如表 2.1 所示：

表 2.1 状态转移表

输入状态	1	0
S_1	S_1	S_2
S_2	S_2	S_1

2.3 虚拟世界

...

2.3.1 坐标系

还可以用 TIKZ 画坐标系。

目前工业界的各类软件采用坐标系也各不相同，譬如 Unreal、Unity、Babylon、DirectX 均采用左手坐标系（如图 2.3），而 Three.js、Cocos、OpenGL、Vulkan 等则采用右手坐标系（如图 2.4）。

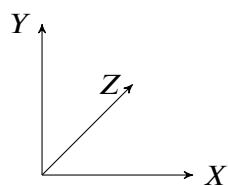


图 2.3 左手坐标系

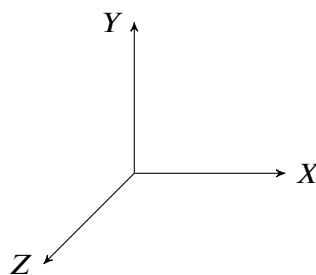


图 2.4 右手坐标系

2.4 数学公式

2.4.1 四元数的定义

譬如四元数：

$$i^2 = j^2 = k^2 = -1 \quad (2.1)$$

$$ij = -ji = k$$

$$ki = -ik = j \quad (2.2)$$

$$jk = -kj = i$$

2.4.2 四元数的几何意义

TIKZ 画坐标系。

在平面坐标系上，以横坐标 R 代表实部，纵坐标 I 代表虚部，复数的几何意义如图 2.5。

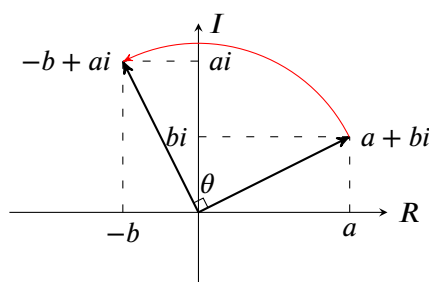


图 2.5 复数的几何意义

$(a + bi) * i = -b + ai$ ，复数乘以 i ，相当于将平面上复数做逆时针旋转 θ (90°)。

同理非零四元数通常可用于描述三维空间的拉伸与旋转变换，单位 i 、 j 、 k 的几何

意义则可以理解为三维空间的一种旋转。以右手坐标系（如图 2.4）为例，其中 i 代表绕 Z 轴（ X 正向向 Y 轴正向）旋转， j 代表绕 Y 轴（ Z 正向向 X 轴正向）旋转， k 代表绕 X 轴（ Y 正向向 Z 轴正向）旋转。

四元数模长定义如公式 (2.3)，当 $q = a + bi + cj + dk$ ， $\|q\| = 1$ 时，它所代表的变换并不会对原向量进行缩放，而是一个纯旋转。

$$\|q\| = a^2 + b^2 + c^2 + d^2 \quad (2.3)$$

此时 q 被称为单位四元数，也因此它通常被用于表示三维空间的旋转。同时四元数在静态坐标系中的代数计算，使其避免了动态欧拉角所造成的万向锁问题。

2.4.3 四元数的其他优势

四元数相比旋转矩阵与欧拉角，除了可以避免万向锁问题，还具有其他方面的优势。譬如速度更快、占用存储空间小、更易于平滑插值等。

3D 空间中任意一个旋转都能够使用三个四元数相乘的形式进行表达，即 $R_q(p) = qpq^{-1}$ ，其中 q 为一个单位四元数， p 为一个纯四元数。 R_q 表示一个可以将空间点 p 旋转到空间另一个点 $R_q(p)$ 的一个旋转。

定义一个四元数 $q = a + bi + cj + dk$ 的共轭为 $q^* = a - bi - cj - dk$ 。

设 $\mathbf{v} = bi + cj + dk$ ，则 $q = [s, \mathbf{v}]$ ， $q^* = [s, -\mathbf{v}]$ ，此时：

$$qq^* = s^2 + v^2 = \|q\|^2 \quad (2.4)$$

由于 $q^{-1} = \frac{q^*}{\|q\|^2}$ ，若使用单位四元数，则模长 ($\|q\|^2$) 为 1，故此时：

$$q^{-1} = \frac{q^*}{1} = q^* \quad (2.5)$$

存在 3D 旋转公式，任意向量 \mathbf{v} 沿着以单位向量定义的旋转轴 \mathbf{u} 旋转 θ 度之后的 \mathbf{v}' 可以使用四元数乘法获得。令 $v = [0, \mathbf{v}]$ ， $q = [\cos \frac{1}{2}\theta, \sin \frac{1}{2}\theta \mathbf{u}]$ ，则：

$$\mathbf{v}' = qvq^{-1} = qvq^* \quad (2.6)$$

同时存在与单位四元数旋转效果相同的 3D 旋转矩阵，即任意向量 \mathbf{v} 沿着以单位向量定义的旋转轴 \mathbf{u} 旋转 θ 角度之后的 \mathbf{v}' 可以使用矩阵乘法来获得。

令 $a = \cos \frac{1}{2}\theta$, $b = \sin \frac{1}{2}\theta u_x$, $c = \sin \frac{1}{2}\theta u_y$, $d = \sin \frac{1}{2}\theta u_z$, 那么:

$$\mathbf{v}' = \begin{bmatrix} 1 - 2c^2 - 2d^2 & 2bc - 2ad & 2ac + 2bd \\ 2bc + 2ad & 1 - 2b^2 - 2d^2 & 2cd - 2ab \\ 2bd - 2ac & 2ab + 2cd & 1 - 2b^2 - 2c^2 \end{bmatrix} \mathbf{v} \quad (2.7)$$

由上可知, 四元数相对于旋转矩阵, 使用 q^* 替换 q^{-1} , 大大提高了求四元数的逆的计算效率, 同时比旋转矩阵的计算更为快速方便。此外, 四元数对应实部与虚部 ijk 只需要存储四个数值, 空间占用也远小于同等效果的旋转矩阵。

2.4.3.1 绘制箭头示例

线性插值 (Lerp) 会沿一条直线进行插值, 如图 2.6 所示, v_t 在 v_0 与 v_1 组成的三角形斜边上滑动。

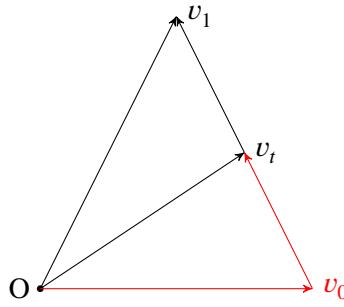


图 2.6 线性插值示意图

v_t 可写为 $v_0 + (v_1 - v_0)t$, 其中 t 是插值的参数, 取值范围为 $[0, 1]$ 。

$$v_t = \text{Lerp}(v_0, v_1, t) = v_0 + (v_1 - v_0)t = (1 - t)v_0 + tv_1 \quad (2.8)$$

将 Lerp 应用于单位四元数上, 可得:

$$q_t = \text{Lerp}(q_0, q_1, t) = (1 - t)q_0 + tq_1 \quad (2.9)$$

2.4.4 贝塞尔曲线

贝塞尔曲线最初由 Paul de Casteljau 使用 de Casteljau 算法开发, 是一种计算曲线的稳定方法。其根据控制点的数量可分为不同阶, 常见类型有线性贝塞尔曲线、二次贝塞尔曲线 (也曾被用于 TrueType 的字体平滑)、三次贝塞尔曲线等。

线性贝塞尔曲线等同于线性插值, 给定两个控制点 P_0 与 P_1 , 则公式为:

$$\mathbf{B}(t) = \mathbf{P}_0 + (\mathbf{P}_1 - \mathbf{P}_0)t = (1 - t)\mathbf{P}_0 + t\mathbf{P}_1, t \in [0, 1] \quad (2.10)$$

二次贝塞尔曲线给定三个点 P_1 、 P_2 与 P_3 ，公式为：

$$\mathbf{B}(t) = (1-t)^2\mathbf{P}_0 + 2t(1-t)\mathbf{P}_1 + t^2\mathbf{P}_2, t \in [0, 1] \quad (2.11)$$

三次贝塞尔曲线如图 2.7 所示，给定四个点 P_1 、 P_2 、 P_3 与 P_4 ，公式为：

$$\mathbf{B}(t) = \mathbf{P}_0(1-t)^3 + 3\mathbf{P}_1t(1-t)^2 + 3\mathbf{P}_2t^2(1-t) + \mathbf{P}_3t^3, t \in [0, 1] \quad (2.12)$$

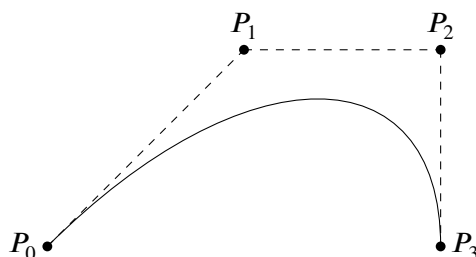


图 2.7 三次贝塞尔曲线

n 阶贝塞尔曲线给定 P_0 、 P_1 、 \dots 、 P_n 等 $n+1$ 个点，公式可递归表达为：

$$\mathbf{B}(t) = \mathbf{B}_{P_0P_1\dots P_n}(t) = (1-t)\mathbf{B}_{P_0P_1\dots P_{n-1}}(t) + t\mathbf{B}_{P_1P_2\dots P_n}(t) \quad (2.13)$$

De Casteljau 算法^[3]即基于此公式，采用递归方式来计算一组控制点所构建的贝塞尔曲线所经过的点。

给定 P_0 到 P_3 的控制点，将其分解为 $L_0L_1L_2L_3$ 与 $R_0R_1R_2R_3$ 继续绘制，并可通过级别控制绘制的精细度，关键点如图 2.8 所示：

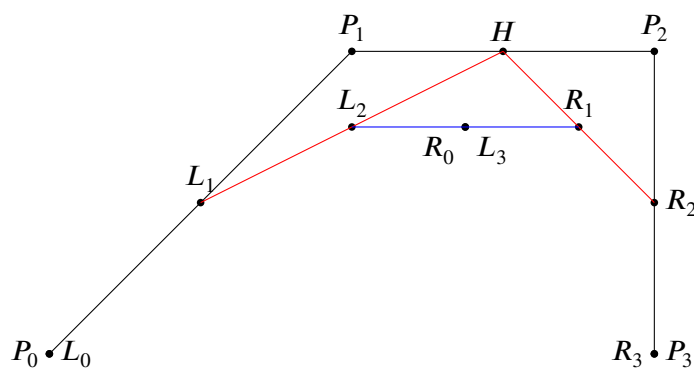


图 2.8 De Casteljau 算法绘制过程

绘制 P_0 到 P_3 曲线的代码示例如下所示：

```
1 % 返回这两点间线段的中点
2 def midpoint((x1, y1), (x2, y2)):
3     return ((x1+x2)/2, (y1+y2)/2)
4
5 % 绘制级别，可决定精细度
6 MAX_LEVEL = 5
```

```
7 def draw_curve(P0, P1, P2, P3, level=1):
8     % 当级别达到最大时, 则绘制两点之间直线
9     if level == MAX_LEVEL:
10         d.line((P0, P3))
11     else:
12         L0 = P0
13         L1 = midpoint(P0, P1)
14         H = midpoint(P1, P2)
15         R2 = midpoint(P2, P3)
16         R3 = P3
17         L2 = midpoint(L1, H)
18         R1 = midpoint(R2, H)
19         L3 = midpoint(L2, R1)
20         R0 = L3
21         % 绘制左半段贝塞尔曲线 L0-L1-L2-L3
22         draw_curve(L0, L1, L2, L3, level+1)
23         % 绘制右半段贝塞尔曲线 R0-R1-R2-R3
24         draw_curve(R0, R1, R2, R3, level+1)
25
26 % 根据输入的点绘制曲线
27 draw_curve((10,10), (100,100), (100,10), (100,100))
```

2.5 本章小结

本章在第一节中介绍了... 在第二节中介绍了...

3 西瓜种植

本章节设计了一种西瓜种植方案... 旨在简化种植的工作流程，同时...

3.1 传统种植方案

...

3.1.1 西瓜种植对比实验

...

3.1.1.1 实验设计

JSON 代码插入示例：

```
1 {  
2   "scene": {  
3     "characters": [  
4       {  
5         "avatar": "http://placeimg.com/640/480",  
6         "name": "Jonathan Mitchell Jr."  
7       },  
8       {  
9         "avatar": "http://placeimg.com/640/480",  
10        "name": "Jerry Torp Sr."  
11      }  
12    ],  
13    "dialog": {  
14      "text": "Quia quia eos ut assumenda quos. Est voluptate  
↪ officia consequatur sint assumenda natus. ..."  
15    }  
16  }  
17 }
```

XML 代码插入示例：

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
2 <scene>  
3   <characters>  
4     <avatar>http://placeimg.com/640/480</avatar>  
5     <name>Jonathan Mitchell Jr.</name>  
6   </characters>  
7   <characters>  
8     <avatar>http://placeimg.com/640/480</avatar>  
9     <name>Jerry Torp Sr.</name>  
10  </characters>
```

```
11     <dialog>
12       <text>Quia quia eos ut assumenda quos. Est voluptate officia
          consequatur sint assumenda natus. ...</text>
13       <name>Jonathan Mitchell Jr.</name>
14     </dialog>
15   </scene>

...
```

3.1.1.2 实验结果及分析

表格示例：

表 3.1 JSON VS XML 读取解析性能（Chrome V8）

方法	每秒操作次数	误差率	相对落后比例
JSON.parse	28 ops/s	±4.36%	0%
xml2js.parseString	12 ops/s	±4.16%	65.52%
xml2js.parseStringPromise	10 ops/s	±6.23%	68.97%

柱状图示例：

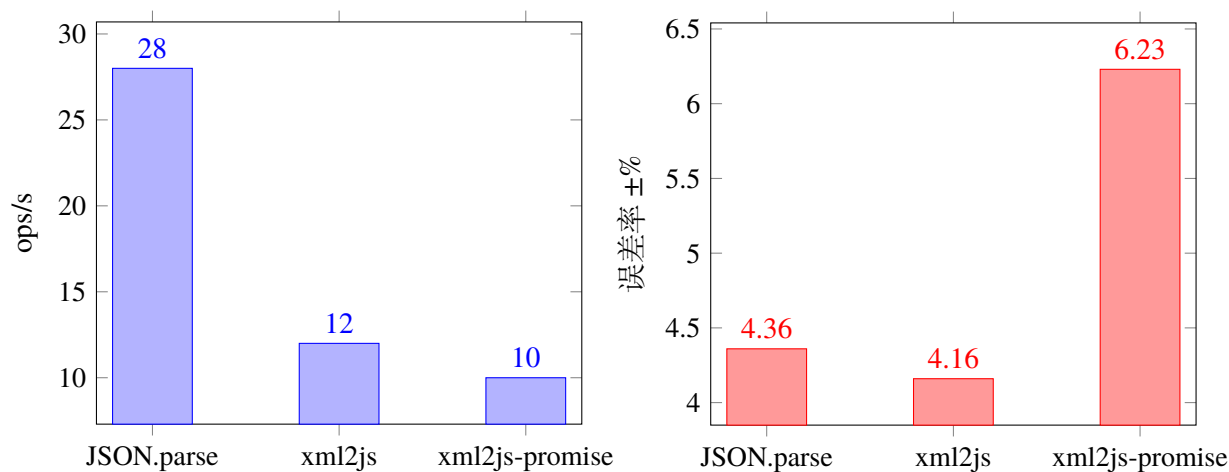


图 3.1 JSON VS XML 读取解析性能对比

引用可以使用 \ref 的方式。

写成如表 3.1，将会自动编号。

3.2 西红柿炒鸡蛋

普通文本代码示例：

1 好久不见，你去哪儿啦？
 2
 3 江湖。
 4
 5 江湖在什么地方？
 6
 7 我指给你看 … … … … …

3.3 一些语法帮助

3.3.1 枚举示例

有序号枚举示例：

1. 新年快乐
2. 万事如意
3. 心想事成

没序号的：

- 新年快乐
- 万事如意
- 心想事成

\backslash 表示反斜杠，\。

1 > 好久不见，你去哪儿啦？
 2 >
 3 > 江湖。
 4 >
 5 > 江湖在什么地方？
 6 >
 7 > 我指给你看 … … … … …

树节点画法：



图 3.2 节点示意图

3.3.2 实验设计

...

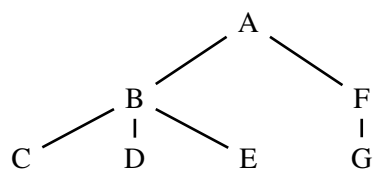


图 3.3 多节点语法树示意图

3.3.3 实验结果及分析

。。。

3.4 本章小结

本章第一节回顾了...

发现还是直接买最方便。

4 鸡鸭养殖

...

4.1 画图示例

TIKZ 画线:

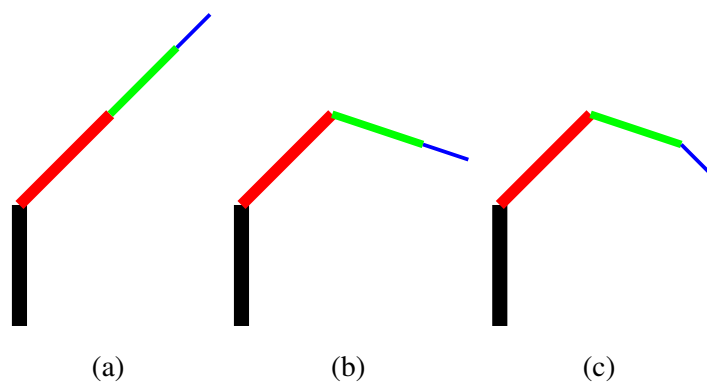


图 4.1 正向动力学

JavaScript 代码高亮示例:

```

1 function setup() { // 初始化骨骼结构
2   let current = root;
3   for(const bone in bones) { // 遍历骨骼 (可能存在多个顺序)
4     const next = bone.mesh; current.child = next; current = next;
5   }
6 }
7 function draw { // 更新绘制
8   while(next) {
9     next.update(); next.show(); next = next.child;
10  }
11 }
12 class Segment { constructor(point, length, angle) {...} update() { /* 根据
    父节点更新位置 */ } }
```


4.1.1 公式示例

欧拉角转四元数公式：

$$\begin{aligned}
 \mathbf{q} &= \mathbf{q}_z \mathbf{q}_x \mathbf{q}_y \\
 &= \begin{bmatrix} \cos(\frac{y}{2}) \\ 0 \\ 0 \\ \sin(\frac{y}{2}) \end{bmatrix} \begin{bmatrix} \cos(\frac{p}{2}) \\ \sin(\frac{p}{2}) \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \cos(\frac{r}{2}) \\ 0 \\ \sin(\frac{r}{2}) \\ 0 \end{bmatrix} \\
 &= \begin{bmatrix} \cos(\frac{p}{2}) \cos(\frac{r}{2}) \cos(\frac{y}{2}) - \sin(\frac{p}{2}) \sin(\frac{r}{2}) \sin(\frac{y}{2}) \\ \cos(\frac{p}{2}) \cos(\frac{r}{2}) \sin(\frac{y}{2}) - \cos(\frac{p}{2}) \sin(\frac{r}{2}) \sin(\frac{y}{2}) \\ \cos(\frac{p}{2}) \cos(\frac{r}{2}) \sin(\frac{y}{2}) + \cos(\frac{p}{2}) \sin(\frac{r}{2}) \sin(\frac{y}{2}) \\ \cos(\frac{p}{2}) \cos(\frac{r}{2}) \sin(\frac{y}{2}) + \cos(\frac{p}{2}) \sin(\frac{r}{2}) \sin(\frac{y}{2}) \end{bmatrix}
 \end{aligned} \tag{4.1}$$

四元数转欧拉角时，设 $\mathbf{q} = [q_0 \ q_1 \ q_2 \ q_3]^T = [q_w \ q_x \ q_y \ q_z]^T$ ，而求解欧拉角时计算机中的 \arctan 与 \arcsin 只能获得区间 $(-\frac{\pi}{2}, \frac{\pi}{2})$ 的结果，因此使用 asin 替换 \arcsin ， atan2 替换 \arctan 函数^[4]，则：

$$\begin{bmatrix} \text{pitch} \\ \text{roll} \\ \text{yaw} \end{bmatrix} = \begin{bmatrix} \text{asin}(-2(q_z q_y - q_x q_w)) \\ \text{atan2}(2(q_z q_x + q_y q_w), q_z^2 - q_x^2 - q_y^2 + q_w^2) \\ \text{atan2}(2(q_x q_y + q_z q_w), -q_z^2 - q_x^2 + q_y^2 + q_w^2) \end{bmatrix} \tag{4.2}$$

4.1.1.1 插值公式

$$y = y_0 + (x - x_0) \frac{y_1 - y_0}{x_1 - x_0} = \frac{y_0(x_1 - x) + y_1(x - x_0)}{x_1 - x_0} \tag{4.3}$$

4.2 养殖算法

4.2.1 为什么呢

画立方体：

放一些看起来高大上的公式。

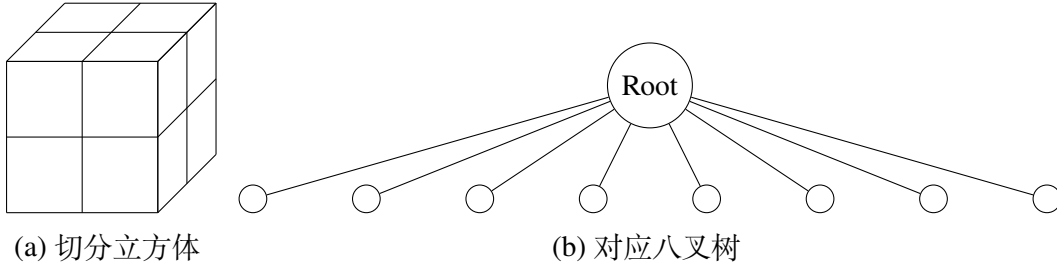


图 4.2 八叉树示例

K-Means 的主要思想是将已知观测集 (x_1, x_2, \dots, x_n) 划分到 k 个集合 ($k \leq n$)，并使每个集合内平方和最小。即寻找使得下式 (4.4) 最小的聚类 S_i (μ_i 是 S_i 中所有点的均值)。

$$\arg \min_{S_i} \sum_{i=1}^k \sum_{x_i \in S_i} (x_i - \mu_i)^2 \quad (4.4)$$

算法实现可主要分为以下两步：分配与更新。

譬如设定存在 k 个初始均值点为 m_1, \dots, m_k 。

分配：将每个点分配到聚类中，使得集合中的平方和 (WCSS) (即平方后的欧式距离) 最小。即满足公式 (4.5)：

$$S_i^{(t)} = \left\{ x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2 \forall j, 1 \leq j \leq k \right\} \quad (4.5)$$

更新：对于前一步骤得到的每一聚类，重新计算聚类中心 (如公式 (4.6))，作为新的均值点。

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \quad (4.6)$$

C 代码格式示例：

```

1  register short *to, *from;
2  register count;
3  {
4      register n = count % 8;
5      while (n-- > 0) {
6          *to = *from++;
7      }
8      n = count / 8;
9      if (n == 0) return;
10     do {
11         *to = *from++; *to = *from++; *to = *from++; *to = *from++;
12         *to = *from++; *to = *from++; *to = *from++; *to = *from++;
13     } while (--n > 0)
14 }
```

三线表：

表 4.1 JsPerf （Chrome 84） 中循环性能对比

方法	Opts/sec
普通循环	165,877
Duff's Device	396,801

4.2.1.1 示例图片

...
多图并列：

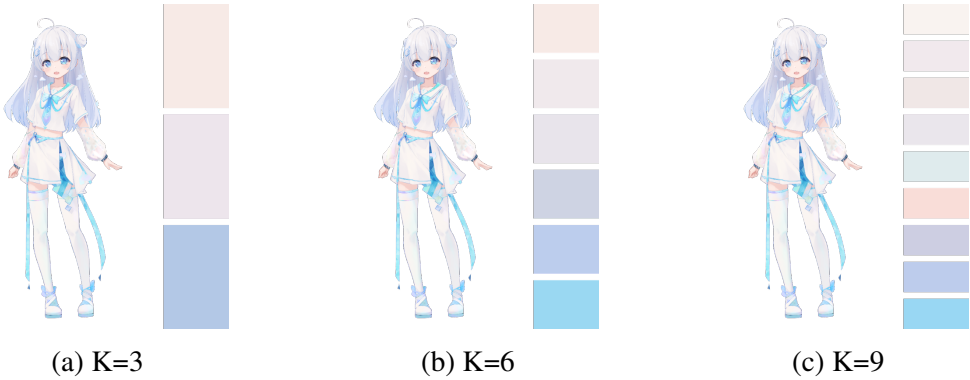


图 4.3 小云立绘

`\href{https://yunyoujun.cn/}{云游君的小站}` 可生成链接云游君的小站。

4.3 本章小结

本章主要介绍...

5 系统设计与实现

本文第三四章中详细介绍了..., 这为最终系统和我财富自由的实现提供了技术基础。

...

本章将负责介绍系统的设计与实现, 同时这也是本文的最终目的。

5.1 设计架构

pnpm 管理依赖的性能相较 npm、yarn 提升显著, 如图 5.1。

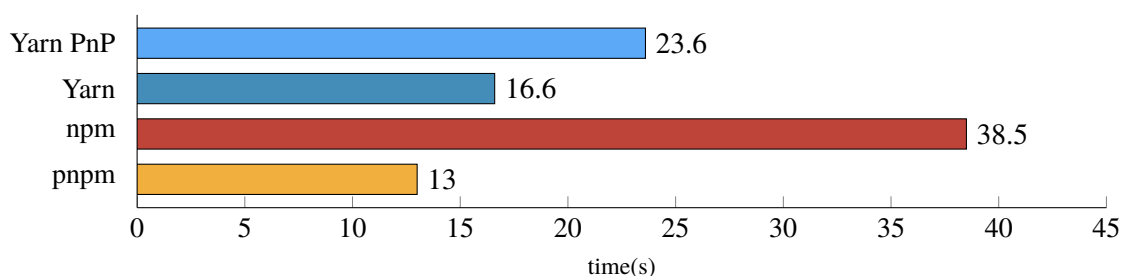


图 5.1 Monorepo 中 Yarn/pnpm/npm Benchmark

TIKZ 也可以画思维导图, 如图 5.2。

流程图可以看这里。

[https://www.overleaf.com/learn/latex/LaTeX_Graphics_using_TikZ%3A_A_Tutorial_for_Beginners_\(Part_3\)%E2%80%94Creating_Flowcharts](https://www.overleaf.com/learn/latex/LaTeX_Graphics_using_TikZ%3A_A_Tutorial_for_Beginners_(Part_3)%E2%80%94Creating_Flowcharts)

5.2 客户端开发

...

5.3 系统实现

各几节介绍一下系统实现的关键技术吧

5.4 系统展示

可以贴点图什么的……

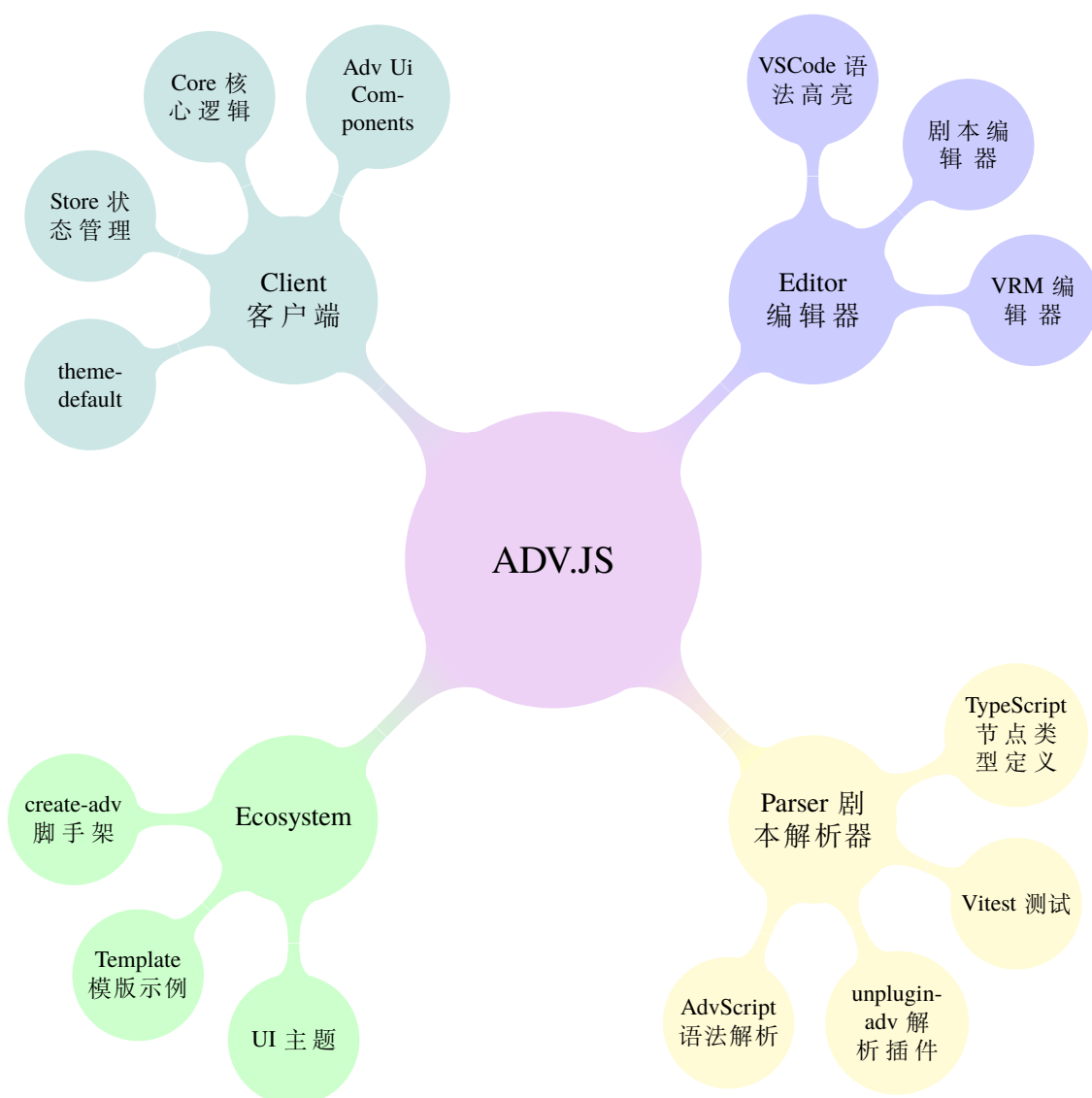


图 5.2 模块关系图

5.5 系统测试与评价

测试、调查

5.6 本章小结

本章...

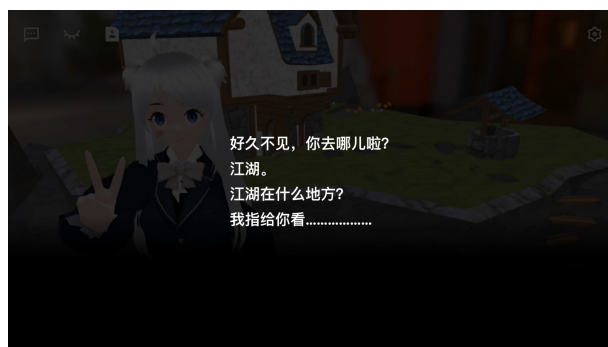


图 5.3 测试

6 总结与展望

6.1 总结

本文详细介绍了在... 中所做的工作，...

6.2 展望

找不足、展望！

参考文献

- [1] CNNIC. 第 48 次中国互联网络发展状况统计报告[R/OL]. 北京: 中国互联网络信息中心, 2021. <https://news.cctv.com/2021/08/27/ARTIAQ8bIAmQ68Vs88OMHnRa210827.shtml>.
- [2] Newzoo. 2021 全球游戏市场报告[R]. Amsterdam: Newzoo, 2021.
- [3] Farin G, Hansford D. The essentials of cagd[M]. CRC Press, 2000.
- [4] Babylon.js. src/math/math.vector.ts | babylon.js[EB/OL]. 2022-02-04. [https://github.com/BabylonJS/BabylonJS/blob/master/src/Maths/math.vector.ts#L3631-L3633](https://github.com/BabylonJS/Babylon.js/blob/master/src/Maths/math.vector.ts#L3631-L3633).

附录

表 A.1 3D 模型格式及作用

格式	描述	优缺点
.abc	特效工作室主导的通用格式	功能强大、通用，但体积大
.glTF	面向 Web 端的格式	体积小、跨平台，但功能相对有限
.fbx	Autodesk 家族通用格式	支持动画、兼容广泛，体积与功能介于 abc 与 glTF 之间，通常与 Autodesk 强绑定
.bvh	BioVision 等设备人物动作捕捉数据	通用的人体特征动画文件格式
.dae	fbx 替代品，基于 XML（Google 地图采用该格式）	通常需要额外下载软件转换，不够通用
.stl	三维打印的通用格式	格式简单，只能描述三维物体的几何信息，不支持颜色材质等信息
.3ds	三角面静态模型	格式简单，难以二次编辑，逐渐被淘汰
.obj	静态多边形模型，标准 3D 模型文件格式	通用，附带 UV 信息与材质路径，但不包含动画、粒子等信息
.ply	静态多边形模型（obj 的升级版），常见于点云扫描	改进了 obj 格式所缺少的对任意属性及群组的扩充性
.psk	Unreal Engine 独有	模型带有骨骼动画
.x3d	基于 XML 的 Web3D 标记语言，ISO 认证的国际标准	VRML 标准的升级版本，但制作工具和开发环境相对落后，占有率不高
.dxf	CAD 通用格式	矢量作图通用标准

表 A.2 人体模型骨骼索引

骨骼索引名	描述	骨骼索引名	描述
hips	臀部	leftThumbProximal	左拇指根部
leftUpperLeg	左大腿	leftThumbIntermediate	左拇指中部
rightUpperLeg	右大腿	leftThumbDistal	左拇指端部
leftLowerLeg	左小腿	leftIndexProximal	左食指根部
rightLowerLeg	右小腿	leftIndexIntermediate	左食指中部
leftFoot	左脚	leftIndexDistal	左食指端部
rightFoot	右脚	leftMiddleProximal	左中指根部
spine	脊柱	leftMiddleIntermediate	左中指中部
chest	胸部	leftMiddleDistal	左中指端部
neck	脖子	leftRingProximal	左无名指根部
head	头部	leftRingIntermediate	左无名指中部
leftShoulder	左肩	leftRingDistal	左无名指端部
rightShoulder	右肩	leftLittleProximal	左小指根部
leftUpperArm	左大臂	leftLittleIntermediate	左小指中部
rightUpperArm	右大臂	leftLittleDistal	左小指端部
leftLowerArm	左小臂	rightThumbProximal	右拇指根部
rightLowerArm	右小臂	rightThumbIntermediate	右拇指中部
leftHand	左手	rightThumbDistal	右拇指端部
rightHand	右手	rightIndexProximal	右食指根部
leftToes	左脚趾	rightIndexIntermediate	右食指中部
rightToes	右脚趾	rightIndexDistal	右食指端部
leftEye	左眼	rightMiddleProximal	右中指根部
rightEye	右眼	rightMiddleIntermediate	右中指中部
jaw	下颌	rightMiddleDistal	右中指端部
lastBone	最后一块骨骼索引分隔符	rightRingProximal	右无名指根部
upperChest	上胸膛	rightRingIntermediate	右无名指中部
rightRingDistal	右无名指端部	rightLittleIntermediate	右小指中部
rightLittleProximal	右小指根部	rightLittleDistal	右小指端部

符号和缩略语说明

AVG/ADV	文字冒险游戏 (Adventure Game)
AST	抽象语法树 (Abstract Syntax Tree)
FSM	有限状态机 (Finite-state Machine)
DFA	确定有限状态机 (Deterministic Finite Automaton)
BFS	广度优先搜索 (Breadth-First Search)
DFS	深度优先搜索 (Depth-First-Search)
PGC	专业生产内容 (Professionally-generated Content)
UGC	用户生成内容 (User-generated Content)
VRM	一种可用于虚拟现实场景的 3D 模型格式 (Virtual Reality Model)
PWA	渐进式 Web 应用 (Progressive Web App)
Lerp	线性插值 (Linear Interpolation)
Nlerp	正规化线性插值 (Normalized Linear Interpolation)
Slerp	球面线性插值 (Spherical Linear Interpolation)
SPA	单页应用 (Single Page Applications)
SSR	服务端渲染 (Server-Side Rendering)
SSG	静态站点生成 (Static Site Generator)
FK	正向动力学 (Forward Kinematics)
IK	反向动力学 (Inverse Kinematics)
CCDIK	循环坐标下降反向动力学 (Cyclic Coordinate Descent IK)
FABRIK	前后延伸反向动力学 (Forward And Backward Reaching IK)

致 谢

「反正你还年轻，人生有无限可能。」似乎的确经常从这里或那里看到或听到这样的话。但我已诞生于此世间足足有四分之一个世纪，同样也几乎走完了人生道路中最重要的四分之一。时至今日，原本某些柔软的东西也像胡茬一样开始变得硬邦邦的，再也无法逆转。

自三年前进入中国传媒大学读研以来，我的想法与人生目标已发生了许多变化。我的状态也如同状态机一般不断地从一种状态过渡到另一种状态，可惜其并非确定有限型的。我的初态与自己的期望相去甚远，但在此期间，却很庆幸地遇到了许多无法预测的可爱的人和事，因此，我决定把它叫做非确定无限状态机。

「假如当时选择了别的道路，会过着不同的大学生活吗？」我不止一次又一次地询问自己，并幻想平行世界中自己的现状。人生就像是玩文字冒险游戏，做着一个个的选择，最后迎来自己所决定的结局。但与确定有限状态机「确定输入」返回「确定结果」不同，即便同样的初态与输入，时机不同，我也未必拥有相同的结果。继续倒退，在灰暗的中学时代，我也曾幻想在进入大学后过上如「四叠半神话」中所言的玫瑰色校园生活，与明石相遇、在社团挥洒汗水、精进学业、锻炼身体，未来成为社会栋梁。但世事难料，我机缘巧合地到了我的母校就读计算机专业，并拖着步伐踩着绊脚石把时光花费在每一条岔路上，最终成为奇形怪状的自己。而我又抱着类似的希冀与偏执，打了几次挺，来到了如今的母校，并一定程度上再次重复了幻想与放弃的过程。只是在我以为会望到尽头的研究生生活中，与诸多未曾设想的人与事不期而遇，并走向了人生剧本前几幕中一个还算不错的小结局。

在写代码还是论文时，我似乎总有些洁癖，总是试图寻找更完美的解决方案与排版格式，并在其中蹉跎了大量时间。也许在几次反复后，项目终究可以收敛于目前可见的最佳方式。但人生却不会像 ADV 一样可以存档读档，二次修正，从来没有完美的人生与故事，不过正如阿德勒所言，人的可爱之处便在于接受不完美的自己。过去所有的人与事就如同正向动力学中的骨骼解算，连续的关节依次计算出了如今的自己。仅仅某个节点的偏差，便可能使端点大相径庭。我也想不出有限骨骼抵达目前结局的更好解算方式，因此我很感谢在我人生轻小说剧本中出场的关键人物们与此期间的变量。

首先衷心感谢我的引路人 XXX 老师，这篇论文的顺利完成，离不开她的悉心指导和鼓励。在实验室的学习生涯和平时的生活中，也从 X 老师这里学到了很多。其次则要感谢我的导师 XXX 老师，平易近人的 X 老师给了我更多的自由空间。也十分感谢 XXX 老师，在游戏策划课上以及实验室例会等，从 X 老师这里开阔了视野与眼界。还有在研一研二学习课程时，学院老师们给予的指导。他们的言传身教将使我终生受益。而我想，这大概也是中传生涯这块骨骼解算的起点。

也感谢实验室的同窗伙伴们 X sXX、XXX、XXX、XX、XXX 和 XXX，和他们共同度过的实验室生活是我难以忘怀的宝贵记忆。感谢 XXX 师兄解答了很多写论文时规范经验的问题，感谢 XXX、XXX 师妹们帮忙校验。感谢实验室所提供的学习平台与轻松愉快的氛围，也感谢 CBD 中的每一位同学。感谢过去与现在的室友们，以及同班同学们在校园所交织的故事。感谢腾讯与支付宝实习期间遇到的同事们，从他们身上我学到了不少开发经验。感谢开源社区与开源项目们，让我得以快速完成开发工作，并从中学到了许多代码、项目的实践与思考。感谢相关课题与参考文献的研究学者们，感谢 L^AT_EX 与 ThuThesis，让我得以更好地完成论文的撰写与排版工作。

我也想对父母道声谢，在家写论文期间他们为我营造了一个衣食无忧，安静的学习环境，也是他们的支持和鼓励推动着我成为现在的自己。

如今我又将抱着新的期冀启程，幻想玫瑰色的生活，续写有限又无限的可能，直至我的状态机抵达终态。我不知道到那时骨骼解算的终点在何处，但我想，当我的剧本落下帷幕时，我一定还会回忆并感谢此间登场的变量与关键人物，你们。