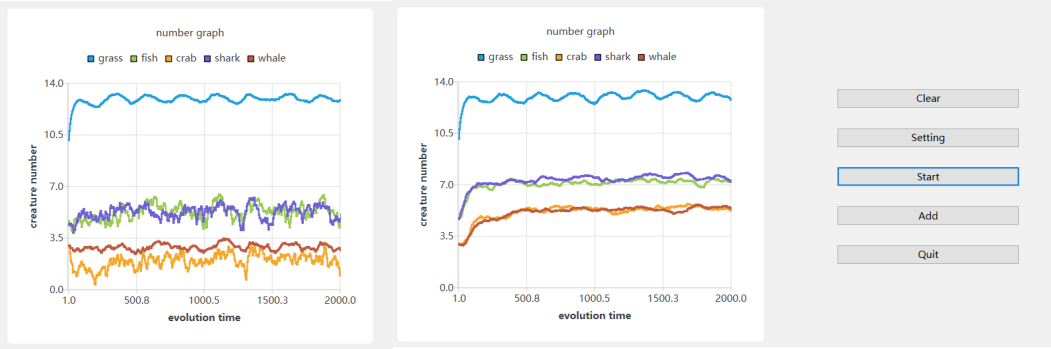
1-作业报告

一、程序功能介绍

我们组基于QT完成一个生态系统的模拟器，以模拟生物演化。具体功能分为个体层面演化和群体层面演化：

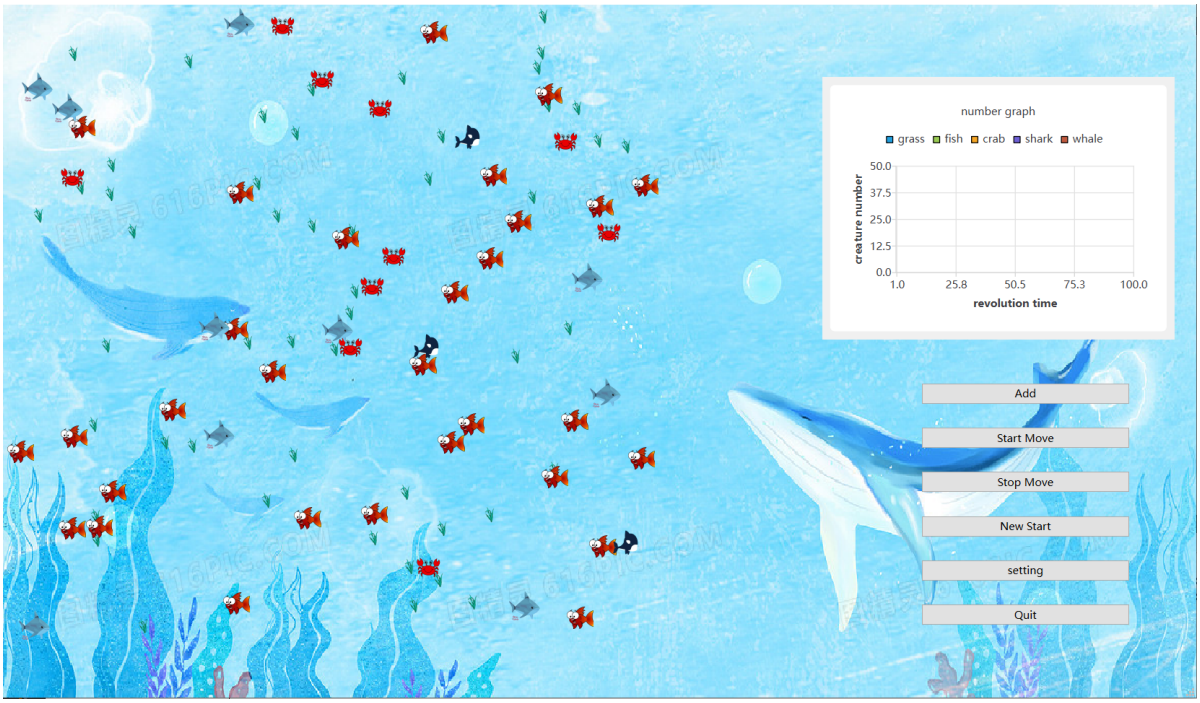
群体层面

通过定义捕食率、出生率、初始种群数量等"超参数"，观察生态系统中种群数量的变化，验证生态学中的一些结论。每一轮演化为50代，在演化过程中也可以随机增加某一个生物的数量。从结果来看，生态系统具有较好的自我调节能力，经过快速的演化后，会达到相对稳定的平衡状态。



个体层面

定义creature 类模拟不同生物，对于每个生物，用一个神经网络模拟生物的行为，遗传方式为遗传神经网络的超参数；每个生物需要捕食食物同时躲避天敌、繁衍后代。个体层面随机性更强，可以更好地模拟生态环境中种群的进化过程。



二、项目各模块与类设计细节

mainwindow类

继承自QMainwindow，个体演化中用于显示窗口，在mainwindow中有一个图表用于记录生物数量随时间的变化，以及6个按钮，分别用于在演化过程中添加某一动物数量，开始演化，暂停演化，重新开始演化，初始状态设置和退出。

我们用一个vector来记录演化中存在的生物，每一轮演化过程中，通过计算不同生物之间距离来确定是否捕食，如果某一生物一直没有捕捉到事物，则会死亡

creature类

继承自QWidget，是个体演化中最重要的一个类，记录了每个生物的年龄、种类、速度等信息

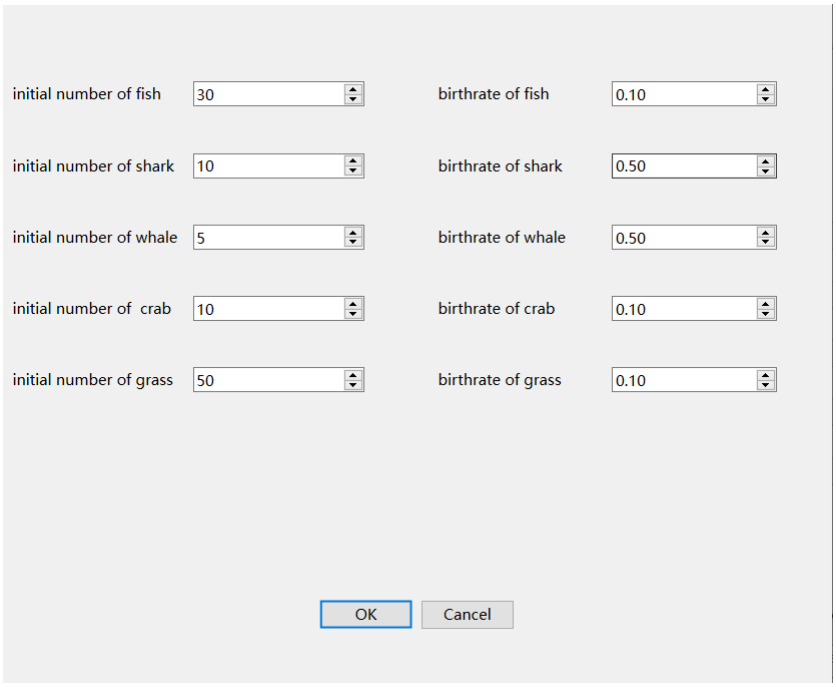
```
class creature : public QWidget
{
    Q_OBJECT
public:
    explicit creature(int x,int y,QWidget *parent = nullptr);
signals:
public:
    decision *dec;
    int type,life;
    int age,orgx,orgy,hp;
    double labelx,labely;
    int distance,eyesight;
    double v;
    double vx,vy;
    QLabel *label;
    QMovie *movie;
    creature(const creature& a,QWidget *parent=nullptr);
    void set_speed(std::vector<creature*>::iterator
st,std::vector<creature*>::iterator en);
};
```

mymove类

继承自QObject，其通过一个creature指针来控制生物，用于实现个体演化中生物的移动

setting类

用于调整生物的初始数量、出生率



神经网络部分

神经网络用分层图的方式构建，按拓扑序进行正向和反向传播（不用矩阵运算是因为网络结构小，并行运算带来的性能提升无法抵消矩阵对象生成、消亡的开销）。network使用了三个类：edge为神经元之间的连边、neural为神经元、network为网络。此外，为了辅助加载数据，我们创建了data_set类，以便设置训练的batch、epoch数，并设有随机旋转函数用以数据增强。环境数据进入data_set之前，先通过data_cache，等到获得奖励后批量加载进data_set。
神经网络需要输入当前环境情况，如果把整个环境中所有生物信息记录下来开销太大，所以我们将环境其它生物相对位置映射到一个六维向量：前两维类似电场力的形式记录相对位置，后四维加入速度信息。神经网络输入这个六维向量以及当前速度信息，预测期望收益，多次训练后可以预测最优速度方向。

三、小组成员分工情况

我们小组在项目合作中积极沟通，王默涵主要负责神经网络的构建、群体演化部分代码；陈思危主要负责将二者接入QT，实现图表显示、生物捕食繁殖移动等功能，搭建QT的运行框架；孙泽华主要负责界面美化，增加项目功能，梳理项目思路等。

我们三个人各自负责的部分也并不是完全独立，我们也一起讨论了神经网络的接入、如何获取神经网络数据、QT的信号槽机制等，在项目推进的过程中共同成长。

四、反思总结

1.在合作项目中要及时反馈项目进度，我们遇到因为沟通不及时或者反馈不够准确而导致的问题。比如某个团队成员可能已经完成了一部分工作，但因为没有及时通知其他成员，导致其他人重复做了相同的工作。因此及时沟通是项目成功的关键。在项目过程中，我们需要建立有效的沟通机制，确保团队成员之间能够实时、准确地传递信息。

2.善用github、csdn等工具，可以提高效率。在合作项目中，我们可能会面临代码管理、文档共享、问题讨论等多方面的挑战。如果没有合适的工具支持，这些工作可能会变得非常繁琐和低效。善用工具是提高项目效率的重要手段。Github等版本控制系统可以帮助我们更好地管理代码，实现多人协同开发；CSDN等技术社区则提供了丰富的技术资源和交流平台，帮助我们快速解决问题和学习新知识。因此，在合作项目中，我们应该积极寻找和使用适合的工具，提高工作效率。

3.QT的平台比我们想象的难用，要提前熟悉工作平台。在使用QT平台进行开发时，我们可能会遇到一些之前没有预料到的问题。由于QT的复杂性和特定性，如果我们没有提前熟悉和了解这个平台，那么在开发过程中就可能会遇到很多困难。提前熟悉工作平台是确保项目顺利进行的基础。