

《随机地形生成演示》

——Qt 大作业小组报告

组名：不知道取啥名队

组长：陆昶安 组员：张天齐，范潇帷

一、项目简介

本项目是三人小组“不知道取啥名队”在课程“程序设计实习”完成的大作业。本项目的灵感来源于 Minecraft 等游戏的随机地形生成方法，并通过 Qt 平台和 C 语言初步实现了最基础的环境构建工作。本项目基于 C/C++ 实现，使用少量 QML 代码制作 3D 引擎以展现 3D 效果。本项目基于 Qt 开源框架，实现了基于 Value Noise、Perlin Noise 等算法的随机地形生成，并提供 bmp 格式的高线图、3D 预览图等多种演示形式。

地址：<https://github.com/Decmofofs/Course-Projects-For-Learning-in-PKU/tree/main/Practice-of-Programming-in-C%26C%2B%2B>

二、程序功能介绍

（一）地形生成算法介绍

为实现随机地形生成，我们需要基于合适的物理规则进行地形数据的随机生成与平滑化。由于时间与技术限制，我们最终提供了以下四种地形生成算法，每一种都各有特色。下面是算法介绍：

1. Value Noise 算法

Value Noise 算法（值噪声）是一种用于生成或模拟自然现象的噪声算法，常用于计算机图形学中生成纹理、地形等。它是一种基于格点（网格的顶点）的噪声生成方法，通过在每个格点上随机分配值，然后对这些值进行插值来生成整个网格上的噪声值。这种方法可以产生平滑的、连续的噪声图案，适用于创建多种自然效果，如云彩、山脉或其他纹理。

2. Perlin Noise 算法

Perlin Noise 算法（柏林噪声）是一种广泛使用的程序生成噪声算法，由 Ken Perlin 在 1983 年开发，主要用于计算机图形学中生成自然现象的纹理，如火、烟、

水、云以及各种材质表面等。与 Value Noise 相比，Perlin Noise 生成的噪声更加自然和连续，没有那么明显的方向性，因此它在生成自然景观和纹理方面特别受欢迎。

3. Midpoint Displacement 算法

Midpoint Displacement 算法（中点位移算法）是一种用于生成分形地形或其他自然现象的算法，特别适用于创建山脉或岛屿等自然景观的高度图.这种算法通过递归地细分线段或平面，并在每次细分时对中点进行随机位移，从而生成具有自然变化的图案。中点位移算法是一种简单而强大的技术，能够以相对低的计算成本生成复杂的自然形态。

4. Hydraulic Erosion 算法

Hydraulic Erosion 算法（流水侵蚀算法）模拟自然界中水流对地形的侵蚀作用，用于生成更加真实和详细的地形模拟。这种算法通过模拟雨水的降落、流动、蒸发以及与地面的相互作用，来改变地形的高度，从而创建河流、山谷和其他侵蚀特征。水力侵蚀算法可以使计算机生成的形看起来更加自然和逼真。

（二）展示方式介绍

在获得一系列高度数据后，除了最基础的生成二维图像外，我们借助 QML 代码实现了简单的 3D 视觉效果，并支持放大、缩小、视角变化的操作，方便使用者去观察。

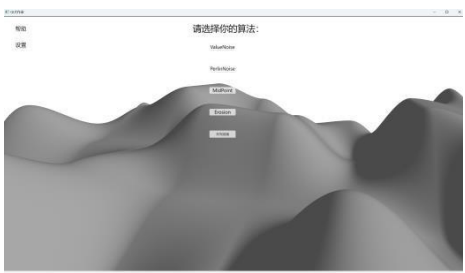
下面是二维图像和 3D 视觉效果的对照图：



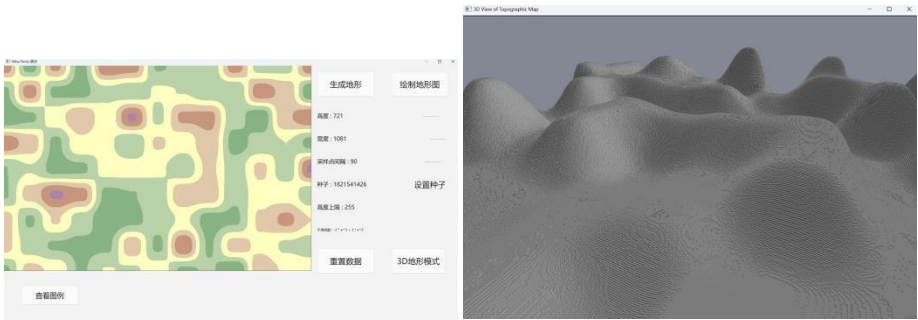
（三）操作说明及四种随机算法事例

运行程序后，首先我们看到的便是起始界面（如下图），这里我们可以按帮助键来获取算法、参数、地形图和 3D 视图的信息，按设置键可以设置网格样式（方

块状网络和平滑曲线）和地形图模式（等高线模式，灰度模式）。

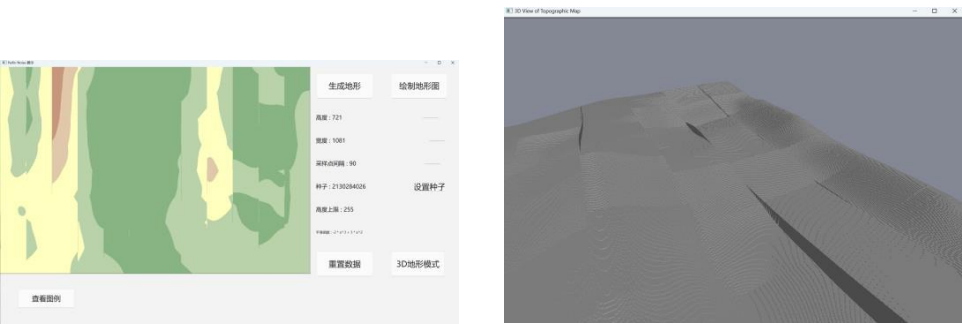


之后，我们可以选择中央四种算法之一，例如先点击ValueNoise，并且点击生成地形后，稍等几秒便会出现如下界面：

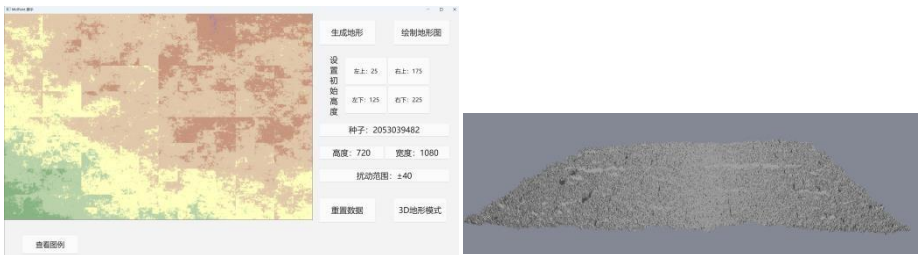


其中，查看图例可以更改图例颜色，拉动滑动条可以修改高度、宽度、采样点间隔和种子，点击 3D 地形模式便可以切换至 3D 模式。

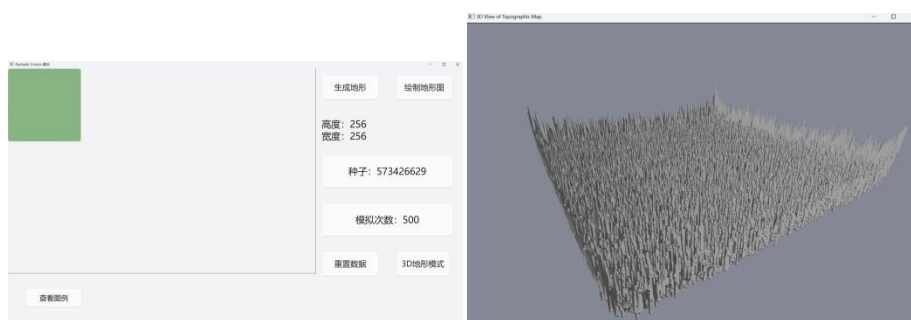
查看 PerlinNoise 的实例如下图，可修改内容与ValueNoise 一致。



查看 Midpoint 的实例如下图，可修改内容为 4 个角的初始高度、种子、高度宽度和扰动范围：



Erosion 的实例如下图，可修改的数据有，高，宽，种子和模拟次数。



三、项目各模块与类设计细节

本项目讲一个又一个小算法分散开来，每个程序 `xxx.cpp` 都配套了说明性文件 `xxx.h`，方便组内成员和助教查看。

项目代码大致可以分为预备功能模块、QTAPP 框架模块、核心算法模块、3D 引擎模块这四个部分。

预备功能模块为后续的代码实现提供了基本的支撑，并且提供了 UI 的搭建方式：图中的 `color_extension` 和 `setcolor` 为高度图的着色代码提供了颜色支持；`math_expand` 中自定义了可用的二维数组，这成为高度图的基本构成方式；`STrandom` 中定义了一种随机算法，这种算法更加适应本项目的需求；`height_reader` 和 `heightgraph` 中定义了项目主要使用的 `heightGraph` 类以及相应的文件读取方式及写入方式；`objwriter` 中，我们实现了 `heightGraph` 向 `bmp` 文件、`obj` 文件（QML 的 3D 引擎所用的一种文件格式）、`mesh` 文件（QT 所能识别的 3D Mesh 文件）的转化，在经过努力后，所有文件均以相对路径的方式，在程序安装目录进行读写。

QTAPP 框架模块为 QT 应用程序提供支持。在代码中，主要由 `mainwindow`，`main`，以及各类 `widget` 文件组成。这些代码构建了 QT 应用程序，并通过 `widget` 文件完成应用程序 UI 的输入向程序的回传。

核心算法模块是最重要的部分。该模块共有四种算法，分别是 Value Noise, Perlin Noise, MidPoint Displacement, Hydraulic Erosion 算法。每种算法包括对应的 `.cpp` 和 `.h` 文件，在 `.cpp`

中给出了算法代码的具体实现。前两种算法包括地图构建、随机取样、函数拟合、写入地图这几个部分；第三种算法包括传入初值、地图构建、中点扰动这几个部分；最后一种算法则包括传入地图、固定次数模拟这两个部分。算法对应的 `widget` 文件在前文已述，在此不再赘述。

3D 引擎模块为项目提供了良好的可视性。这部分主要使用 QML 实现，在 `main.qml` 和 `start.qml` 中，分别实现了算法生成地形的展示、主菜单界面地形的展示。

在类设计方面，我们为每个功能都设计了相应的类，声明放在头文件中。这样，在各文件中调用时，我们只需 `include` 要用的头文件，并使用类型存取符，就可以方便地调用类中的变量和函数。

为了地图的创建、读写顺利，我们设计了 `Vector2` 类并重载了对应的运算符；并实现 `heightGraph` 类作为地图，它的构造函数使用 `Vector2` 类为参数。这样，我们使用一种统一的接口，完成了地图数据的读写。

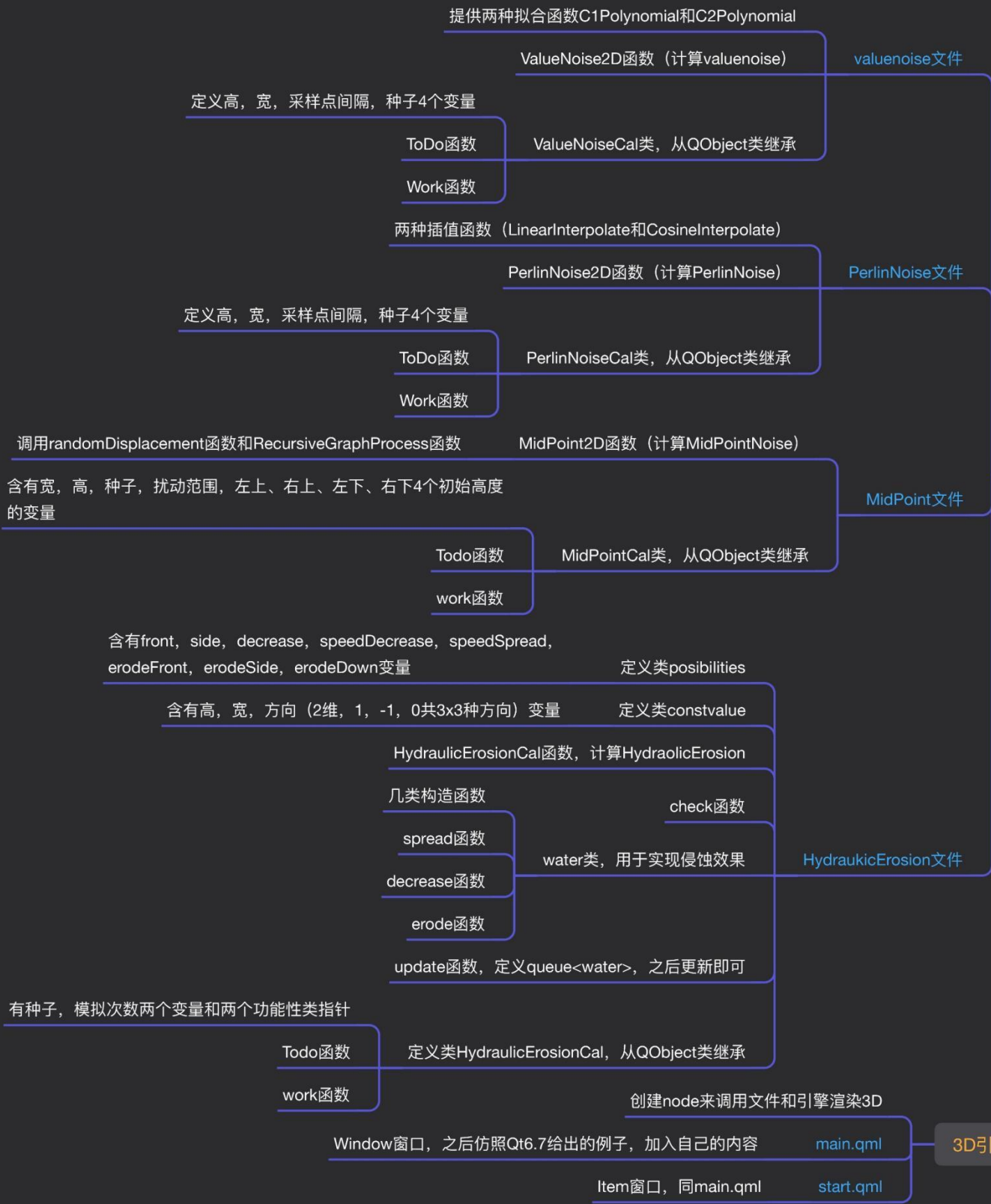
我们定义了 namespace: `STrandom`，以进行随机操作。为了避免与其他随机算法混淆，我们使用了 namespace 来做准确读取。

在算法中，我们为每个算法创建了 `AlgorithmNameCal` 类，它继承自 `QObject` 类，其中存放了具体计算函数，以及与 QtAPP 的 `Thread` 相连接的 `ToDo,Work` 函数，便于调用。特别地，在 `Hydraulic Erosion` 中，我们使用 namespace 来存放超参数。

为了方便文件操作，我们定义了 `objwriter` 类，调用时可以按照需求创建、删除、修改指定格式的文件。

我们还定义了各 `widget` 类，它们同样继承自 `QObject` 类，其中实现了与 QT 应用程序的交互，存放了接口用于数据回传。

具体类设计细节可以借助思维导图查看：

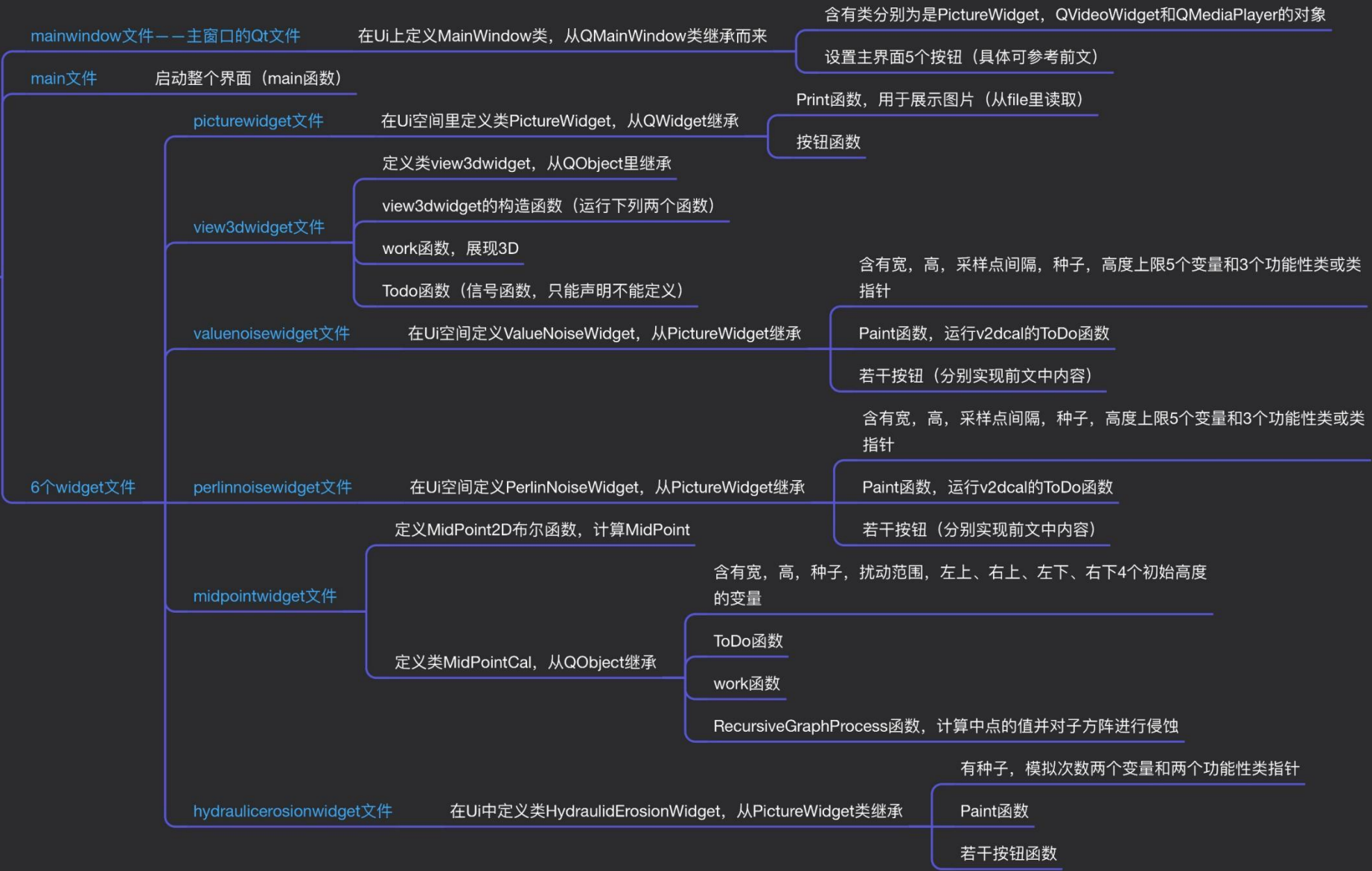


项目代码

预备功能模块



QtAPP框架模块



四、小组成员分工情况

组长陆昶安完成了项目的基本框架和部分算法.他实现了本项目使用的Value Noise 算法、Hydraulic Erosion 算法，完成了基本类的构造、设置了可用的 API、创建了基本的 UI，并实现了 bmp 文件的读写、obj 文件的生成、obj 到 mesh 文件的转换，且使用占项目代码量较少的 QML 语言完成了3D可视化引擎的制作.他还完成了对其他组员代码的校对工作.组长还完成了最后的审阅工作，并录制操作、讲解视频，最后提交项目文件至助教处。

组员张天齐在组长陆昶安的工作的基础上，解决了程序的环境依赖问题，提出了可行、轻量且稳定的项目构建方案，使得项目在各 Windows 计算机可复现。他还完成了 Perlin Noise 算法，MidPoint Displacement 算法的设计、整合，并协助组长陆昶安完成了 Hydraulic Erosion 算法代码向本项目的整合。

组员范潇帷为项目提出了许多宝贵的建议。他完成了 UI 的最终美化，调整了 bmp 地形图的初始参数，并为项目撰写了readme 及 PDF，调整校对了部分语句和代码，使项目呈现出了更好的效果。

五、项目总结与反思

经过 2 个月的努力，我们基本实现了一开始定下的目标，我们完成了 4 种随机地形生成算法，并且提供了多种视图便于使用者查看，以帮助使用者构建地形的空间感，同时，我们也设置了很多参数可以给使用者较大的自由度去随机调整地形。

当然，我们也有很多需要反思的地方，第一点就是分工，范同学考完后仍有课程要上，并且同时负责今年招生工作，所以没有在代码层面多帮助另外两位同学，但在后期上完课结束招生后也努力填补了文案等工作，另外三位同学考完后并不在一起（分散在祖国各地），因此在沟通上也是困难重重。第二点就是环境搭配问题，组内三位同学的环境搭配比较复杂，在开始的一段时间内只能用陆同学的电脑检查代码情况。此外便是项目定位与最后呈现的还是有一些区别，我们一开始的方向并不明确，只是认为生成地形这个方向是可以尝试的，因此在初期有点畏手畏脚，但到了 6 月初，又认为所需要完成的内容太少，于是又加上了很多预定目标，结果到最后，我们还有很多任务并没有完成。

整体说来，我们这次项目完成情况还是差强人意的，再给我们几天，我们一定能够呈现出更好的效果。此外，我们也录制了一个视频，视频介绍了我们完成的一小部分内容，更多内容还需要使用者去自行探索，也希望使用者们给我们提供宝贵的建议！

最后，我们要感谢助教和老师的帮助！非常感谢北大提供了如此好的平台来供我们进行程序设计训练！