



青岛大学
QINGDAO UNIVERSITY

本科毕业论文(设计)

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

姓 名： 韩沁东

指导教师： 周 强

2019 年 5 月 17 日

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，而是说噪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 $2^n+1, n \in \mathbb{Z}$ 。
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 16.369km² 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。

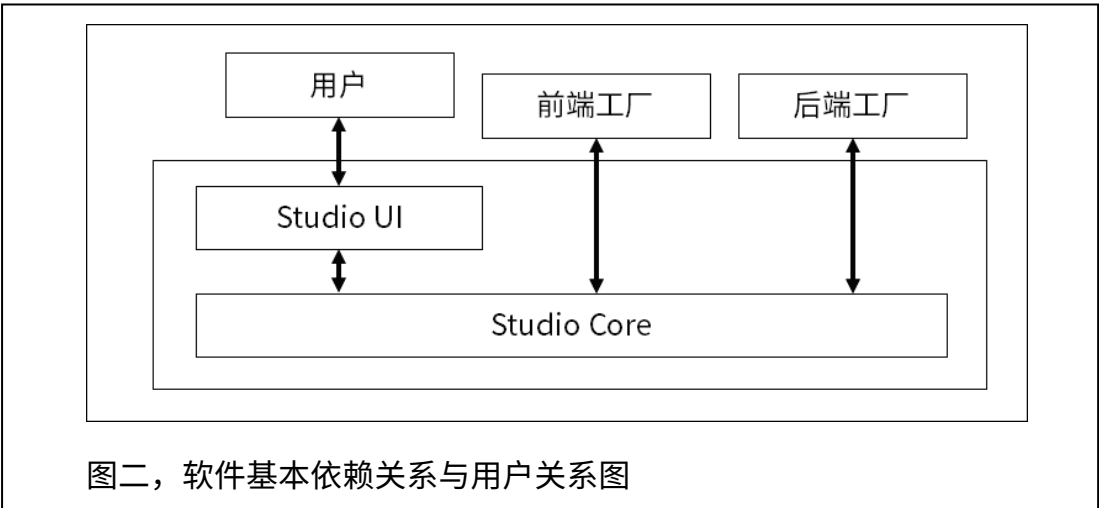
软件分为 Studio、前端算法和后端算法三部分。

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：



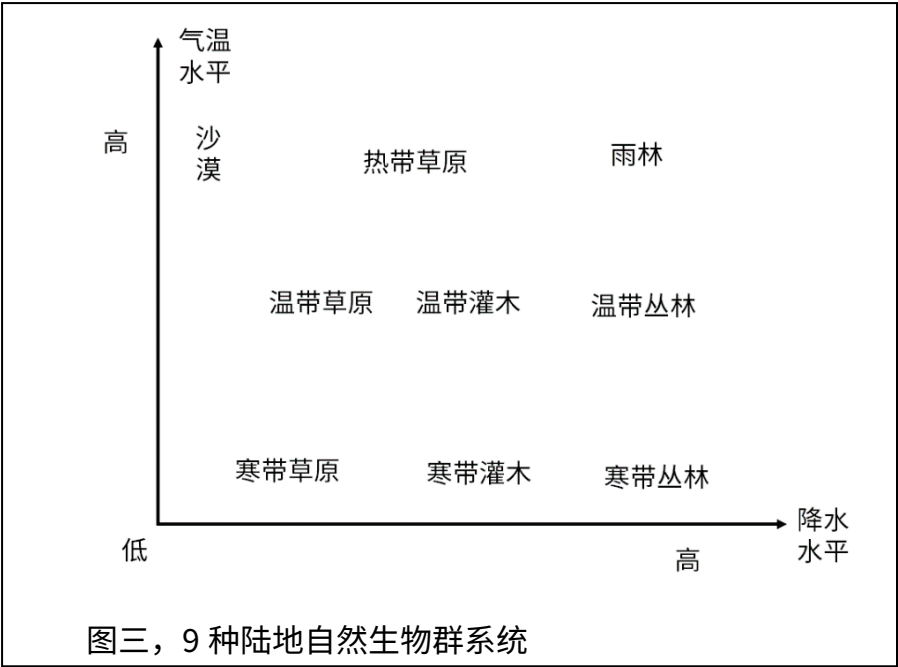
● 地形生成相关内容

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。



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

附件 1:

青岛大学本科毕业论文（设计）任务书

论文（设计）题目	地形生成及环境演变		
学 院	数据科学与软件工程	专 业	软件外包
学生姓名	韩沁东	学 号	201540704318
选题来源	教师科研课题（ ）；生产、工程或社会实践课题（ ） 学生自拟课题（ <input checked="" type="checkbox"/> ）；师生共同拟定课题（ ）		
	大学生创新创业训练项目（ ）； 学科竞赛 （ ）		
<p>研究目标或设计构想：</p> <p>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。</p> <p>对随机趋势化地形生成算法进行可行性实验。</p> <p>所有数据以二维图的形式描述。</p> <p>工作以类流水线形式进行，各工作点需依次完成，不可乱序。</p> <p>程序对工作内容以工程形式进行管理，工程内包含多个工作，每一个工作内容相互独立，又可以通过一些特殊流程相互联系。</p>			
<p>研究内容或设计方案：</p> <p>软件分为 Studio、前端算法和后端算法三部分。</p> <p>Studio 提供交互界面和统一的 API 接口，studio 必须使用 .net framework 开发。</p> <p>前端工厂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。</p> <p>后端工厂算法为环境演变算法，要求与前端算法一致。</p> <p>对于前端工厂算法，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。</p> <p>前端工厂算法需实现随机趋势化算法。</p> <p>后端算法过程分为空气模拟、水循环模拟、光照模拟及生物群模拟四个过程，后边的过程可依赖之前的过程结果，所以四个过程需依次完成。</p> <p>后端工厂算法需实现各自快速的算法。</p>			
<p>研究方法或技术路线：</p> <p>软件基于 .net framework 开发，开发工具为 visual studio 2017/2019。使</p>			

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。

进度安排：

第 1-4 周 完成软件原型设计、API 接口设计和功能逻辑设计

第 5-8 周 完成 Studio 主体与前端算法的开发

第 9-10 周 完成后端算法的开发

第 11-14 周 进行整体完善准备答辩

论文起止时间：

2019 年 2 月 25 日—— 2019 年 5 月 15 日

学生（签名）：

教研室意见：

指导教师（签名）：

教研室主任（签名）：

年 月 日

年 月 日

注：本表格由学生填写，指导教师审核同意后，由教研室审批留存。

郑重声明

本人呈交的学位论文（设计），是在指导教师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。除文中已经注明引用的内容外，本学位论文（设计）的研究成果不包含他人享有著作权的内容。对本论文（设计）所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文（设计）的知识产权归属于青岛大学。

本人签名：_____ 日期：_____

WorldCreator 软件开发文档

文档版本：1.2

软件版本：1.2

历史版本

版本	作者	参与者	起止日期	备注
1.0	韩沁东		2019.2.9-2019.4.23	
1.1	韩沁东		2019.4.25-2019.5.2	
1.2	韩沁东		2019.5.3-2019.5.17	

一、 文档规范	6
1. 字体及样式使用规范	6
2. 文档内编程语句规范	6
3. 注释规范	6
二、 软件需求说明	7
1. 开发目的	7
2. 软件功能要求	7
3. 开发技术及软件运行目标环境	8
4. 相关链接	8
三、 软件逻辑设计	9
1. 工程管理逻辑	9
2. 功能使用操作流程及方式	9
四、 软件数据存储结构及节点逻辑结构定义	10
1. 工程文件存储格式	10
2. 工程文件的物理存储逻辑	12
3. 数据图结构定义	12
4. 工程的展示逻辑	13
五、 代码规范规约	14
1. 命名风格	14
2. 代码格式	14
六、 运算结果数据类型定义及限定	15
1. 前端工厂	15
2. 后端工厂 AtmosphericMotion	15
3. 后端工厂 RainfallMotion	16
4. 后端工厂 SolarIlluminance	16
5. 后端工厂 Biomes	17
七、 软件模块划分及关系	18
1. 模块功能简介	18
2. 模块依赖关系图	19
八、 WorldCreatorStudio_Core 模块功能及重要模块细则	20
1. 子命名空间或逻辑文件夹功能描述	20

2. [I]ListNode/IWorkLogicNodeAble	20
3. ListNode/IDataCalclaterAble	23
4. ListNode/IDataCalclaterConfigurationAble	24
5. ListNode/IDataCalclaterFactoryAble<CalcT, ConfigT>	25
九、 WorldCreatorStudio 模块功能细则	26
1. [static]Commands.....	26
2. [static]Resouses/StoreRoom	27
3. [static]Resouses/Icons	27
4. [WPF]Resouses/ControlTemplates [WPF]Resouses/Theme.....	27
5. [WPF]Windows/MainWindow	27
6. [WPF]Windows/NewProject.....	28
十、 BackEndFactorys/AtmosphericMotion/MiRaI.BE.AM.SingleValue 模块功能细则.....	29
1. SingleValue	29
2. SingleValueConfig	29
3. SingleValueFactory	30
十一、 FrontEndFactorys/RandomTend 模块功能细则	30
1. RandomTendCreator	30
2. RTConfiguration	31
3. RandomTendCreatorFactory	32
十二、 BackEndFactorys/RainfallMotion/MiRaI.BE.RM.SingleValue 模块功能细则	32
1. SingleValue	32
2. SingleValueConfig	33
3. SingleValueFactory	33
十三、 BackEndFactorys/SolarIlluminance/MiRaI.BE.SI.QuickCalculating 模块功能细则	33
1. QuickCalculating.....	33
2. QuickCalculatingConfig	34
3. QuickCalculatingFactory.....	34

十四、 BackEndFactorys/Biomes/MiRal.BE.BI.QuickCalc 模块功能细则	35
1. QuickCalc.....	35
2. QuickCalcConfig.....	35
3. QuickCalcFactory	36
十五、 ValueToImage 模块功能细则	36
1. 关于常见图片格式的说明	36
2. [static]ValueToGrayImage	37
3. [static]ValueToColorImage.....	37
4. [static]AtmosphericMotionToImage.....	37
5. [static]RainfallMotionToImage	38
6. SolarIlluminance	38
7. BiomesToImage.....	38
十六、 WorldCreatorStudio_Resouses 模块功能及重要模块细则	38
十七、 ValueTo3DModel 块功能及重要模块细则	39
1. [static]HeightToModel.....	39
十八、 软件测试内容概要	40
1. 新建工程	40
2. RMNode 测试、SINode 测试、BINode 测试类比 AMNode 测试 .	42
十九、 总结汇报	42
1. 本版出现的问题	42
2. 下版改进	43

该项目已使用 MIT 开源协议 (<https://opensource.org/licenses/mit-license.php>) 开源，软件开源地址：<https://github.com/ijrys/WorldCreator>

一、文档规范

1. 字体及样式使用规范

不使用非商用免费字体

中文使用思源黑体、黑体、隶书等免费字体

英文使用 consolas

2. 文档内编程语句规范

非必要情况下，不出现代码语句，只展示定义。

类型使用浅色标注(rgb(128,128,128))。

定义的名称使用加粗字体。

私有内容使用斜体。

方法定义，若参数列表为空，括号与函数名保持在同一行；若参数列表不为空，参数列表单独一行。。

代码使用“代码”格式。

如：

```
public static string Fun1()  
  
private string Fun2  
(string str, int start)
```

单独的代码块使用外侧框线框起来，配合表格使用的代码段（如模块具体功能中的方法的声明和解释）可忽略外框线

3. 注释规范

可使用中括号括起来的内容对内容做注释，并对内容以浅色颜色(rgb(128,128,128))

如

- 统一注释：

[WPF]：使用了 WPF 技术的模块

[static]：静态的模块

[i]：interface，接口

[a]：abstract，抽象类

[FE]：前端工厂（FrontEndFactory）相关

[BE]：前端工厂（BackEndFactory）相关

二、软件需求说明

1. 开发目的

整合三维地形生成及环境模拟运算，为非精确的大规模地形建模工程提供便利，为快速的地形生成算法做实验性研究平台。并且基于此软件提供的平台实验自行设计的随机趋势化地形生成算法，为以后的实验提供便利。

2. 软件功能要求

提供一个统一的、易扩展的平台环境，为相关算法的实验、应用提供一个可行的环境。Studio 要有对工程、项目和相关资源管理能力。

算法分为前端工厂部分和后端工厂部分。前端工厂为地形生成器，该部分要通过接收一组设定值而产生一个合理的二维的高度数据，作为地形的高度信息。后端工厂要根据用户的设定和其他后端工厂的产出数据，对前端工厂产出的地形进行合理的环境模拟。

软件将实现前端工厂的随机趋势化算法和后端工厂中的空气流动模拟、降水模拟及生态系统模拟。Studio 要求能对工厂产出的数据进行整合，并且可以导出为三维模型或是二维贴图，以方便后续对产出数据的使用。

3. 开发技术及软件运行目标环境

a) 开发技术

- C# (C#7.0)
- WPF (WPF4.5)

b) 开发环境

- .NET Framework (.NET Framework 4.7.2)
- Visual Studio 2017 Community 及兼容产品

c) 软件运行目标环境

- .NET Framework 4.7.2 及兼容版本
- Windows 10 1607 及后支持.NET Framework 4.7.2 及 WPF 的操作系统
- 显示器 800 * 600 及更大分辨率
- 内存：剩余 1G 以上可用的内存空间
- 磁盘：剩余 2G 以上可用的磁盘空间，磁盘可读写

4. 相关链接

- C#7.0

<https://docs.microsoft.com/zh-cn/dotnet/csharp/whats-new/csharp-7>

- WPF4.5

<https://docs.microsoft.com/zh-cn/dotnet/framework/wpf/getting-started/whats-new>

- .NET Framework 4.7.2 Download Page

<https://dotnet.microsoft.com/download/dotnet-framework-runtime/net472>

- Visual Studio

<https://visualstudio.microsoft.com/zh-hans/>

- MIT 开源协议

<https://opensource.org/licenses/mit-license.php>

三、软件逻辑设计

1. 工程管理逻辑

软件工程逻辑为树形分层逻辑，根节点为工程节点（Project）。工程节点下分为多个工作节点（Work），每一个工作节点对应一个地形从生成到最终的环境模拟演变结束的整个过程。在同一工程内的各个工作是相互关联而又相互独立的，他们既可以是同一个项目的不同部分的地形，也可以通过更强大的合成器将多个地形合成到一起，实现更加复杂的效果控制。

工作分为图片资源管理器、前端工厂和后端工厂。图片资源管理器管理整个工作过程中产生的图片资源，进行统一管理。其他节点只能通过引用这些资源来进行图片资源的分类。

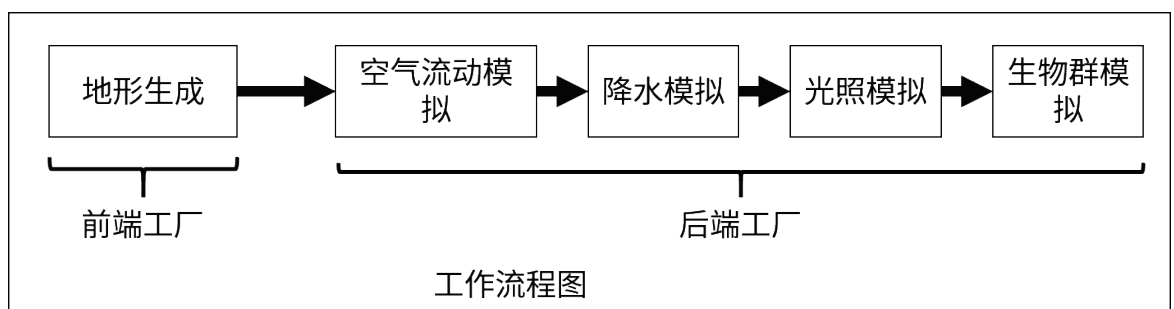
2. 功能使用操作流程及方式

用户通过鼠标、键盘进行输入，并通过 ui 界面获得操作输出。

使用统一的 ui 平台对算法功能进行整合，用户通过控制 ui 平台来实现对算法的调用和数据的获取。

各算法按照算法目的分为五大部分，对应于工作的五个步骤。使用流水线工作方式，只有在依赖的前一工作点完成工作后才可使用当前工作点。

流程示意如下：



同一节点只依赖之前节点产生的数据，不依赖之前节点的数据生成算法类型，故可根据需求自行选择合适的算法或自行扩展。

四、软件数据存储结构及节点逻辑结构定义

1. 工程文件存储格式

工程涉及到两种文件存储，为工程文件和工作文件。

a) 工程文件

工程文件后缀名为 **mriwcpro** (MiRal World Creator Project)，以标准 xml 文档格式存储。

工程文件记录了工程的基本信息 (GUID、工程名) 和工程拥有的工作信息 (GUID、工作所在的文件夹、工作文件名，工作名)。

工程文件内容具体定义如下：

```
<project guid="[工程 GUID]" name="[工程名称]">
  <work guid="[工作 1GUID]" dictionary="[工作 1 存储文件夹]"
file="[工作 1 文件]" name="[工作 1 名]" />
  <work guid="[工作 2GUID]" dictionary="[工作 2 存储文件夹]"
file="[工作 2 文件]" name="[工作 2 名]" />
</project>
```

如

```
<project guid="c99a3e2b-58eb-47e4-1234-85cf72f19deb"
name="DemoPro">
  <work guid="46a1d881-f42d-4a7e-4567-0123456789ab"
dictionary="demowork" file="demowork.mriwcw" name="测试工作
集" />
</project>
```

b) 工作文件

工作文件后缀名为 **mriwcw** (MiRal World Creator Work)，以标准 xml 文档格式存储。

工作文件记录了工作的基本信息 (GUID) 和工作拥有的图片资源(images 节点)、工作的前端工厂信息、工作的后端工厂信息。

工作文件内容具体定义如下：

```
<work guid="[工作 GUID]">
  <images>
    <image key="[资源 1 唯一键]" name="[资源 1 展示名称]"
file="[资源 1 文件，根目录为工作文件夹下 images 文件夹]"
description="[对资源 1 的描述信息]" />
  </images>
</work>
```

```

        <image key="[资源 2 唯一键]" name="[资源 2 展示名称]"
file="[资源 2 文件，根目录为工作文件夹下 images 文件夹]"
description="[对资源 2 的描述信息]"/>
    </images>

    <FrontEndFactory creator="[记录 Creator 的 ProgramSet 信息]">
        <setting>
            [该节点为相应类产生其具体内容]
        </setting>
        <Resault name="" value="[运算结果的存储文件]"
imgrefkey="[用于展示的图片资源 key]" />
    </FrontEndFactory>

    <BackEndFactory>
        <AMNode creator="[后端工厂的 ProgramSet]" state="[节点的状态]">
            <Resault dataName="[运算结果的存储文件]"
imgrefkey="[用于展示的图片资源 key]" />
            <Config>
                [该节点为相应类产生其具体内容]
            </Config>
        </AMNode>
    </BackEndFactory>
</work>

```

如

```

<work guid="54fe1651-5c29-45c5-9a31-379e4b18b100">
    <images>
        <image key="FE.RandomValue"
file="eec6e1f3e0a64d4c97962b64bf67.png" description="前端工厂的
随机值图" />
        <image key="FE.HeightValue"
file="36089b82aacb4afa97544c5d5d8c.png" description="前端工厂高
度值图" />
        <image key="BE.AM.Map"
file="684083b24a3a4b08ab8aa9c6da1c.png"
description="AtmosphericMotion Visual Map" />
    </images>
    <FrontEndFactory
creator="MiRaI.RandomTend.RandomTend|0.1">
        <setting>
            <add key="blockSize" value="5" />
            <add key="wbn" value="1" />
            <add key="hbn" value="1" />
            <add key="seed" value="2019" />
        </setting>
        <Resault name="" value="HeightValue"

```

```

imgrefkey="FE.HeightValue" />
</FrontEndFactory>
<BackEndFactory>
    <AMNode creator="MiRaI.BE.AM.SV|0.1" state="ok">
        <Resault dataName="AtmosphericMotion Visual
Map" imgrefkey="BE.AM.Map" />
        <Config power="76" dir="NW" />
    </AMNode>
    <RMNode creator="MiRaI.BE.RM.SV|0.1" state="ok">
        <Resault dataName="RainfallMotion Visual Map"
imgrefkey="BE.RM.RIMap" riimgrefkey="BE.RM.RIMap"
atimgrefkey="BE.RM.ATMap" />
        <Config ri="4020" sl="0" />
    </RMNode>
    <SINode creator="" state="ready" />
    <BINode creator="" state="unable" />
</BackEndFactory>
</work>

```

2. 工程文件的物理存储逻辑

```

Project
|-proj.mrimcproj [工程文件]
|-Geo1
|   |-geo1.mrigeoproj [工作文件]
|   |-Images [工作中产生的图像资源]
|   |   |-img1.png
|   |   |-.....
|   |-HeightValue [高度数据]
|   |-AtmosphericMotion Visual Map [空气流动数据]
|   |-RainfallMotion Visual Map [降数据]
|   |- [其他产生的过程数据]
|-Geo2
|-.....

```

3. 数据图结构定义

数据均为矩阵，内部操作时使用二维数组实现存储。

数组第一索引为行号，第二索引为列号，如下图(4.3.1)所示

0行	Map[0,1]	Map[0,2]	Map[0,3]	Map[0,4]	Map[0,5]	……
1行	Map[1,1]	Map[1,2]	Map[1,3]	Map[1,4]	Map[1,5]	……
2行	Map[2,1]	Map[2,2]	Map[2,3]	Map[2,4]	Map[2,5]	……
3行	Map[3,1]	Map[3,2]	Map[3,3]	Map[3,4]	Map[3,5]	……

图 4.2.1

数据存储时先存储数据图宽、高(int_32)，再按行存储其中每一列数据。伪代码如下：

```
Write (map.Width);
Write (map.Height);

for (int h = 0; h < map.Height; h ++) {
    for (int w = 0; w < map.Width; w ++) {
        Write (map[h, w]);
    }
}
```

数据图按 0 行在上，0 列在左处理。方位上按上方为北，左方为西处理。

为了兼容各个流程，对于图的数据意义和长宽进行限制，规定如下：

- 生成的地形边长应均为 $n * 2^k$ ， n 、 k 均为正整数，且 $k \geq 4$ 。
- 高度图中每个点代表的是一个正方形块的角的高度，则高度图的数据数量为 $(a + 1) * (b + 1)$ ， a 、 b 分别为设定目标地图的长、宽。
- 后端工厂对数据的处理是基于块的，记录的是块的数据，则后端工厂生成的数据量应为 $a * b$ ， a 、 b 分别为设定目标地图的长、宽。

4. 工程的展示逻辑

软件中对用户展示一个逻辑树，来展示整个工程。用户通过对逻辑树的相应节点进行操作从而控制整个工程。

```
Project 【不可点击】
|-Geo1 【工程设置】
| |-RandomMap 【随机数图】
| |-HeightMap 【高度图】
| | |-SubMap 子图 1
| | |-SubMap 子图 2
| |-TerrainMap 【地势图】
| |-LaterFactory 【后期工厂】
```

```

|   | -AirMotion 【空气流动图】
|   | -Precipitation 【降水】
|   | - 【水域】
|   | - 【生物群】
| -Geo2
|   | -RandomMap 【随机数图】
|   | -HeightMap 【高度图】
|   | -TerrainMap 【地势图】
|   | -LaterFactory 【后期工厂】
|   | -AirMotion 【空气流动图】
|   | -Precipitation 【降水】
|   | - 【水域】
|   | - 【生物群】

```

五、代码规范规约

1. 命名风格

- 程序集、命名空间、类名、方法名、属性名使用 UpperCamelCase 风格，如 FrontEndFactory

- 参数名、成员变量、局部变量使用 lowerCamelCase 风格，如 inputName

- 私有字段使用下划线开始，如_nodeName
- 异常类命名使用 Exception 结尾
- 接口以 I 开头，以 Able 结尾，如 IwriteToFileAble

2. 代码格式

- 大括号的使用约定。

如果是大括号内为空，则简洁地写成{}即可，不需要换行；

如果是非空代码块则 左大括号前不换行，左大括号后换行。右大括号前换行。右大括号后还有 else 等代码则不换行；表示终止的右大括号后必须换行。

- 左右小括号和字符之间不出现空格。

- 缩进

级别缩进使用 tab，统一级别的代码块应保证前边有相同数量的 tab

内容缩进在使用 tab 实现级别缩进后按需使用空格进行内容对齐

- 换行

在一行代码过长时进行换行保证阅读方便。换行时应保证语句块不被拆分（如调用方法时不将方法名与参数列表拆在两行），调用操作符应跟在新行，其他操作符保持在旧行。

- 代码格式示例

```
public static void Function(int id,
.....string name,
.....string nickName){
→   if(id < 10 &&
→   ....name != null){
→   //do something
→   } else {
→   →   nickName.DoFun1('.')
→   →   .....DoFun2("||")
→   →   .....DoFun3();
→   }
}
```

六、运算结果数据类型定义及限定

- 涉及到的类型定义

符号	描述	在 c#中的符号
int_32	有符号的 32 位整形	int
unsigned_int_8	无符号的 8 位整形	byte

1. 前端工厂

前端工厂产生数据为 int_32 类型的二维数组。

每一个数据指示相应点的高度。为了后续工作的需要，数据范围应为 $[-2^{29}, 2^{29}-1]$ ，即 $[-536870912, 536870911]$

2. 后端工厂 AtmosphericMotion

空气流动模拟生成的数据应表示两个属性：流动的方向和速度

为了简化需求，速度为 `unsigned_int_8`，与真实风力级别对应关系为真实风力= $\text{value}/20$

流动方向简化为 9 个方向，中心表示无风，枚举值为 0，其他各方向以向西北方开始顺时针按序排列。枚举值具体见下图：

↖ (1)	↑ (2)	↗ (3)
← (8)	● (0)	→ (4)
↙ (7)	↓ (6)	↘ (5)

如空气向西北方流动（东南风），枚举值为 1；空气向南方流动（北风），枚举值为 6。

3. 后端工厂 RainfallMotion

降水模拟生成的数据应表示三个属性：降水强度、水深、水域类型。

降水强度为每单位时间降下 1 个全局高度单位的水，类型为 `int_32`，但数据范围应在 $[0, 2^{20}]$ 甚至更小范围内的合理值。

水深表示当前区域的水体水深，类型为 `int_32`，单位与全局同单位。

水域类型表示当前区域的水体类型，枚举值对应关系如下表：

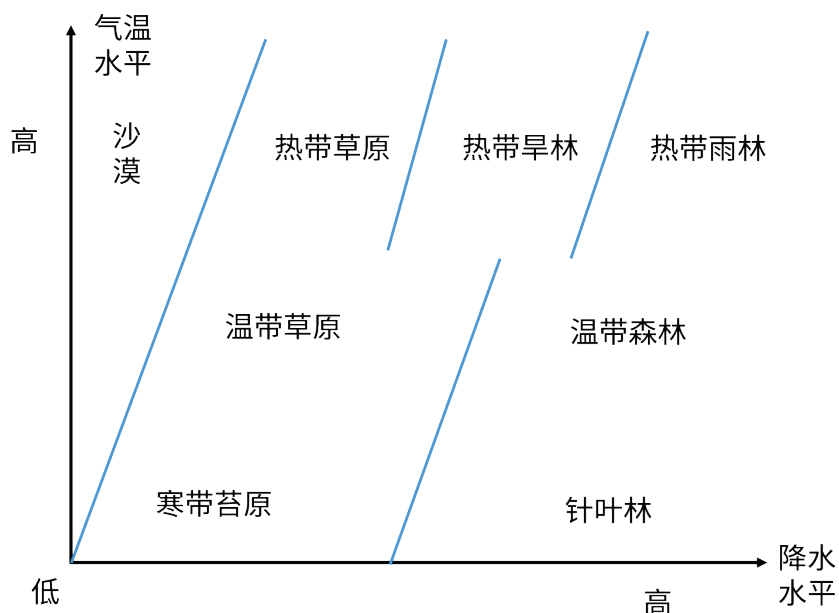
中文描述	枚举名	枚举值
陆地	land	0
江，河	river	1
湖	lake	2
海，洋	sea	3
沼泽，湿地	marsh	4

4. 后端工厂 SolarIlluminance

光照模拟输出为每个区域的光照辐射强度。数据类型为 `unsigned_int_8`，数据范围为 $[0, 255]$ 。

5. 后端工厂 Biomes

生物群模拟主要依靠之前的数据对区域内生物类型进行判别，按照降水和光照条件，可大致分为以下八种陆地生物群，同时水域生物群可分为淡水水域和咸水水域两种。



每种枚举值的关系如下表：

中文描述	枚举名	枚举值
淡水水域	FreshWater	0b1_000_0000
咸水水域	SaltWater	0b1_001_0000
沙漠	Desert	0b0_000_0000
炎热草原	TropicalSavanna	0b0_010_0001
炎热旱林	ThornForest	0b0_010_0010
炎热雨林	TropicalRainforest	0b0_010_0011
温和草原	TemperateGrassy	0b0_001_0001
温和森林	TemperateForest	0b0_001_0010
冰雪苔原	Tundra	0b0_000_0001
针叶林	ConiferousForest	0b0_000_0010

七、软件模块划分及关系

1. 模块功能简介

a) WorldCreatorStudio

软件 UI 提供模块，将各模块功能整合提供完整的操作流程，是用户与 WorldCreatorStudio_Core 模块的桥梁。

b) WorldCreatorStudio_Core

定义整个软件平台的核心算法、核心类，并为算法功能模块定义基本的开发接口，为整个平台的基础和各个组件联通的纽带。

c) WorldCreatorStudio_Resouses

存储 WorldCreatorStudio 所使用到的资源。

d) MiRaIUIProject

该模块为 WPF 框架的主题项目，基于 WPF 控件定义了一套视觉效果，现已独立为单独的项目，并已通过 MIT 协议开源。开原地址：
<https://github.com/ijrys/MiRaIUIProject>

该模块为 WorldCreatorStudio 模块提供 WPF 程序的 UI 样式。

e) [FE]FrontEndFactorys.RandomTend

前端工厂相关算法的实现。随机趋势化算法核心代码。

f) [BE]MiRaI.BE.AM.SingleValue

后端工厂空气运动模拟相关算法的实现。提供一个快速的、简单的实现，方便必要的需求和软件整体框架测试。

g) [BE]MiRaI.BE.RM.SingleValue

后端工厂降水运动模拟相关算法的实现。提供一个快速的、简单的实现，方便必要的需求和软件整体框架测试。

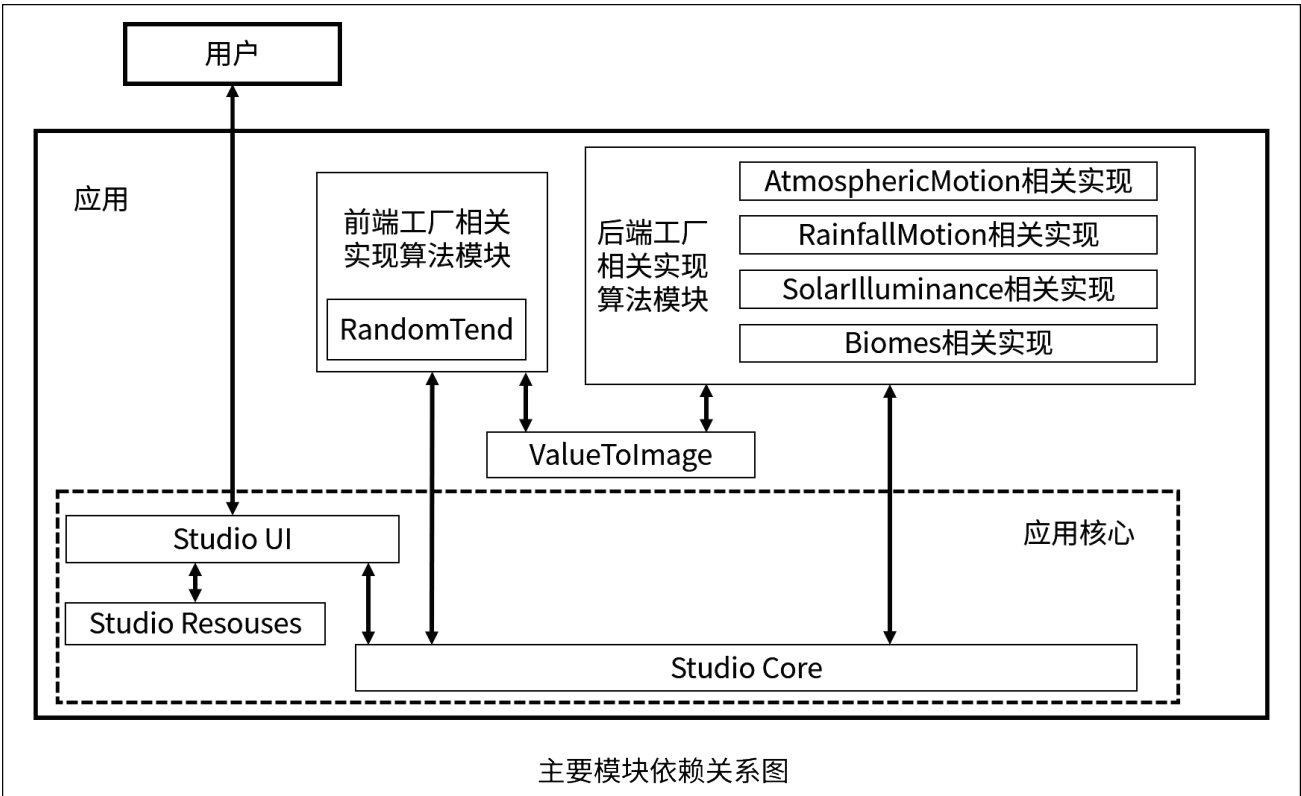
h) [BE]MiRaI.BE.SI.QuickCalculating

后端工厂光照强度模拟相关算法的实现。提供一个快速的、简单的实现，方便必要的需求和软件整体框架测试。

i) [BE]MiRaI.BE.BI.QuickCalc

后端工厂生物群模拟相关算法的实现。提供一个快速的、简单的实现，方便必要的需求和软件整体框架测试。

2. 模块依赖关系图



八、WorldCreatorStudio_Core 模块功能及重要模块细则

1. 子命名空间或逻辑文件夹功能描述

- ListNode

逻辑节点文件夹，内部元素直接存在于 WorldCreatorStudio_Core 命名空间下。逻辑节点类型，提供主功能节点和逻辑节点接口。

- MapCreator

FrontendFactory 的 Creator 相关基类的定义。

- FrontendNode

用于定义 FrontendFactory 的相关节点。

- BackendNode

BackendFactory 的逻辑节点的相关定义，内包含 AtmosphericMotion、RainfallMotion、SolarIlluminance、Biomes 四个文件夹，用以区分四个过程的节点。

- Resouses

资源节点和资源管理节点的定义。

- StoreRoom

应用储藏室，存储各种注册到应用中的数据。

- Tools

各种工具类。

- Exceptions

用于定义所有应用可能用到的异常类型。

2. `[I]ListNode/IWorkLogicNodeAble`

表示在程序中的逻辑节点，用以搭建层级逻辑关系。

继承：INotifyPropertyChanged

a) 字段和属性

名称	类型	可赋值	可获取	默认值	描述
Work	Work	N/A	T		获取节点所在的工作
ShowPanel	ControlTemplate	N/A	T		获取节点可展示的面板
NodeName	string	N/A	T		获取节点在工程树形图中的展示名称
Icon	ImageSource	N/A	T		获取节点在工程树形图中的展示图标
Childrens	ObservableCollection<IWorkLogicNodeAble>	N/A	T		获取节点的子节点
Changed	bool	N/A	T		获取节点是否有值发生了改变

b) 事件

● NodeValueChanged

`event NodeValueChangedEventType NodeValueChanged`

节点值发生改变事件，用于通知需要保存

c) 方法

● XmlNode

`XmlElement XmlNode(XmlDocument xmlDocument, bool save = false)`

获取节点的 XML 节点

xmlDocument	XML 节点所在文档的根节点
save	是否为保存动作

d) 主要节点规划

● ListNode/Project

表示一个工程

对于工程文件的存储信息详见[工程文件格式及存储结构](#)=>[工程文件格式](#)

继承: IworkLogicNodeAble

- 该节点没有可用于展示的面板。

● ListNode/Work

表示一个工作

继承: IWorkLogicNodeAble, InotifyPropertyChanged

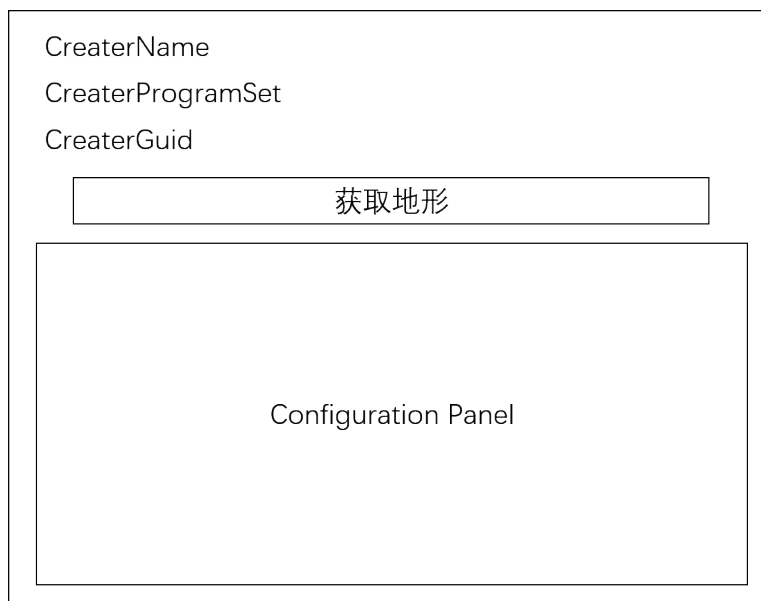
Work 是一个工作流程的最小完整管理节点，用于管理一个工作的所有内容，包括资源和流程。

● ListNode/FrontEndFactory

前端工厂管理节点。

继承: IworkLogicNodeAble

- 展示面板原型设计

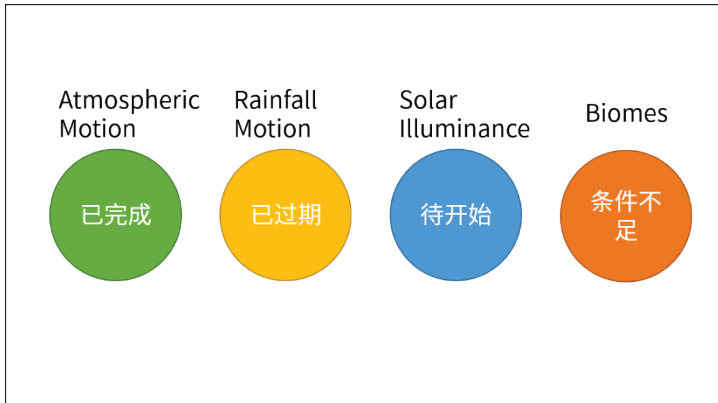


● ListNode/BackEndFactory

后端工厂管理节点。

继承：IworkLogicNodeAble

● 展示面板原型设计



● Resouses/ImageResourceManager

管理工作的全部图片资源，通过 key 进行图片索引和资源获取。

图片的 key 值在工作内唯一，与文件存储名称可以不相同。

● 该节点没有展示面板

3. ListNode/IDataCalculaterAble

后端工厂的模拟及数据生成器接口。

后端工厂相应算法的封装类必须实现该接口。

a) 字段和属性

属性名	属性类型	可赋值	可获取	默认值	描述
CreatorName	string	N/A	True	N/A	获取或设置 Creator 的名称
CreatorProgramSet	string	N/A	True		获取 Creator 所在的 ProgramSet。

					ProgramSet 为 Creator 能力的唯一标识， ProgramSet 相同的 Creator 将认定为实现效果相同
CreatorGuid	Guid	N/A	True		获取 Creator 类的 GUID

b) 事件

● OnProcessingChanged

`event DataCalculatingProcessingEventType OnProcessingChanged`

节点运算过程进度更新时事件

4. ListNode/**IDataCalculaterConfigurationAble**

后端工厂的模拟及数据生成器的设置类接口。

后端工厂相应的类必须实现该接口。

继承：INotifyPropertyChanged

a) 属性

属性名	属性类型	可赋值	可获取	默认值	描述
ShowPanel	ControlTemplate	N/A	True	N/A	获取与 Configuration 相配套的 PanelTemplate

b) 事件

● ValueChanged

```
event NodeValueChangedEventType ValueChanged
```

配置类中值发生改变时事件。

c) 方法

LoadFromXMLNode

```
void LoadFromXMLNode (XmlElement xmlnode)
```

从 XML 节点中加载数据

xmlnode	信息来源节点
---------	--------

XmlNode

```
XmlElement XmlNode (XmlDocument xmlDoc, bool save = false)
```

获取节点的 XML 描述节点。

xmlDocument	XML 节点所在文档的根节点
save	是否为保存动作

5. ListNode/IDataCalculatorFactoryAble<CalcT, ConfigT>

```
where CalcT : IDataCalculatorAble  
where ConfigT:IDataCalculatorConfigurationAble
```

IDataCalculatorAble 和 IDataCalculatorConfigurationAble 的生成工厂。

a) 属性

属性名	属性类型	可赋值	可获取	默认值	描述
DisplayName	string	N/A	True	N/A	数据生成及模拟器的展示名称

DisplayType	string	N/A	True		数据生成及模拟器的展示时的所属类型
CalclaterProgramSet	string	N/A	True		数据生成及模拟器的程序集名称
CalclaterGuid	Guid	N/A	True		数据生成及模拟器的 GUID

b) 方法

● GetACalclater

```
CalcT GetACalclater ()
```

获取一个 Calclater

● GetAConfiguration

```
ConfigT GetAConfiguration ()
```

获取一个适用的 Configuration

九、WorldCreatorStudio 模块功能细则

1. [static] Commands

Studio 使用到的自定义命令

● 属性

声明	描述
<code>public static RoutedUICommand NewWork</code>	命令 · 新建工作
<code>public static RoutedUICommand NewProject</code>	命令 · 新建工程

2. [static]Resouses/StoreRoom

存储 Studio 需要用到的一些资源，为前端绑定提供便利。

3. [static]Resouses/Icons

提供所有用到的的图标资源。

资源存储在 WorldCreatorStudio_Resouses 模块中。

4. [WPF]Resouses/ControlTemplates

[WPF]Resouses/Theme

MiRaUIProject 模块提供的内容

5. [WPF]Windows/MainWindow

主功能窗体

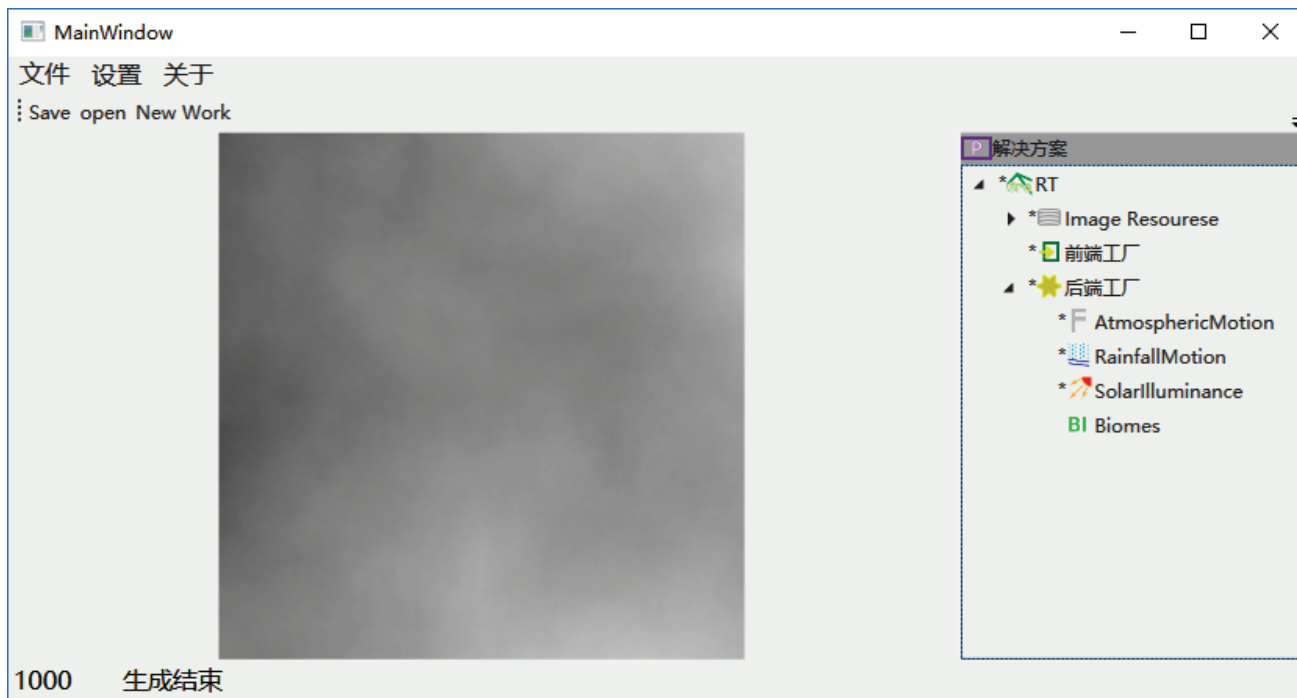
继承：Window

a) UI 设计

● 原型设计图



● 最终产品图



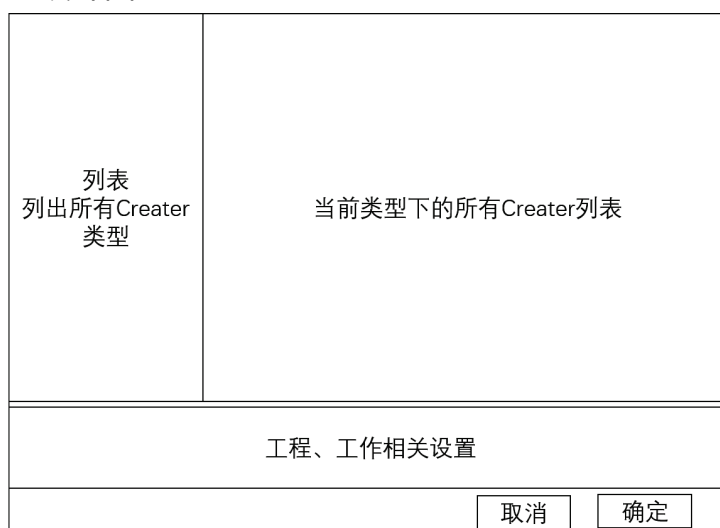
6. [WPF] Windows/NewProject

新建工程、工作窗体 用于选择新建的工程、工作的类型，确定名称和存储位置等

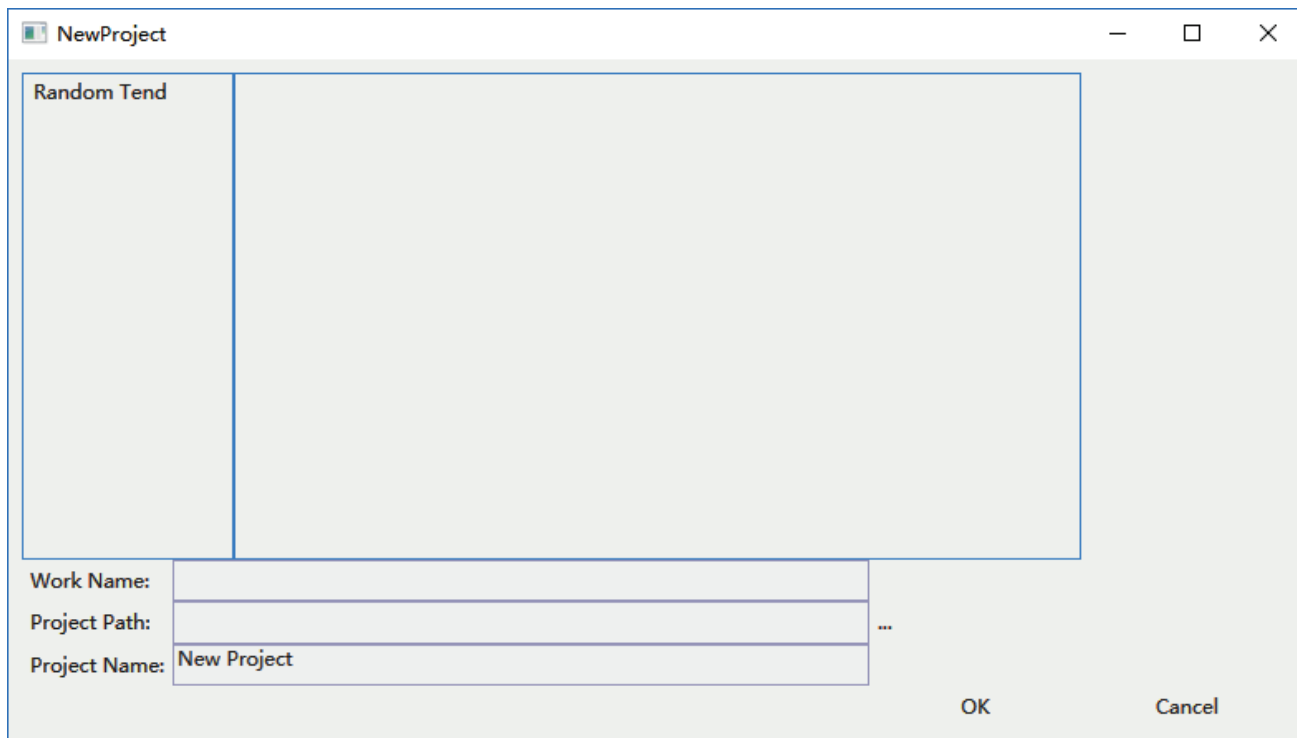
- 继承：Window

a) UI 设计

- 原型设计图



- 最终产品图



十、BackEndFactorys/AtmosphericMotion/MiRaI.BE.AM .SingleValue 模块功能细则

1. SingleValue

使用单值快速填充的空气运动模拟器。

继承：IatmosphericMotionCalclaterAble

- 基本算法：

使用用户设定的值填充所有区域，实现思路为双层 for 循环。

2. SingleValueConfig

记录 SingleValue 进行模拟所需要的配置数据

继承：IAtmosphericMotionConfigAble

3. SingleValueFactory

用于生成 SingleValue 和 SingleValueConfig 的工厂类。

- 配置面板

原型设计：

↖	↑	↗	风力： <input type="text"/>
←	●	→	
↙	↓	↘	

实现效果：

↖	↑	↗	风力 <input type="text"/> 0
←	*	→	
↙	↓	↘	

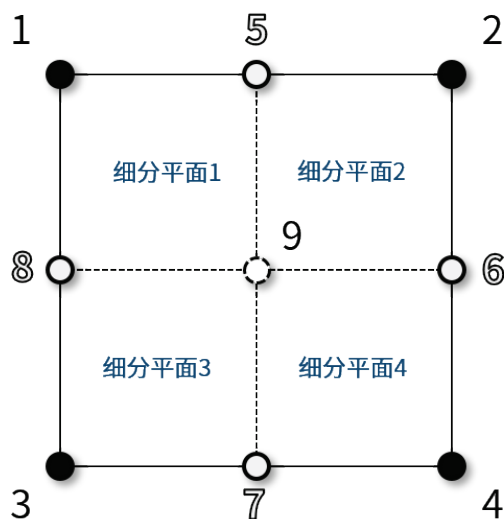
十一、FrontEndFactorys/RandomTend 模块功能细则

1. RandomTendCreator

随机趋势化地形生成算法

继承：MapCreator

- 算法基本原理



0. 如图，对区域的角点（点 1、2、3、4）进行随机，获得区域的大体高度趋势。

1. 对边上中点（点 5、6、7、8）进行高度值计算，方式为边上角点的平均值加随机值，即 $[5] = ([1] + [2]) / 2 + \text{random}$ ，6、7、8 同理。

2. 对中心点利用于边上中点类似的方式确定高度，公式为 $[9] = ([5] + [6] + [7] + [8]) / 4 + \text{random}$ 。

这样一个平面会被细分为四个平面，每个平面再递归地进行步骤 1、2，直到细分区间达到要求。

因为递归算法会有数据初始化问题【如递归计算平面 1 时要先确定确定点 1、5、8、9 的值，递归平面 2 时要先确定点 5、2、9、6 的值，对于点 5、9 来说计算顺序并不容易确定】，故采用循环方式逐层计算。

2. RTConfiguration

RandomTendCreator 的配置类型

继承：Configuration

● 配置面板

原型设计

地图宽度: } 不可修改的值,
 地图高度: 通过下方设定计算

随机值:

块大小:

横向块数量:

纵向块数量:

实际效果

地图宽度	33
地图高度	33
随机值	2019
块大小	5
横向块数量	1
纵向块数量	1

3. RandomTendCreaterFactory

用于创建、获取相应的 MapCreater 和 Configuration 的工厂类

继承: MapCreaterFactory

十二、BackEndFactorys/RainfallMotion/MiRaI.BE.RM.

SingleValue 模块功能细则

1. SingleValue

使用单值快速填充的降水运动模拟器。

继承: IrainfallMotionCalculaterAble

- 基本算法

对于降水强度，按照用户设定填充所有区域

水深计算：若地形高度低于设定的水平面值，水深为水平面高度- 地形高度，
 否则水深为 0。

根据水深设置水域类型。

2. SingleValueConfig

记录 SingleValue 进行模拟所需要的配置数据

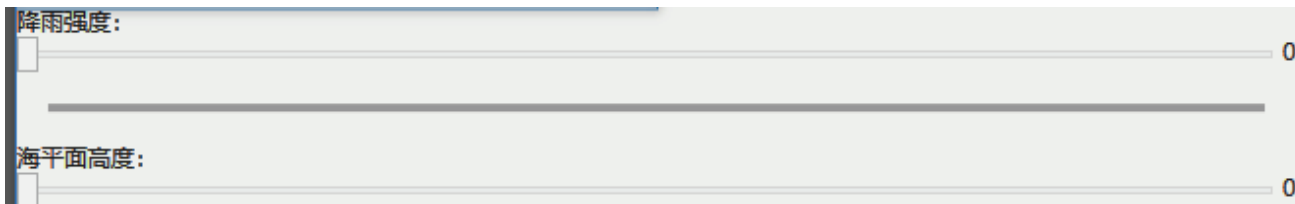
继承: IRainfallMotionConfigAble

- 配置面板

原型设计:

降雨强度:	<input type="text"/>	▲▼
海平面高度:	<input type="text"/>	▲▼

实现效果:



3. SingleValueFactory

用于创建、获取相应的 Calculator 和 Config 的工厂类

继承: IRainfallMotionCalculatorFactoryAble

十三、BackEndFactorys/SolarIlluminance/MiRaI.BE.SI.

QuickCalculating 模块功能细则

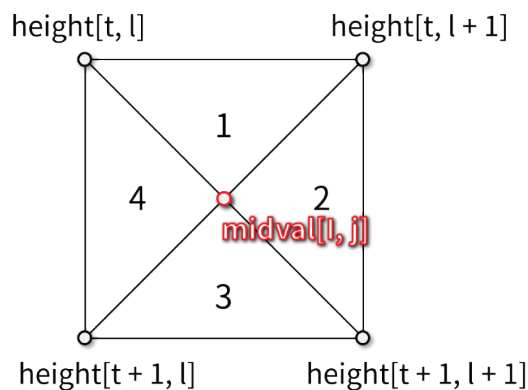
1. QuickCalculating

利用法线快速模拟光照强度模拟器。

继承: ISolarIlluminanceCalculatorAble

- 基本算法原理

对每个区域计算中点值（四角高度的平均值），将区域分为 4 块，分别计算四个区域的强度值，取强度值的平均值做最终结果。



每个区域的强度计算使用直线与平面夹角的计算方式，取夹角的 \cos 值即可得到 $[0,1]$ 之间的值，再乘 255 得出结果。

计算过程中使用 double 作为过程变量类型。

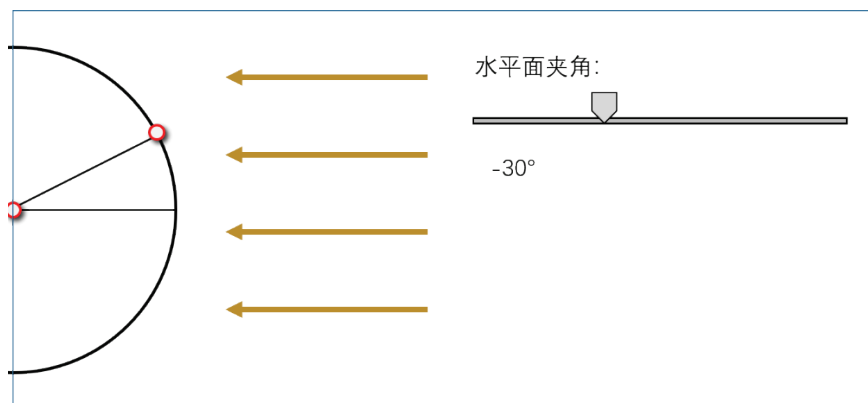
2. QuickCalculatingConfig

记录 QuickCalculating 进行模拟所需要的配置数据

继承：IsolarIlluminanceConfigAble

● 配置面板

原型设计：



3. QuickCalculatingFactory

用于创建、获取相应的 Calculator 和 Config 的工厂类

继承：IRainfallMotionCalculatorFactoryAble

十四、BackEndFactorys/Biomes/MiRal.BE.BI.QuickCalc 模块功能细则

1. QuickCalc

根据已有快速模拟光照强度模拟器。

继承：IBiomesCalclaterAble

温度指数计算方式如下：

$$\text{templIndex} = \text{SIRes}[i,j] - \text{height}[i,j] * \text{scale}$$

湿度指数计算方式如下：

$$\text{RMRes}[i,j].\text{RainfallIntensity} * \text{scale} - \text{templIndex}$$

温度、降水与结果对应表：

温度指数	气候类型	湿度指数	生物群类型	枚举值
(-∞,10)	寒带	(-∞,0)	寒带沙漠	Desert
		[0,30)	冰雪苔原	Tundra
		[30,+∞)	针叶林	ConiferousForest
[10,25)	温带	(-∞,5)	沙漠	Desert
		[5,100)	温和草原	TemperateGrassy
		[100,+∞)	温和森林	TemperateForest
[25,+∞)	热带	(-∞,10)	沙漠	Desert
		[10,90)	炎热草原	TropicalSavanna
		[90,180)	炎热旱林	ThornForest
		[180,+∞)	雨林	TropicalRainforest

2. QuickCalcConfig

记录 QuickCalc 进行模拟所需要的配置数据

继承：IbiomesConfigAble

3. QuickCalcFactory

用于创建、获取相应的 Calculator 和 Config 的工厂类

继承：IbiomesCalculatorFactoryAble

十五、ValueToImage 模块功能细则

1. 关于常见图片格式的说明

图片均使用点阵图操作，资源存储时使用无损压缩的 png 格式。为减少内存占用，产生的数据内存使用量尽可能小，即每像素占用的比特数尽可能少。以下的格式指像素格式。

名称	像素 比特 数	图片类型	描述
BlackWhite	1	黑白	二值化图片，仅可表示黑、白两色
Gray2	2	灰度	灰度图像，随着每像素比特数增加可表示的中间色越多。
Gray4	4		
Gray8	8		
Index1	1	索引颜色	索引格式。可索引 2 种颜色。
Index2	2		索引格式。可索引 4 种颜色
Index4	4		索引格式。可索引 16 种颜色
Index8	8		索引格式。可索引 256 种颜色
BGR32	32	真彩色	按照 B、G、R、A 三通道存储的图片，每通道 256 种值。BGR32 会忽略 A 通道。因该软件需求不大，故不建议使用
BGRA32	32	真彩色，带透明通道	

			该种格式。
--	--	--	-------

2. [static]ValueToGrayImage

将前端工厂生成的高度数据转换为可视的灰度图，输出为 Gray8 格式或是 Index8 格式的图片。

实现要求：通过设定 minValue、maxValue、minGray、maxGray 实现将 map 中的值映射到目标灰度值中。其中 map 中 value≤minValue 的映射为 minGray，value≥maxValue 的映射为 maxGray。

计算原理公式为：

$$\frac{\text{resGray} - \text{minGray}}{\text{maxGray} - \text{minGray}} = \frac{\text{height} - \text{minValue}}{\text{maxValue} - \text{minValue}}$$

得

$$\text{resGray} = \text{minGray} + (\text{maxGray} - \text{minGray}) * (\text{height} - \text{minValue}) / (\text{maxValue} - \text{minValue})$$

3. [static]ValueToColorImage

将高度数据转换为可视的彩色图，用于不同高度的较精确的检查。

输出为 Index8 格式的图片。

数据计算方法见 ValueToGrayImage 的原理。

4. [static]AtmosphericMotionToImage

配合 AtmosphericMotion 的可视化值转换器。

数据分为若干边长为 16*16 的块，每个块内求平均值，展示每个块的平均值即可。

对每个块内的数据进行累加统计，得出 9 个方向的风力占比。按照对向相消的法则获得空气在两个正交的方向上的速度分量，进而算出块的风向和风力。

5. [static]RainfallMotionToImage

配合 RainfallMotion 的可视化值转换器。

因为 RainfallMotion 计算器生成的数据的三个分量在表示时并无相关性，故该部分需要分别实现对三个分量的可视化转换。

降雨强度的结果输出建议为 Gray8 格式、Index8 格式。

区域类型图输出建议为 Index4 格式。

水深图输出建议为 Gray8 格式、Index8 格式。

6. SolarIlluminance

配合 SolarIlluminance 的可视化值转换器。

输出建议为 Gray8 格式或是 Index8 格式。

7. BiomesToImage

配合 BiomesToImage 的可视化值转换器。

输出建议为 Index8 格式。

十六、WorldCreatorStudio_Resouses

模块功能及重要模块细则

该模块存储了程序的图标等二进制资源，为纯粹的资源模块。

十七、ValueTo3DModel 块功能及重要模块细则

该模块提供将生成的结果转换为 3D 模型的方法。

模型使用标准的 fbx 的实现，使用三角面。

1. [static] HeightToModel

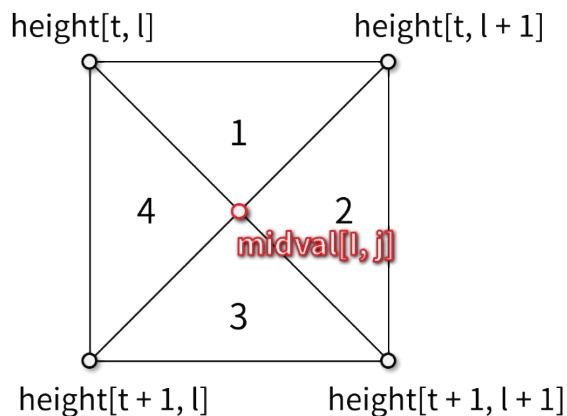
b) GetModelWithSubdivide

```
public static MeshGeometry3D GetModelWithSubdivide (int[,]  
heightmap)
```

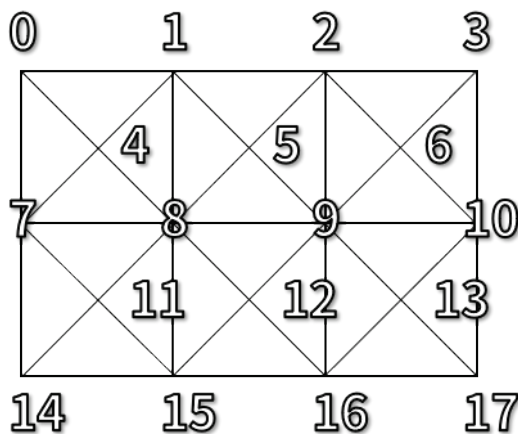
将高度数据转换为 3D 模型，并带有细分。

- 基本思路

如下图对每一个块进行一次中间值计算，将一个正方形区域分为四个三角形。



存储时点按行相间隔顺序存储，如下图所示



先规定如下数值：

width：图的宽。

对于 height[t,l]可有如下计算公式：

$$id = (width * 2 - 1) * t + l$$

对于 midval[t,l]可有如下计算公式：

$$id = (width * 2 - 1) * t + width + l$$

十八、软件测试内容概要

1. 新建工程

a) 基础测试

● 新建工程

Work Name：RTTest

Project Name：New Project

- 确保过程中没有程序崩溃。

● 新建地形

以默认值创建地形。

- 确保过程中没有程序崩溃。
- 检查生成过程是否正常
- 检查图片资源是否正确生成

● 保存工程

将工程保存到磁盘中。

- 确保过程无崩溃
- 确保工程文件、图片资源及数据资源已存储在磁盘中

● 打开工程

检查工程是否完好。

- 图片资源是否完整
- 前端工厂生成器是否正确选择
- 前端工厂生成配置是否正确

● 配置保存

调整生成设置，生成地形，保存工程，打开工程，检查工程是否完好。

- 确认配置是否正确存储，并正确加载

● 多工作新建保存测试

新建工作

Work Name 为 RTTest2

修改 RTTest2 的生成配置，保存工程。

- 确保新建界面的 Project Path 和 Project Name 正确加载，并不可更改。
- 确保过程没有异常出现

● 多工作打开测试

打开工程

- 确保过程中没有崩溃
- 确认工作名正确
- 确认配置正确

b) 后端工厂测试

打开工程 New Project

● AMNode 测试

选择相应节点，选择默认运算器，以默认配置生成数据。

- 确保过程中没有产生崩溃
- 检查后端工厂节点中相应节点状态是否正确
- 检查图片资源正确生成
- 检查可视化面板功能正确

保存工程

- 确保过程无异常
- 确保相关资源被保存到磁盘

打开工程

- 确保过程无异常
- 确认数据被正确加载
- 确认配置被正确加载

2. RMNode 测试、SINode 测试、BINode 测试类比 AMNode 测试

十九、总结汇报

1. 本版出现的问题

使用 abstract class 实现 oop 的开发，但因为 abstract class 天然的限制，导致代码自由度低，实现困难。

应用前后端使用数据绑定进行数据管理，但是部分功能前后端耦合程度依旧过高。

部分功能设计时耦合太高，降低了自由度，特别是在数据可视化相关的功

能上，用户没有可选择的方式。

主要功能基本基本实现，但许多功能依旧欠缺。

2. 下版改进

● 面相全异步的设计

所有计算工作以异步形式完成。为实现异步中的程序前后端同步，需要对框架进行重新设计。

● 以 interface 取代 abstract class

使用 c#8 作为主要的交互逻辑开发语言，使用接口和接口方法实现类的继承关系，使代码逻辑更清晰、更自由。

● 更自由的工作流程

将可视化模块与生成器解除绑定，生成器将不再参与可视化过程，由用户选择可视化转换器进行值转换。

工作将不再与地图生成器绑定，之后将可以自由更换工作的所有设定。

● 更强大的内容组织管理

使用更强大的 uri 取代现行的 path，提供更丰富、强大的数据管理方案。

WorldCreator 产品文档

文档版本：1.2

软件版本：1.2

一、 产品简介	2
1. 功能介绍及使用方法	2
二、 评测报告	5
三、 安装手册	6
1. 硬件要求	6
2. 系统及软件环境要求	6
3. 安装方法	7

一、 产品简介

本软件为地形生成算法及相关环境模拟算法实验、调用平台，目的为相关算法学习者和实现者提供一个可用的平台，并为相关需求者提供方便，如游戏、影视中远景山脉的建模。

本软件使用 MIT 协议开源，可在协议范围内使用改造，方便相关研究者对研究内容的整合。

1. 功能介绍及使用方法

软件采用核心程序加插件的形式实现功能整合。

软件实现的功能有工程的创建、打开与保存，工作的添加，地形生成，空气运动模拟，降水模拟，光照模拟，生物群模拟。

a) 工程创建、打开与保存：

工具栏中单击“文件”->“新建工程”，即可打开工程新建窗口。

在窗口中选择工作类型，并填写工作名、工程路径和工程名，点击“确定”按钮即可创建新工程。

工具栏中单击“文件”->“打开工程”，在选择窗口中选择工程文件，点击确定，即可打开已有的工程。

工具栏中单击“文件”->“保存”，即可将工程保存到磁盘上。

b) 添加工作

可以为一个工程添加多个工作，但是在没有打开工程情况下无法添加工作。

工具栏中单击“文件”->“新建工作”，在弹出的界面中选择工作类型，并填入工作名称，在核对下方的工程信息无误后点击“确定”按钮，即可为当前工程添加一个工作。

c) 生成地形

在主界面右侧的目录树中选择目标工作，双击相应的“前端工厂”节点，将会在主界面展示相应的配置界面。

在调整好参数后，点击“获取地形”按钮，即可开始地形生成过程。

d) 空气运动模拟

该功能需要地形数据，只能在生成地形步骤之后进行。

展开“后端工厂”节点，双击 AtmosphereMotion 节点，主界面将展示相应配置面板。

在面板中选择合适的运算器，下方会根据不同运算器展示不同配置面板。在配置面板中选定配置，点击上方的“计算”按钮，即可开始运算。

e) 降水模拟

该功能需要地形数据及空气运动模拟数据，只能在相关步骤均完成之后进行。

展开“后端工厂”节点，双击 RainfallMotion 节点，主界面将展示相应配置面板。

在面板中选择合适的运算器，下方会根据不同运算器展示不同配置面板。在配置面板中选定配置，点击上方的“计算”按钮，即可开始运算。

f) 光照模拟

该功能需要地形数据、空气运动模拟及降水模拟数据，只能在相关步骤均完成之后进行。

展开“后端工厂”节点，双击 SolarIlluminance 节点，主界面将展示相应配置面板。

在面板中选择合适的运算器，下方会根据不同运算器展示不同配置面板。在配置面板中选定配置，点击上方的“计算”按钮，即可开始运算。

g) 生物群模拟

该功能需要地形数据、空气运动模拟、降水模拟及光照模拟数据，只能在相关步骤均完成之后进行。

展开“后端工厂”节点，双击 Biomes 节点，主界面将展示相应配置面板。

在面板中选择合适的运算器，下方会根据不同运算器展示不同配置面板。在配置面板中选定配置，点击上方的“计算”按钮，即可开始运算。

二、 评测报告

a) 评测目的

检查软件可用性及稳定性，确保算法正确无误。

b) 评测范围

- 软件使用流程
- 数据处理算法正确性
- UX 设计

c) 评测环境

详见安装手册系统及软件环境要求

d) 实测数据

见软件开发文档相应内容及软件测试文档

e) 性能表现

性能在设计要求范围内

三、 安装手册

1. 硬件要求

a) 最低硬件要求

- 显示器 800 * 600 及更大分辨率
- 内存：剩余 1G 以上可用的内存空间
- 磁盘：剩余 2G 以上可用的磁盘空间，磁盘可读写

b) 推荐硬件要求

- 显示器 1366 * 768 及更大分辨率
- 内存：剩余 4G 以上可用的内存空间
- 磁盘：剩余 4G 以上可用的磁盘空间，磁盘可读写

2. 系统及软件环境要求

- .NET Framework 4.7.2 及兼容版本
- Windows 10 1607 及后支持.NET Framework 4.7.2 及 WPF 的操作

系统

3. 安装方法

运行 setup.exe，运行安装程序。

按程序指引安装。安装过程会自动检测并安装需要的环境 Microsoft .NET Framework 4.7.2 (x86 和 x64)，安装过程中提示 Microsoft .NET Framework 4.7.2 Setup 的权限提示（如下图），请点是。



待安装程序安装完毕后即可在开始菜单中找到已安装的程序。

若已安装旧版本，需先卸载旧版本后再安装新版本。

附件 3:

青岛大学本科毕业论文评分表(指导教师用)

学生姓名:

学院:

专业班级:

评分内容	具体要求	分值	得分
选题质量	选题的理论意义或实用价值；难易程度及工作量	20	
学习态度	整个过程的工作态度，是否遵守纪律，作风严谨务实，按指导教师要求圆满完成规定的任务	10	
研究方案	能独立查阅文献资料及从事其他形式的调研，能较好地理解课题任务并制定合理的研究方案，有分析整理各类信息并从中获取新知识的能力	20	
研究创新能力	论文研究过程中综合运用系统的专业知识和基本理论，善于发现和解决实际问题，具有独立的科研创新和思考能力。	30	
写作水平	概念清晰、分析严谨；数据真实、结论正确；用语、书写符合格式规范；摘要及外文资料翻译准确，文字流畅，符合规定内容及字数要求。	20	
得分合计			
具体评阅及修改意见：			
<div style="height: 150px; border: 1px solid black;"></div>			
是否同意答辩：是（ ） 否（ ）			
指导教师签名： <div style="float: right;">年 月 日</div>			

青岛大学本科毕业论文评分表（评阅教师用）

学生姓名：学院：专业班级：

评分内容	具体要求	分值	得分
选题质量	选题的理论意义或实用价值；难易程度及工作量	20	
文献综述	对本研究领域文献资料掌握的程度，对所要解决的问题及意义是否论述清楚	20	
研究成果	是否综合系统的运用专业知识和理论原理；是否具有新见解、新思路；是否具有独立的科研创新和思考能力。	40	
撰写规范性	结构严谨；文字通顺，用语、书写符合格式规范；外文资料翻译准确，文字流畅，符合规定内容及字数要求。	20	
得分合计			
具体评阅及修改意见：			
是否同意答辩：是（ ） 否（ ）			
评阅教师签名：年 月 日			

青岛大学本科毕业论文评分表（答辩委员会用）

学生姓名:

学院:

专业班级:

评分内容	具体要求	分值	得分
答辩	能够正确回答所提出的问题，基本概念清楚，有理论根据。	40	
自述总结	思路清晰，语言表达准确，概念和论点清楚，概括性和逻辑性强。	30	
创新	有独特见解，有一定理论意义和社会价值。	20	
资料	资料齐全，符合《青岛大学毕业论文（设计）基本规范要求》	10	
得分合计			
意见及建议：			
是否达到毕业并授予学士学位的水平：是（ ） 否（ ）			
答辩委员会成员签名：			
答辩委员会主任签名：			
年 月 日			

青岛大学本科毕业设计评分表（指导教师用）

学生姓名：学院：专业班级：

评分内容	具体要求	分值	得分
选题质量	设计的难易程度及工作量	20	
学习态度	整个过程的工作态度，是否遵守纪律，作风严谨务实，按指导教师要求圆满完成规定的任务	10	
设计方案	能较好地理解课题任务，通过多种方案论证，选择制定出合理、新颖和可行的设计方案	20	
实践创新能力	课题设计过程中，综合运用系统的专业基本理论，掌握多种技术路线和实现路径，有重要改进；表现出较强的动手能力和独立工作能力。	30	
设计成果及质量	数据分析、实验结论、绘图（表）等正确合理；设计成果（包括实物、作品或软件）具有相对完整性和一定的实用价值	20	
得分合计			
具体评阅及修改意见：			
是否同意答辩：是（ ） 否（ ）			
指导教师签名：		年 月 日	

青岛大学本科毕业设计评分表（评阅教师用）

学生姓名:

学院:

专业班级:

评分内容	具体要求	分值	得分
设计方案	设计方案合理，有一定创新性或艺术性	30	
技术路线或实现手段	表现手段规范、完整，技术路线较先进	30	
成 果	实验结论、论证分析或设计计算正确合理，取得一定的成果，具有一定的应用价值。	40	
得分合计			
具体评阅及修改意见：			
<div style="height: 200px;"></div>			
是否同意答辩：是（ ） 否（ ）			
评阅教师签名： <div style="float: right; margin-top: -60px;">年 月 日</div>			

青岛大学本科毕业设计评分表（答辩委员会用）

学生姓名： 学院： 专业班级：

评分内容	具体要求	分值	得分
自述总结	思路清晰，语言表达准确，能概括和突出设计的主题和成果	30	
创新	有独特的设计理念和一定的创造性，技术手段先进	20	
答辩	能够正确回答所提出的问题，数据分析论证等科学合理；设计成果（包括实物、作品或软件）具有相对完整性和一定的实用价值	50	
合计得分			
意见及建议：			
是否达到毕业并授予学士学位的水平：是（ ） 否（ ）			
答辩委员会成员签名：			
答辩委员会主任签名：			
年 月 日			