

第一版

100*100 空间

支持某种特定生命的自然出现、自然消亡。害怕孤独。

可以根据区域的情况产生或强或弱的新物种

能模拟不同物种间的捕食关系、允许不同物种间存在共生关系

传代过程中，允许进行属性的微调或者大改，也即遗传和变异

允许环境资源限制，并依据适应性的强弱决定生死

疾病因素，可随机的产生，并且疾病具有遗传性，传染性，可能会自愈

自然环境会自我演化，往好的或者坏的方向发展

//能进行自动统计，算出区域内的优势种，甚至规模群落

特别强的动物必须得捕食其他的生命才能活到下一轮

第二版

初始时：按照随机产生物种的方法，产生一定量的生命。

出现和消亡：一轮繁衍结算时，对每个格子进行讨论，一个生命周围同种生物少于或多于一定的量会消亡，一个无生命的格子周围的同种生命多于一定的量则会产生新生命。若不满足上述条件则格子不发生变化。

模拟物种关系：

捕食&共生：在自然生成和取食自然资源的轮次，如果一个有生命的格子的周围的强的生物比弱的或同等级的生物数量多，则该格子被捕食。捕食将带来能量。

遗传变异：在格子自然生成的过程中，正常情况下，是取生成它的格子的属性进行微处理得到最终属性，在异常情况下，会产生较好或较坏的属性变异。

环境限制：环境内在游戏初可以配置一定的自然资源，在生成的一轮过后，在捕食共生判定之前，按照生物的生命等级的反序进行排序，并分配资源，分配不到自然资源的生物会饿。

饥饿计算：自然取食和捕食过程都会带来能量，捕食所带来的能量可以等效为对自然资源的需求，在三轮结束后，判定是否摄取了足量能量，若不足则饿死。因捕食而产生的多余的能量可以被积累（但有上限）。

新物种产生：结合区域中自然条件，和物种情况，生成能适应环境，且与区域内生物存在比较相近的生物属性。不同的物种具有不同的颜色加以区分。

疾病系统：

特性：疾病通常是较为温和的，即各种传播突进的概率相对低，偶尔会有烈性传染疾病

产生：极低的概率突然出现，区域生物死亡数目影响疾病产生的概率。但总的来说还是特别低的那种，即一盘游戏几乎出不来几次病毒。

传播：依附于生物而存在，在繁衍过程中，依概率传递，临近格子存在传播可能（一

个轮次)，捕食将更有概率得病。

治愈：病毒本身有概率被自愈，生命自愈后会携带有抗体，抗体携带者本身将降低感染的概率。有病个体周围有抗体携带者，将更容易自愈，一般个体周围若有抗体携带者，则不易被感染。

环境演化：生命数量过多，会造成环境破坏，生物死亡和生物数量较少，会使得环境优化。(有上限，无下限)

统计计算：对自然环境的能量，及生物加权统计量以及生物数量的稳定性进行评估，得到对小区块的打分。

软件工程课程设计实验报告一

组员：陶贊 李宜璟 周尊康

任课教师：贾东宁

项目名称：

虚拟群落

游戏背景：

在这片苍茫的大地上，零星分布着多种多样的生命群落，捕食、竞争、互助、生产，生命的活动和更替每天都在悄然发生。病毒肆虐、弱肉强食，区域间的物种更替又会以何种形式展现？一个随机产生的病毒，是否会成为让整片大陆再无生机的罪魁祸首？一块资源丰富的湿地，是否又会成为生命富集的温床？一个个起初不起眼的种群，是会发展壮大，还是日渐式微，让我们拭目以待。

游戏规则：

整个模拟过程在一个 100×100 的网格地图上进行，地图同时又分为 10×10 个局部区域。

每个格子存在有生命和无生命两种状态，有生命的格子又分不同等级的物种，不同等级的物种由不同颜色加以区分。

环境提供一定量可提供能量的自然资源。

生命的产生：

1. 游戏初始时，随机产生一定量的不同物种生命
2. 自然繁衍：一轮游戏结束后，每个无生命的格子和以其为中心的范围区域内格子产生交互，当这个无生命格子周围同物种生命存在 2 个或以上的量时，则该格子会产生新的同物种生命
3. 新物种产生：结合一定范围区域中环境情况和物种情况，生成能适应环境，且与范围区域内已存在物种生物属性相近的新物种。当然也有极低概率产生偏强或者偏弱的物种。

生命的消亡：

1. 自然消亡：一轮游戏结束后，每个有生命的格子和以其为中心的范围区域内格子产生交互，当这个有生命格子其周围无同种生命有或多于 4 个同种生命，则该格子生命自然死亡
2. 掠食消亡：当一个有生命格子的范围区域内，比他高等级的生命数量多于比他低级或同级的生命数量时，则该格子被捕食死亡
3. 生命需要摄取能量，当能量不足以维持生命时，死亡
4. 当生命感染疾病且病发时，死亡

关于生命产生中遗传与变异的说明：

1. 自然繁衍过程中，将要出现的生命的各种属性受其周围生命的影响，在周围同种生命的基础上产生小幅变化，偶尔会出现极好或者极差的属性变化。

关于能量摄取的说明：

1. 摄取自然环境提供的能量与捕食获取的能量等效
2. 在摄取自然环境提供的能量时，低级物种更容易摄取
3. 摄取自然环境提供的能量不足的生命将在邻近的格子中尝试寻找可供捕食的生命，若满足条件，则进行掠食

关于疾病的说明：

1. 特性：疾病的主要属性为传播途径及传播的可能性，自愈的可能性，致死的可能性。疾病传播通常是较为温和的，即各种传播途径的概率相对低，偶尔会有烈性传染疾病。
2. 传播：现有疾病依附于生命而存在，在生命繁衍过程中，依概率传递给同物种下一代，在掠食过程中，依概率传递给捕食者。
3. 发作：在掠食过程结束后，携带病毒的个体有可能会发病致死。
4. 产生：在发作过程结束后，疾病将以极低的概率出现于某区域中的某

个体身上，区域生物死亡数目影响疾病产生的概率。但总的来说还是很少会产生疾病。

5. 治愈：病毒本身有概率被自愈，生命自愈后会携带有抗体，抗体携带者本身将不再会被感染。且抗体可以在繁衍过程（必然）传播。

关于环境演化的说明：

1. 每个局部区域的数值属性可能发生适当的改变，基本遵循生命多资源少，生命少资源多的规律。

游戏过程：

游戏基本有如下过程：

游戏初始在随机区域内产生一定数量的不同种生命，然后开始游戏

生命自然繁衍产生和自然消亡

已存在的生命对自然资源进行能量摄取

掠食过程

疾病过程

数量统计,在上述过程全部结束后，分区域统计一轮内新增的生命数，消亡的生命数，物种丰富度，生命个体总数等数据并记录。

用户操作及内部具体流程见图 1-a，图 1-b

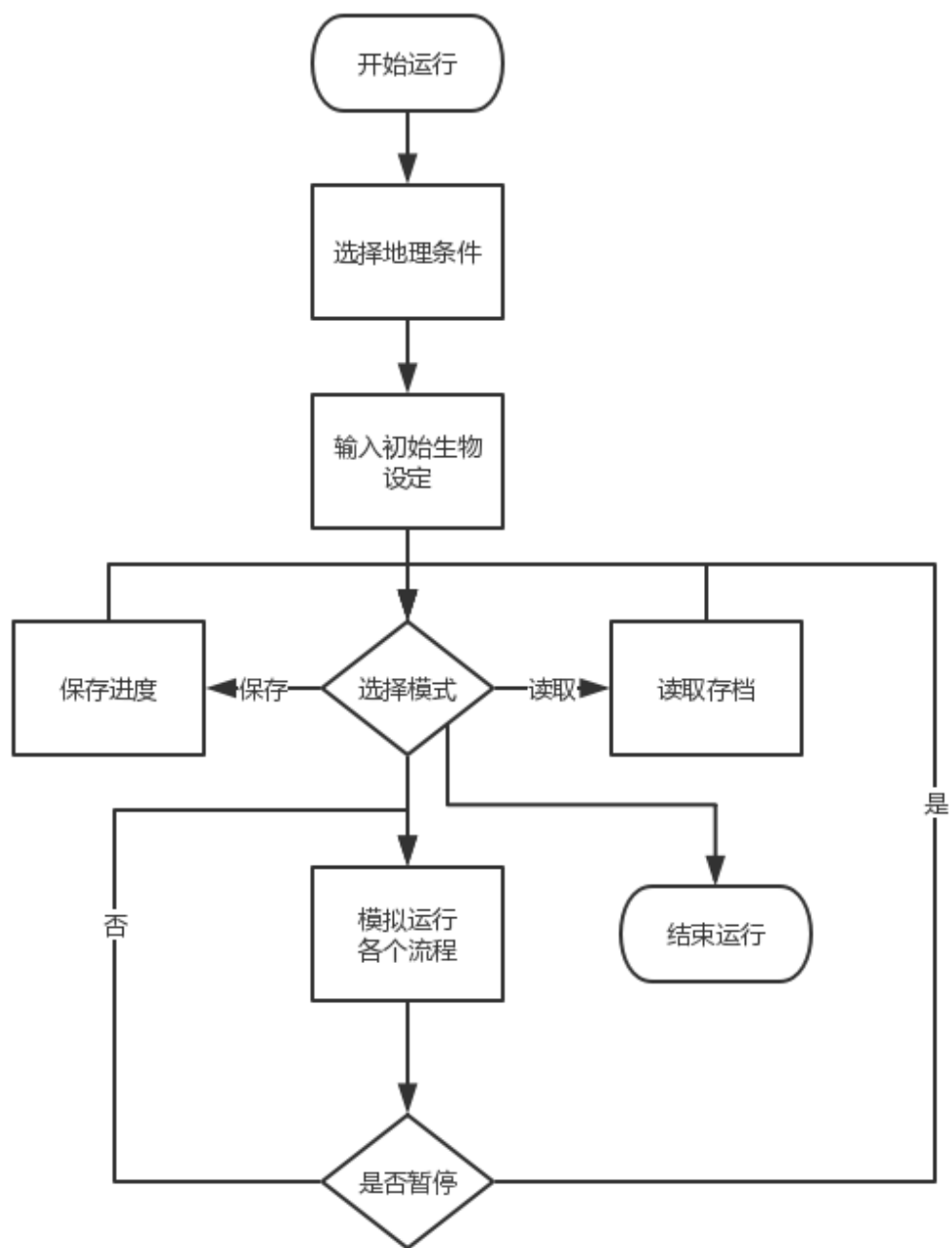


图 1-a 用户操作流程

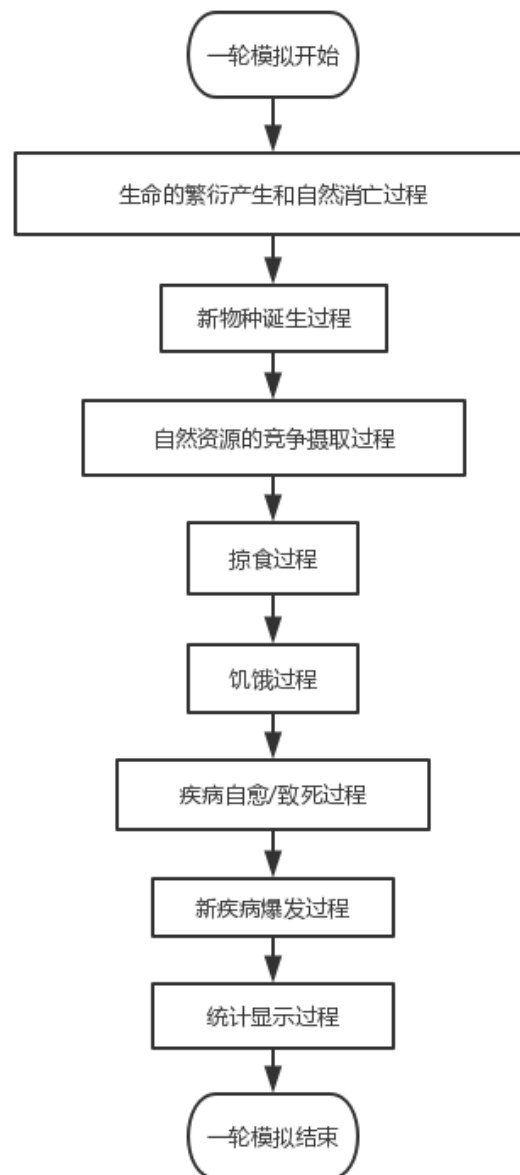


图 1-b 模拟过程的具体内部流程

人工输入：

用户可以自行设置区块资源属性和初始的生命群落。

在区块资源分布上，用户可以模仿地理环境中资源分布的特点，如不规则性、连续性、有限性等特性，以此设置各个区域的自然资源数量，同时也可以模拟有趣的极端条件。

在生命群落方面，用户既可以依据现实世界中群落关系来生成初始的群落，也可以发挥自己的想象，来尝试创造现实中不可能存在的群落。

项目目标：

模拟程序本身可以在任何的设定背景下无限的运行下去，用户既可以通过图形界面宏观的了解到群落的演变过程，如竞争，共存，掠食，迁徙，集群，衰亡等，又可以通过精确的数据反馈，来验证自己的某些推论。

通过对自然过程尽可能贴切的模拟，以此解答怎样的环境情况是适宜生命的，以及怎样的生命群落关系能够稳定的发展持续下去。

该阶段人员分工：

陶贇(15020031070)：分析系统需求，确定游戏基本规则，游戏目的等要素，撰写了报告的初始模版。

周尊康(15020031123)：梳理语言逻辑，对规则设计的不足进行修正。完成后期报告的综合及排版。

李宜璟(15020031034)：梳理设计游戏流程并图表化，明确规则中的细节。

软件工程课程设计实验报告二

组员：陶贊 李宜璟 周尊康

任课教师：贾东宁

对象的识别：

通过上述对于游戏规则和设定的讨论，我们基本得出了这个游戏过程中涉及到哪些对象，从游戏过程本身来看，这当中涉及到了生命，病毒，抗体，区域，地图等对象。从设计实现与完善用户体验的角度来看，还需要增加一个具有综合控制效用的控制器（相当于客户端）来辅助完成数据的读入与游戏结果的视觉呈现。

类的关系：

由于本次课程设计中所涉及到的类较少，相互之间的共性较为难以发掘，因此在设计时，类的关系主要以符合逻辑认知的组合关系为主，即某物是另一物的一部分。基本的关系雏形如图 2 所示

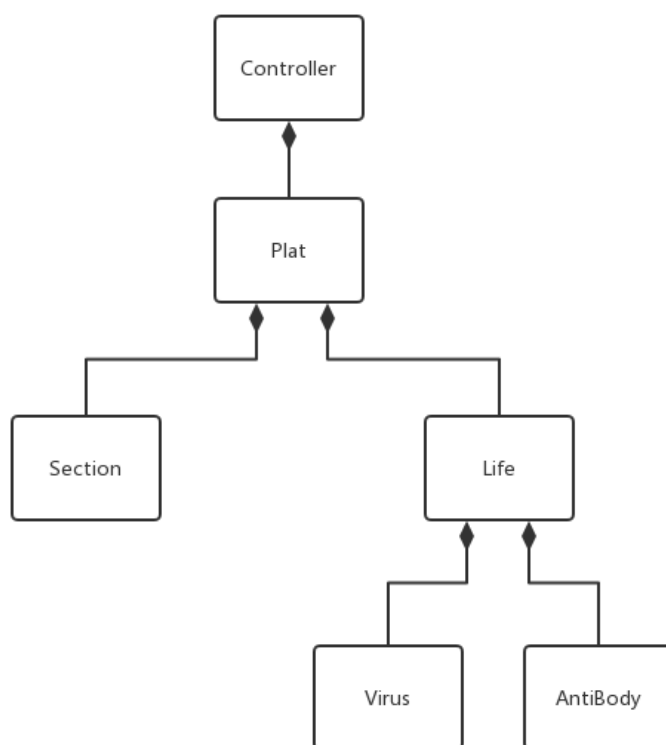


图 2 类的关系示意图

类的设计：

因为游戏过程中各对象之间主要存在的是组合关系，因此类的设计上的难点主要表现为对于对象本身所拥有的子对象的操作和通信，比如生命体和生命体之间传递病毒等。

类的基本定义如图 3 (由于已经基本完成了一个相对简单的版本的编码工作，因而定义部分在思考完善后已经较为完整且冗长)

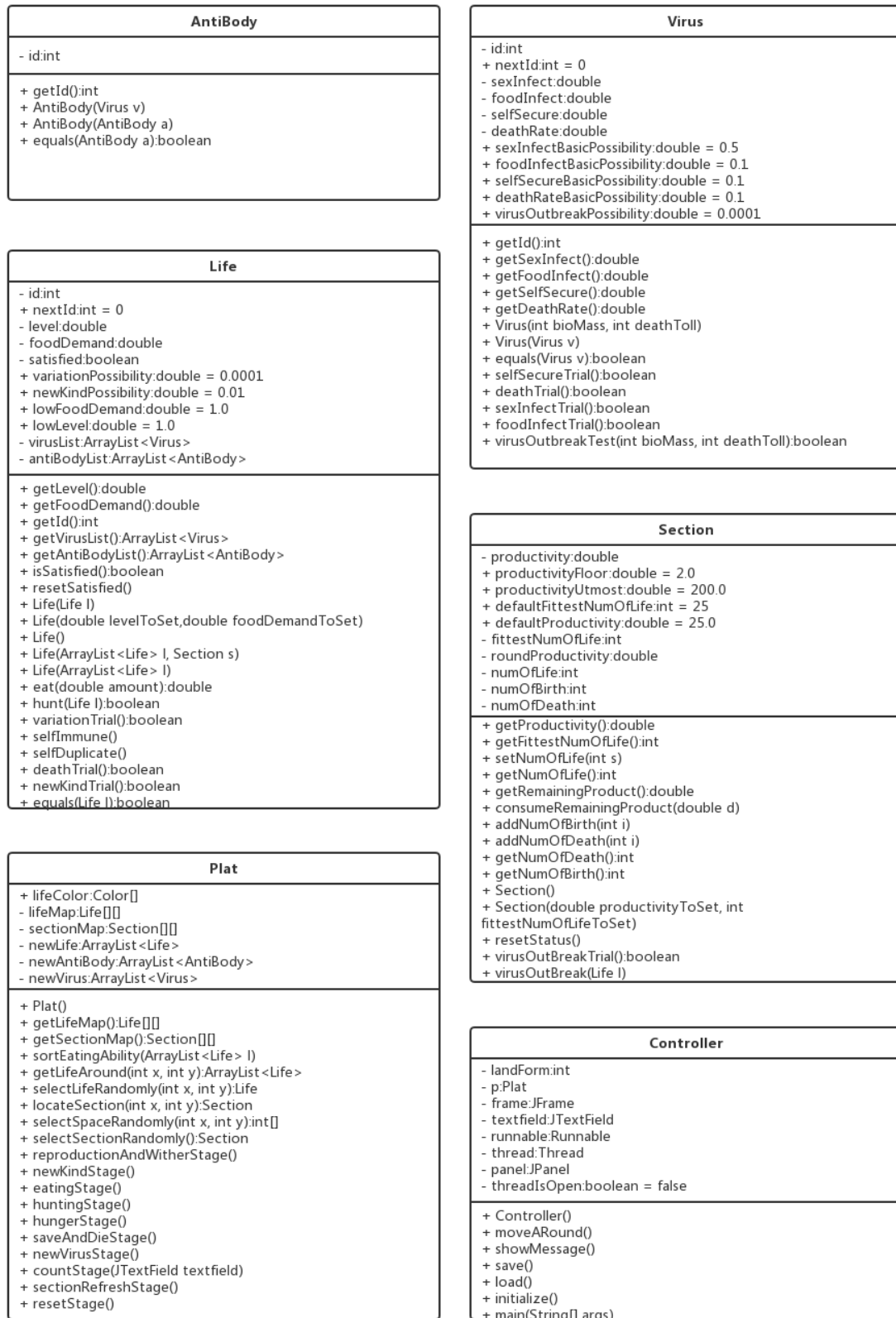


图 3 类的基本定义

难点及初步解决对策：

动态数量的对象列表问题：

问题描述：在该程序中存在若干处包含关系，例如生命包含病毒，地图包含生命这样的关系，用以往的思想会想到用对象里包含着一个数组成员来解决，然而面对数量统计，动态操作等需求时会显得力不从心而且很不灵活，大大影响编码效率。

问题解决：使用标准类库提供的一些容器，例如 ArrayList，可以动态的管理批量数据，学习掌握了例如计数、取元素、增加元素、查找、删除、清空等功能。减少了一定的工作量。又比如可以基于 Collection 对数据进行适当的排序。

可视化的基本解决方案：

问题描述：由于程序的设计目标是要呈现出丰富的生命体系，因此单纯的依靠字符界面的输出能力和表现能力不足以满足需求。亟待寻找更加丰富且易于入门的图形渲染方式。

问题解决：JAVA 自带了一些简单的图形编程库，例如 Swing，既可以构建出基本的控件布局，也可以进行自定义的二维图形渲染工作。通过反复为地图上有生命的点着色，可以有效的表现出游戏过程。

概率事件模拟：

问题描述：游戏过程中涉及到很多判断，比如“以 0.01 的概率爆发疾病”，需要正确的模拟这种概率事件的随机发生过程。

问题解决：参考网上的相关解决方法，发现都是利用随机数的性质来实现的，由于 Math.random() 可以以均等的概率在 [0,1) 区间产生一个随机浮点数，那么就可以以 P 的概率产生在区间 [0,P) 的一个随机数，用形如 $P < \text{Math.random}()$ 既可以判断概率为 P 的事件是否发生。

对象的层次管理：

问题描述：对象的数据访问和操作是依赖于类上定义的 setter 和 getter 的，在实现的时候发现，若出现了跨层次的操作，则会变的麻烦，而且影响了语义的连贯性。这个问题在生命-区域-地图层表现的较为明显，地图类中有需要直接对生命进行的控制操作，比如模拟生命的繁衍过程，然而这个过程不能通过地图-区域-生命逐级向下的查找，因为通过区域反而会限制繁衍的范围。

问题解决：适当的调整对象的组合层次，例如将地图-区域-生命的三层模式改变为地图-区域 地图-生命的 T 型组合模式，从而解决了矛盾。同样的例子还有通过控制台层次兼顾操作地图与图形渲染工作。

自动执行：

问题描述：程序第一次完成的时候需要人工点击才能模拟一轮，要观察连续过程，不得不快速的点击鼠标，这样给用户带来了一些负担，希望能做成点击一下按钮，程序自动连续运行，再次点击，程序可以暂停。

问题解决：查阅网上资料，了解到可以新建额外的线程反复执行特定的程序。因此，尝试新建了一个线程，在该线程内程序会反复的计算游戏的演变过程，在按下按钮时，线程被启动，独立的反复执行计算-显示的工作，再次按下时，相应线程被停止。从而实现了相应需求。

更丰富的内容反馈模式：

问题描述：程序本身交互性相对较差，除了初始化的数据输入，全程大部分是在观察，理解。而且原本设计中只有纯图形输出，对于本来不太懂程序图形输出的含义的用户并不友好。

问题解决：设计并加入了一个类似于热点新闻的模块，即在每轮结束后，会在窗体顶部显示一些本论中发生的关键信息，信息的内容类型不定。

病毒——抗体关系实现：

问题描述：如何用编程语言表述病毒激发出抗体，抗体抑制病毒的过程。

问题解决：抗体的实例是由构造函数产生的，既然和病毒是关联的，就考虑用病毒本身作为参数来构造出抗体，为了方便实现二者的匹配关系，给病毒和抗体各增加一个整形成员变量，表示同一类病毒以及与之对应的抗体。在进行模拟免疫的过程中，比较相应字段即可判断是否免疫成功。用类似的思路，还可以模拟出生命的产生，病毒的爆发等过程。

需求的变更过程：

增加存取进度功能：

因为游戏本身具有随机性，因此较难定向的构造某种特定的格局，若要保持并观察某种特定的初始格局会变的几乎不可能，因此考虑添加保存和读取进度的功能。

增加要闻功能：

原因已在上一部分描述。

增加预设地形设定：

本程序的地形是要分块处理的，即便已经以 10×10 的规模划分了 100 个区块，但是一个一个初始化的任务量还是较大，因此考虑提供两三种相对简单且有区分效果的地形分布，如一半一半，或者中心-边缘等模式。

病毒治愈途径的改进：

在对于疾病过程的模拟中，发现疾病仅靠血缘来治愈是很困难的，仅有局部的少量免疫个体长期存在，因此增加了接触式的自愈这一新的治愈模式。即个体周围有免疫个体，可以使之相应的获取免疫能力。

该阶段内组员分工：

陶贊(15020031070)：负责分析出程序中所涉及的类，按照具体设计画出类图和类的层次示意图。找出并提出程序中的设计缺陷并辅助进行修改。

周尊康(15020031123)：参与分析类的具体设计细节，对游戏交互模式进行设计和修改，汇总并完成该阶段实验报告。

李宜璟(15020031034)：负责构思各个类的具体细节，并基于部分设计编码实现了一个相对简单的演示版本，用于发现可能存在的需求变更。