毕业设计开题报告

题　　目：　地形生成及环境演变

学　　院：　数据科学与软件工程学院

专　　业：　软件工程（服务外包）

年级（班）：　2015 级微软 2 班

学　　号：　201540704318

姓　　名：　韩沁东

指导教师　：　周　强

2019 年 4 月 1 日

一、 设计背景

对这个问题的研究始于好友千里冰封对 Minecraft 地形生成算法的研究。他使用的是噪声算法，但是实现效果并不理想，基于他的学习，我开始研究更符合实际的地形生成算法。我实现了两种算法，但当时的实验平台耦合性过高，若要增改代码需要改动大量内容，于是借此机会实现一个易扩展、功能丰富的相关算法的实验平台。并且希望能为 CG 相关同志提供方便。

二、 理论及现实意义

为相关行业相关需求提供辅助功能。如游戏、影视行业中大型 3D 地形的的粗略建模生成，并为其后续工作提供相关辅助，如河流的生成、更真实的水域分布和生物分布。

三、 设计现状和发展趋势：

市面上常见相关软件有 WorldMachine、QuadSpinner Gaea 等。

WorldMachine 和 QuadSpinner Gaea 在操作逻辑上有许多相同之处。他们都是使用节点去控制过程中参数和对逻辑进行关联，来控制最终的生成数据。

他们都使用不同生成器节点根据不同的参数生成不同的高度数据，比如低起伏的平原，略有起伏的丘陵和起伏较大的山地。而后使用其他功能节点对一个或多个生成器节点的数据进行融合，比如简单的对应数据加减、多张地形中取最大值或最小值、按比例相加等，最终生成一个完整的地形数据。

在对数据处理时，一个地形的高度数据可以看成一张灰度图像，高度指示图像某点的亮度。在对数据整合时可以类比 PS 中多图层的混合方式。

1. 常见的地形生成的算法

- **噪声图**

Minecraft 作为一款非常自由的沙盒游戏，它实现了基本无限大的地图（受到 int 类型的数值上限印象），并且可以在非常快的速度内加载出任意一个指定区域的数据。好在这款游戏是开源的，可以使所有人都了解到工作原理：噪声。噪声的应用在 3D 工作中是比较重要的，许多时候为了体现出世界的不完美，会用噪声去添加一些自由的、不完美的因素。对于一个不平的地面来说，噪声不失为是一种不错的方法。Minecraft 会先生成一个只区分陆地和海洋的地形，然后再慢慢丰富不同的矿产、生物系群等。

Minecraft 所使用的地形生成的算法在各种游戏中应用非常广泛，比如红色警戒 2 中自定义战役的随机地图功能，很可能就是相同的应用。这种算法实现简单，可并行化，已经成为了许多 GPU 并行运算教程的经典例子，并且可连续。可连续就是从一个点的南方向这个点进行生成运算和从这个点的北方向这个点进行生成运算，结果是一样的，或者叫做拥有一致性。

但是，这种算法也是有问题的，在好友千里冰封的实现中发现生成的地形并不是那么真实（如图一），我把这种不真实归结为太完美。这里的完美不是地势平坦的完美，而是说躁波生成的高度值山峰的分布、山谷的分布基本是平均在一大片区域的，没有山地、平原、丘陵的明确区分。真实情况中因为造山运动等地质活动，使得山地是集中出现的，平原也是集中的。并且千里冰封的实现生成的

结果地形过于规整，边缘过于整齐，给人一种很不舒服的感觉。

图一，千里冰封的地形生成实现

- **Diamond-square algorithm**

还有一种生成算法，叫做 Diamond-square algorithm，该算法是由 Fournier, ussell 和 Carpenter 在 1982 年提实现的[1]。这种算法的优点在于可以实现指定想要的世界的大体的样子，并且他的点的高度值是与周围点相关的。在一开始，算法会选取几个分散点作为特征点，并为它们设定高度。这些特征点的高度指示着整个地形的高度趋势。之后迭代的对整个地形进行细分和相应点的高度设定。

但是这个算法的问题也显而易见：

1. 它不能快速的对任意一个指定的区域生成数据，因为这个算法是一个由粗糙精细的细化过程，在没有精细化到最低级或是选取的点没有被选中为特征点时，我们要继续做精细化的迭代。
2. 在获取较大的地图时，很难对地图进行分块单独处理，特别是在当年内存容量不大时，需要做特别的优化。

3. 地图必须为正方形，且边长为 $2^n+1,n\in Z$。
4. 不容易做到 GPU 运算。因为这个算法对于运算时序是有要求的，gpu 的并行运算并不能很好的发挥。

2. 常见的环境模拟算法

关于环境模拟的内容，很多游戏与地图生成方法类似，也是使用噪声图像来对某种资源、生物的分布进行安排。Minecraft 对生成方法进行了一些改进，有些像 Diamond-square algorithm，但并不完全一样。优点同上，快速、可并行运算、有一致性。但是缺点在 Minecraft 中也暴露的很明显：不真实。他生成的世界中寒冷和温和是交替的，并不具有现实中从低纬度到高纬度规律性的变化。

不过在 2017 年的 GDC 会议中，有团体展示了他们的一个整合了地形生成到环境演变到文明聚集的项目[2]。这个项目生成了一个 32k x 32k 的高度图，计算出了 11 种生物群的分布，其中拥有 16.369km^2 的水域面积，并且生成了 656.421km 长的道路，90.411km 长的铁路，56 个村庄。但是，这个项目中并没有实现非常高的自动化，许多内容都是由设计师去指定。

比较接近真实的情况需要用到大量流体力学的知识去模拟空气的流动、降水的流动。这类算法的算法多数是用在工业级 CG 工作中，如电影制作、飞机研究等。随着计算机计算能力的增加和 gpu 运算的流行，使得家用电脑也可以快速的进行少量物体的运动模拟。比如 NVIDIA PhysX FleX[3]可以对流体、布料等进行运动演算。

这类模拟方式都是使用许多的计算微元去模拟真实中大量的水分子。微元的具体单位是根据情况而定的，比如在远观大海的场景中，一个微元代表的真实水

体积要比模拟一杯水倒在地上的微元大得多。这些微元在每秒钟几十到几百次的碰撞、吸引的迭代中模拟真实的情况。这类算法的效果主要由这么几个方面去体现：微元的大小和每次迭代的步长。微元的体积约接近于水分子的真实体积，迭代的步长越短，越能表现的真实。若迭代步长太长，可能会出现两个例子穿越彼此的情况。

还有种算法是从宏观规律上得出流体的运动公式，从宏观模拟。这种方式速度快，单一元素效果好。但是实现效果是与运动公式的准确性成正比的，并且在复杂情况下运算逻辑会变得异常复杂，多种流体混合运算时需要单独去进行公式的编写。

四、 主要实现内容及解决的问题

实现对地形生成的算法和环境模拟的算法提供统一的实验平台。提供地形生成算法接口、环境演变算法接口、结果的可视化接口、结果的查错功能和结果导出功能。

对随机趋势化地形生成算法进行可行性实验。

五、 工作计划与方案

1. 实施方案

软件基于.net framework4.7 开发，因内存限制，目标平台为 x64，开发工具为 visual studio 2017/2019。使用 c#作为主要的后端语言，如果需要，可以使用如 F#、C++/CLI、VB.Net 等其他基于.net 开发的语言，或是与 native code

进行交互。

使用 WPF 作为 UI 框架，配合 XAML 实现前端。前端使用数据绑定实现前后端数据的通信。

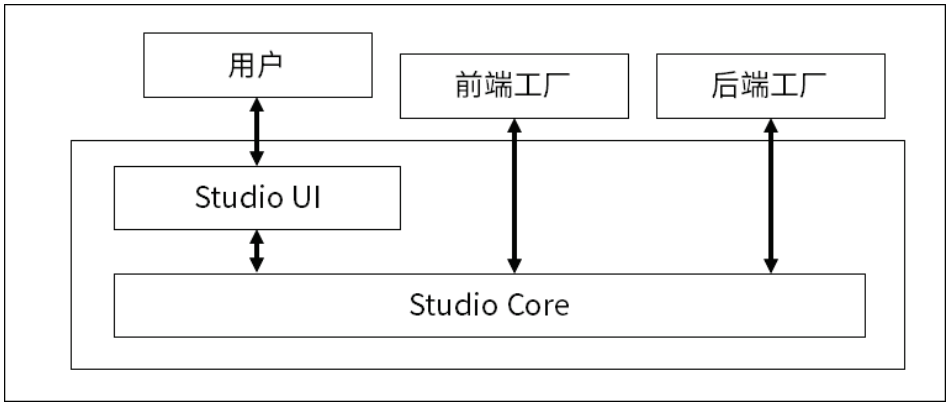
软件分为 Studio、前端算法和后端算法三部分。

Studio 提供交互界面和统一的 API 接口，studio 必须使用.net framework 开发。前端算法为地形生成算法，实现时需要能够提供与 studio 交互的部分，为了效率核心算法可以使用特殊优化的 C++/CLI 代码或是 native 代码完成。后端算法为环境演变算法，要求与前端算法一致。

Studio 分为两大部分：Studio Core 和 Studio UI。

Studio Core 提供整个项目使用到的自定义的接口、基类和基本类型，是整个软件的基础。Studio UI 提供用户界面和用户操作方法，充当用户和 Studio Core 之间的桥梁。

整个软件基本依赖关系如下图图二：



图二，软件基本依赖关系与用户关系图

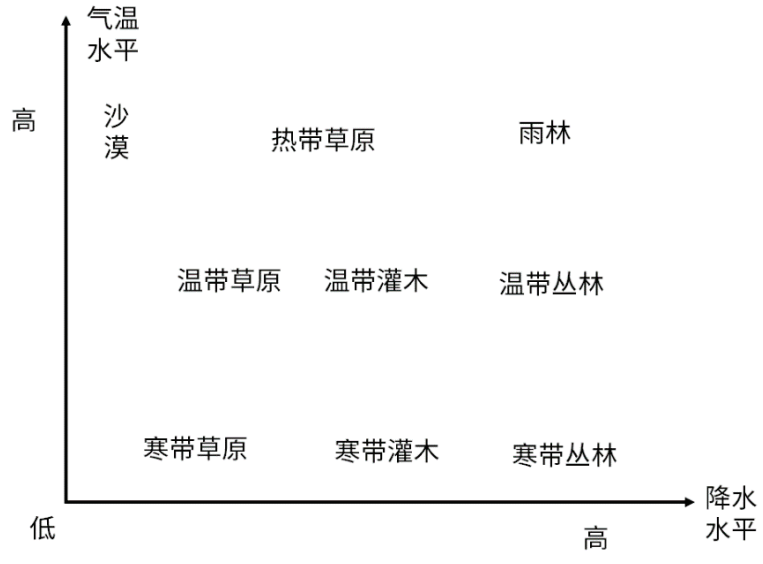
- **地形生成相关内容**

地形以二维高度图形式存储，点(x,y)的值为点(x,y)的高度，类型为 signed

int_32，高度单位不确定，在需要时由用户设定。长轴、宽轴单位相同，但不必与高轴单位相同。

- **环境模拟相关内容**

环境模拟分为空气模拟、水循环模拟、光照模拟及生物群模拟，空气模拟为模拟大气流动，为之后降水的运算提供方便。水循环模拟为降水模拟及降水汇集为江河湖海泊的过程的模拟。光照模拟是模拟光照环境（主要为高山的向阳面、背阴面），与云雾情况相融合得出一个区域的光照水平及气温情况。根据降水和光照，可以得出 9 种大致的陆地自然生物群系统和 2 种大致的水环境自然生物群系统。这里有一些概念是借鉴的游戏 Minecraft 中的概念，不得不说，Minecraft 在对模型的抽象工作上确实做了不少的贡献。图三为 9 种陆地自然生物群系统。



图三，9 种陆地自然生物群系统

2. 工作计划

3 月前完成软件原型设计、API 接口设计和功能逻辑设计。

4 月前完成 Studio 主体与前端算法的开发。

4 月初至 4 月中旬完成后端算法的开发。

4 月中旬到 5 月末进行整体完善。

六、 参考文献

[1]: Communications of the ACM. 25 (6): 371–384

[2]:https://www.gdcvault.com/play/1024029/-Ghost-Recon-Wildlands-Terrain

[3]: https://developer.nvidia.com/flex

Obj 文件格式：https://en.wikipedia.org/wiki/Wavefront_.obj_file