Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования   
«Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель от вуза\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Ю. Андреева

(подпись) (и.о., фамилия)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

(дата)

Отчет об учебной (технологической) практике

«Имитационное моделирование распространения Коронавируса»

УП 09.03.04. 1.15 О

Студент группы ПИ-01 Н.Н. Козловский

и.о., фамилия

Руководитель практики доцент, к.ф.-м.н. А.Ю. Андреева

должность, ученое звание и.о., фамилия

Барнаул 2022

ФГБОУ ВО «Алтайский государственный технический университет   
им. И. И. Ползунова»

Кафедра «Прикладная математика»

**Индивидуальное задание**

**На учебную (технологическую (проектно-технологическую) практику)**

(вид и тип практики по УП)

студенту Козловский Никита Николаевич группы ПИ-01

(Ф.И.О.)

**График проведения практики**

|  |  |  |
| --- | --- | --- |
| **№**  **п/п** | **Наименование работ, выполняемых на практике** | **Сроки выполнения** |
| 1 | Разработка техническое задание |  |
| 2 | Проектирование объектной модели |  |
| 3 | Реализация модели в виде программы с графическим интерфейсом |  |
| 4 | Написание отчета и защита практики | 14.07.2022 |

Руководитель практики от университета АндрееваА.Ю., доцент каф. ПМ

(подпись) (Ф.И.О., должность)

Задание принял к исполнению \_\_\_\_\_\_\_

(подпись) (Ф.И.О.)

**Инструктаж по ОТ, ТБ, ПБ, ПВТР**

Инструктаж обучающегося по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка проведен «20» июня 2022 г.

Руководитель практики от

Университета Андреева А.Ю. доцент каф. ПМ

(подпись) (Ф.И.О., должность)

**Аннотация**

Отчет об учебной практике содержит описание программы, реализующей моделирование распространения коронавируса в России: техническое задание, структуру данных, структуру файлов, описание программного продукта. Код программы на языке C++ CLI размещен в репозитории на Github и в приложении Б. В приложении А приведены снимки экранных форм программы.

Отчет содержит 46 страниц, 9 рисунков, 3 источника литературы.

**Оглавление**

[Введение 5](#_Toc123675721)

[1. Техническое задание 6](#_Toc123675722)

[1.1. Терминология 6](#_Toc123675723)

[1.2. Описание процесса функционирования модели 6](#_Toc123675724)

[1.3. Требования к функциональности программы 6](#_Toc123675725)

[2. Проект программного продукта 7](#_Toc123675726)

[2.1. Математическая модель 7](#_Toc123675727)

[2.2. Диаграмма классов 10](#_Toc123675728)

[2.3. Жизненный цикл объектов модели 11](#_Toc123675729)

[3. Описание программного продукта 12](#_Toc123675730)

[3.1. Выбор средств реализации 12](#_Toc123675731)

[3.2. Описание классов 13](#_Toc123675732)

[Заключение 15](#_Toc123675733)

[Список использованных источников 16](#_Toc123675734)

[Приложение A. Снимки экранных форм пользовательского интерфейса 17](#_Toc123675735)

[Приложение Б. Исходный код 21](#_Toc123675736)

Введение

Моделирование – основной метод исследований объектов, процессов или явлений, с целью получения объяснений этих явлений, а также для предсказания развития явлений и процессов, интересующих исследователей. Задача моделирования - выявить главные, характерные черты явления или процесса, его особенности, его поведение. Моделирование применяется в разных областях человеческой жизни: в медицине, демографии, страховании, социологии, в научно-исследовательской деятельности, в повседневной жизни и даже в компьютерных играх.

Модель (объекта – оригинала) – вспомогательный объект, отражающий наиболее существенные для исследования закономерности, суть, свойства, особенности строения и функционирования объекта-оригинала. Задачи моделирования:

* Понять сущность изучаемого объекта
* Научиться управлять объектом и определять наилучшие способы управления
* Решать прикладные задачи

Также моделирование позволяет прогнозировать прямые или косвенные последствия того или иного процесса или явления. Это одна из самых главных задач моделирования в случае с распространением вирусов.

С тех пор как началась пандемия COVID-19 многие люди задаются вопросом, что это за вирус, как он развивается и распространяется. Почему необходимо носить маски и как прививка может помочь им в борьбе с этим вирусом. С развитием пандемии в интернете появилось множество статей на эту тему: научных и журналистских. Но люди склонны не обращать на них внимания, потому что с каждым днем их становится все больше и больше и не все они вызывают доверие.

Визуальное же представление проблемы гораздо сильнее влияет на людей. Оно занимает мало времени и одновременно дает возможность понять, как распространяется вирус и от чего это зависит. Вот почему задача создать визуальную модель распространения коронавируса так актуальна.

1. Техническое задание
   1. Терминология

**Поле** – это модель мира, которая включает в себя решетку ячеек, отображающих людей. На ней происходит моделирование поведения коронавируса.

**Человек** – это отдельная ячейка на поле. Ее цвет варьируется в зависимости от статуса человека по отношению к вирусу: здоровый, инфецированный (инкубационный период болезни), больной, выздоровевший.

**Переносчик** – это человек, который является носителем вируса без каких-либо симптомов (в том числе во время инкубационного периода) – в модели отмечен розовым цветом.

**Заболевший** – это человек, который является носителем вируса с характерными симптомами и осложнениями– в модели отмечен красным цветом.

**Выздоровел** – это человек, который был переносчиком или заболевшим и успешно справился с боленью (больше не инфицирован), для модели неважно обнаружили у данного человека антитела к вирусу или нет – в модели отмечен серым цветом.

**Инкубационный период**–промежуток между тем, когда человек заражается, и тем, когда проявляются первые симптомы.

* 1. Описание процесса функционирования модели

Модель функционирует на плоской карте и имитирует процесс распространения инфекционного заболевания.

На карте размещены квадратные поля, обозначающие людей с разным состоянием: переносчик, заболевший, выздоровел. С окружающими клетками взаимодействуют только переносчики и заболевшие. При взаимодействии учитывается дальность контакта и количество контактов. Это сделано для имитации случайного тесного взаимодействия переносчика с окружающими людьми (Например, коллеги, семья, кружки и секции). После взаимодействия с некоторым шансом определяется заразился ли человек или нет. При успешном заражении, новый переносчик получает право взаимодействия с окружающим пространством.

Всего на карте размещено 6561 человек, что имитирует взаимодействие маленькой социальной ячейки. Их количество фиксировано, и не изменяется путём пользовательских настроек, только программируемым путём.

* 1. Требования к функциональности программы

В программе имеется графический интерфейс: одна главная форма, на которой расположено поле для отображения карты.

На поле отображаются объекты 2 типов, при этом по мере выполнения программы они осуществляют взаимодействие друг с другом. При запуске в середине карты располагаются пять переносчиков в виде креста.

Предусмотрены следующие типы объектов:

* Человек (Переносчик, заболевший, выздоровевший);
* Карта заболевания

Движение объектов на карте осуществляется по таймеру и начинается сразу же после запуска программы, в данном ПО функция приостановки (паузы) представлена в виде повторного нажатия кнопки потокового запуска симуляции «Run».

Логический конец симуляции наступает в тот момент, когда на карте не остается переносчиков и заболевших, так как шанс единственный предусмотренный исход: выздоровление всех заболевших. При этом симуляция останавливается, о чём сообщает появляющееся диалоговое окно. При повторной попытке запуска, без сброса симуляции соответствующей кнопкой «Restart», сообщение о её завершении появится снова.

Программа является интерактивной, то есть пользователь напрямую может влиять на исход симуляции, благодаря изменению параметров распространения болезни и пошаговому воспроизведению кнопкой «Step»

1. Проект программного продукта
   1. Математическая модель

Human(человек) – единица симуляции, отвечающая за отображение состояния, и технические параметры расположения на карте.

Характеристики Human:

* Location – структура положения на фактической пиксельной карте
  + X – левый верхний пиксель по оси X
  + Y – левый верхний пиксель по оси Y
  + Size – размер квадратной ячейки
* Position – структура положения на формально разделённой карте
  + Row - строка
  + Column – столбец
* AllInfected – заражены ли все в радиусе контакта (флаг)
* Color – цвет ячейки обозначающий её состояние
* ConditionTime – таймер состояния (в количеств событий таймера)

MapDisease – основной объект отвечающий за взаимодействие объектов Human между собой.

* Picture - отображение карты
* Map – карта
* Dimension - размерность карты эпидемии
* Network - все элементы карты
* Ills - Заболевшие люди (Элементы с состоянием)
* Processing - таймер обработки эпидемии (отсчитывает дни)
* Activate - свойство таймера «Активирован»
* Duration - Длительность состояний (в днях)
  + Incubation – инкубация
  + Illness - время болезни
  + Spreading - вероятность заражения от [0;100]
* Interactions - Взаимодействие людей
  + touch - количество контактов в день
  + distance - расстояние контакта

Map (Поле симуляции) - пиксельное изображение, раздробленное на сегменты для расположения объектов. При создании поля учитывается не только формальное разбиение карты на матрицу *N*x*N*, но и ширина ячейки и ширина разделителя. В каждой точки матрицы расположена одна сущность человек, с установленным параметром состояния: здоровый (и соответствующим белым цветом).

При создании карты выделяется память под динамический одномерный массив объектов Human. Так вся матрица NxN располагается построчно в массиве. В центре карты появляются нулевые пациенты (переносчики). Для расчёта середины матрицы используется формула:

position = Dimension \* Dimension / 2, где Dimension – длина строки

После составляется массив с позициями расположенных рядом ячеек:

{position - Dimension, position + Dimension, position - 1, position + 1}

По этим позициям в цикле перекрашиваются ячейки и изменяются их состояния. Все 5 объектов добавляются в вектор Ills, содержащий все объекты, способные взаимодействовать с окружающими (на данный момент только больных).

**Взаимодействие объектов**

За взаимодействие объектов отвечает функции обработки события Tick «истечение времени таймера, при условии, что он включён». Благодаря этой функции реализовано потоковое и пошаговое воспроизведение симуляции, т.к. её можно вызывать не только при событии Tick, но и программно, как при функциональном программировании.

После вызова данной функции для каждого i-ого (diseased) элемента вектора Ills вызывается проверка в четыре основных этапа:

* **Увеличение времени в текущем состоянии** (diseased->ConditionTime += 1)
* **Удаление элементов с завершённым состоянием.**

diseased->ConditionTime >= соответствующему элементу в структуре длительностей состояние (MapDisease::Duration). Соответствие состояния и времени на состояние определяется сравнением цвета элемента. При выполнении всех условий происходит изменения Human: переносчик => больной (изменение цвета ячейки, сброс времени в текущем состоянии); больной => здоровый (изменение цвета, сброс счётчика, удаление из вектора Ills).

* **Определение нового заболевшего**

Если Human не здоровый и в расстоянии контакта (MapDisease::Interactions.distance) есть ещё не заражённые элементы (diseased->AllInfected == false), то для этого объекта происходит выбор случайных элементов, кол-во которых равно кол-ву контактов в день (MapDisease::Interactions.touch). Выбор происходит рандомно.

Процесс выбора случайного контактного объекта Human:

* Считать что вокруг нет здоровых(diseased->AllInfected = true)
* Получаются координаты текущего объекта.
* Получение смещений для row и colum не превосходящий расстояние контакта:

Создание диапазона чисел c равным распределением вероятности для выбора смещения по X из диапазона [-distance; distance], с вероятностью получения у каждого числа равной . Для этого будет использован Вихрь Мерсена, лишённый многих недостатков, присущих другим ГПСЧ, таких как малый период, предсказуемость, легко выявляемые статистические закономерности.

std::random\_device seed – ключ, использует информацию об аппаратном обеспечении вашего компьютера для генерации истинных случайных значений

std::default\_random\_engine random(seed()) – ГПСЧ Вихрь Мерсена. Получаемое распределение напрямую зависит от значения состояния, передаваемого в качестве фактического параметра конструктора.

std::uniform\_int\_distribution<int> row\_random(-Interactions.distance, Interactions.distance) – определение диапазона с равным распределением вероятности.

row = row\_random(random) **–** получение смещения по X (строка)

Расстояние контакта вычисляет в количестве пройденых объектов от центра. Это значит, что объект не может касаться элементов карты по диагонали напрямую. Ему необходимо использовать шаги по X и Y координатам, чтобы преодолеть этот путь. Поэтому визуально распространение болезни имеет вид ромба. Для реализации этого диапазон смещения по столбцам будет:

colum\_distanse = Interactions.distance - std::labs(row) **–** модуль диапазона. Из расстояния вычитается модуль полученного смещения по X;

std::uniform\_int\_distribution<int> colum\_random(-colum\_distanse, colum\_distanse)

int colum = colum\_random(random) - Получение смещения по Y(колонка)

После получения смещений вычисляется позиция этого объекта по формуле:

int position = (row + diseased->Position.Row) \* Dimension + colum + diseased->Position.Colum;

Где:

Dimension - размер строки

diseased->Position.Row - текущая формальная позиция по X

diseased->Position.Colum - текущая формальная позиция по Y

row - смещение по X

colum - смещение по Y

Фактическое значение строки в массиве: (row + diseased->Position.Row) \* Dimension

Фактическое значение элемента в строке: colum + diseased->Position.Colum

На нахождение таким образом позиции здорового объекта в пределах дальности контакта уходит три попытки.

* **Заражение**

Здоровый объект в массиве, по полученной в предыдущем шаге позиции, помечается, как контактировавший с заражённым: Infected = Network[position]. Для таких объектов проводится проверка. С установленной пользователем вероятностью объект будет заражён.

std::uniform\_int\_distribution<int> Chance\_Spreading(1, 100) - диапазон шанса заражения. Т.к. у сотни чисел шанс выпадения каждого 1/100, то суммарный шанс выпадения одного из этих чисел в промежутке от [1; Duration.Spreading] равна , где Spreading – вероятность заболевания.

Таким образом если выпавшее число <= Spreading, то объект заражается, значит вокруг элемента было кого заражать и флаг diseased->AllInfected = false, а объект добавляется в вектор Ills. После заражения этот цикл нахождения позиции в 3 попытки и заражения с пользовательским шансом повторяется.

* 1. Диаграмма классов

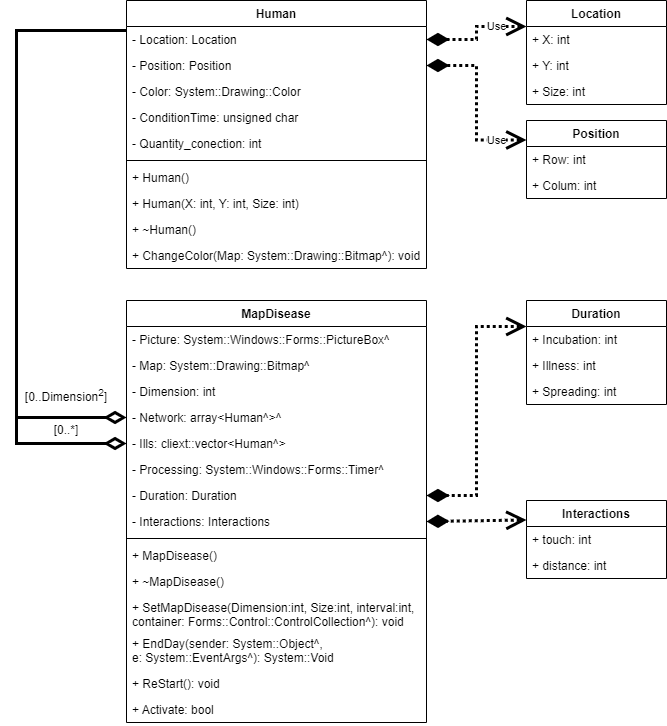


Рисунок 2.1 Диаграмма классов программы

* 1. Жизненный цикл объектов модели

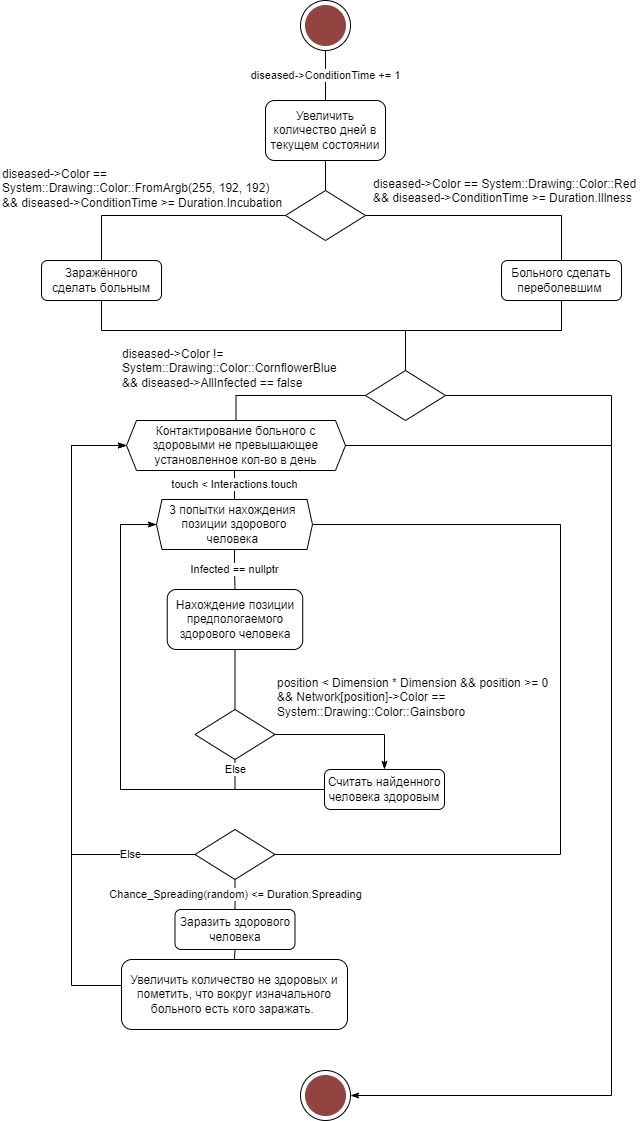


Рисунок 2.2 Диаграмма деятельности объектов Human

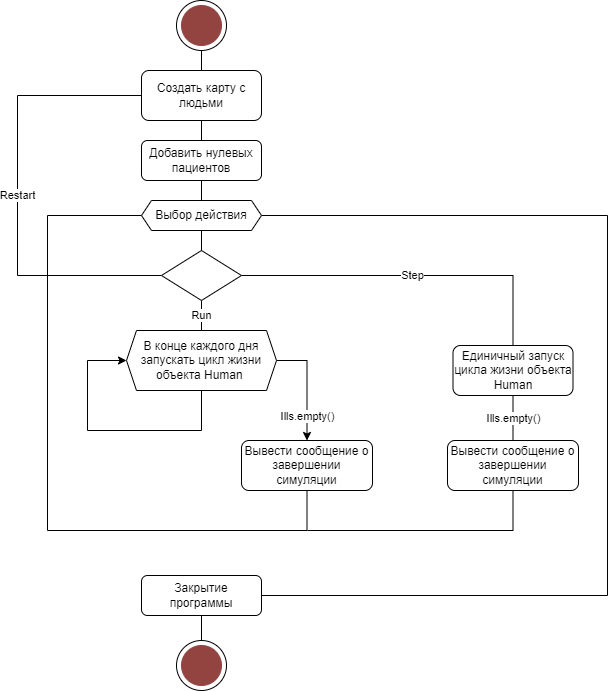


Рисунок 2.3 Диаграмма деятельности объекта карты

1. Описание программного продукта
   1. Выбор средств реализации

Данная модель была реализована на языке С++ с использованием среды разработки программного обеспечения Microsoft Visual Studio 2019. Язык С++ - компилируемый статически типизированный язык программирования общего назначения. Язык C++/CLI – язык для среды программирования Microsoft .NET. Он интегрирует C++ стандарта ISO с Объединённой Системой Типов (Unified Type System, UTS), рассматриваемой как часть общеязыковой инфраструктуры (Common Language Infrastructure, CLI). Он поддерживает и исходный уровень, и функциональную совместимость исполняемых файлов, скомпилированных с родного и управляемого C++

Программа соответствует всем парадигмам объектно-ориентированного программирования.

Использовались стандартные библиотеки языка C++:

#include <cliext> — библиотека контейнеров C++/CLI

#include <cliext/vector> — Объект стандартной библиотеки шаблонов STL/CLR. Класс шаблона описывает объект, который управляет последовательностью элементов с произвольным доступом. Контейнер vector используется для управления последовательностью элементов в качестве непрерывного блока хранилища. Блок реализуется в виде массива, который растет по требованию.

#include <cliext/iterator> — Итератор, объект, который может перебирать элементы в контейнере стандартной библиотеки С++ и предоставлять доступ к отдельным элементам. Все контейнеры стандартной библиотеки С++ предоставляют итераторы, чтобы алгоритмы могли получить доступ к их элементам стандартным способом, независимо от типа контейнера, в котором сохранены элементы.

#include <random> — Определяет средства для генерации случайных чисел с равномерным распределением.

* 1. Описание классов

**Классы:**

**MapDisease** – класс, реализующий работу модели распространения коронавируса среди объектов Human.

Поля:

* System::Windows::Forms::PictureBox^Picture – Объект для отображения карты
* System::Drawing::Bitmap^ Map – Карта заболевания (в изображении)
* int Dimension - Размерность карты эпидемии
* array<Human^>^ Network - Карта заболевания (в объектах)
* cliext::vector<Human^> Ills - Люди, способные распространять болезнь
* System::Windows::Forms::Timer^ Processing - Таймер обработки эпидемии

Методы:

* property bool Activate – свойство, возвращающее активность таймера
* ref struct Duration - Структура длительности состояний (в днях)
  + int Incubation - Инкубация
  + Illness - Течение болезни
  + Quarantine - Самоизоляция
  + Spreading - Шанс заражения
* ref struct Interactions – Структура взаимодействий людей
  + int touch - Количество контактов в день
  + int distance - Расстояние
* MapDisease() - Стандартный конструктор карты эпидемии
* void SetMapDisease(int Dimension, int Size, int interval, System::Windows::Forms::Control::ControlCollection^ container) – Создание карты эпидемии c выводом на экран.
* ~MapDisease() {} - Деструктор
* System::Void EndDay(System::Object^ sender, System::EventArgs^ e) – Конец дня
* void ReStart() – Перезапуск модели.

**Human** – класс человека, реализующий изменение визуальное отображение в модели.

Поля:

* ref struct Location - Структура расположения ячейки на изображении (фактическое расположение)
  + int X, Y - Координаты верхнего левого угла
  + Size - Размер ячейки
* ref struct Position – Структура формального расположения объекта в матриц
  + int Row, Colum - Номер строки и столбца на карте
* System::Drawing::Color Color - Визуализация состояния (Цвет ячейки)
* unsigned char ConditionTime - Время состояния объекта

Методы:

* Human() - Стандартный конструктор
* Human(int X,int Y,int Size) - Создание объекта с заданным размером и позицией
* ~Human() {} – деструктор
* void ChangeColor(System::Drawing::Bitmap^ Map) – визуальное изменение состояния человека на карте заболевания.

Заключение

Была разработана программа, которая обладает следующим функционалом:

* Визуализации распространения заболевания, в зависимости от заданных параметров:
  + Дальность контактов
  + Кол-во контактов
  + Вероятность заболевания
  + Время инкубационного переода
  + Время болезни
* Полное управление моделью пользователем с помощью кнопок:
  + Step – шаг визуализации
  + Run – потоковый запуск программы и её остановка
  + Restart – перезапуск модели.

Это позволяет изменять параметры во время симуляции и влиять на её исход, не дожидаясь завершения с установленными парамметрами.

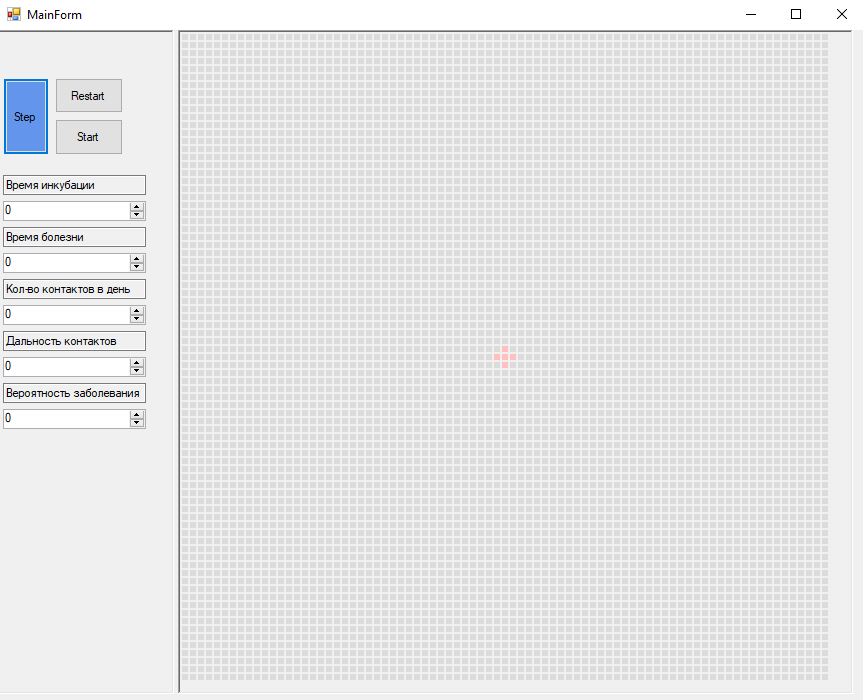
Возможны дальнейшие усовершенствования программы:

* Учет других значимых факторов, влияющих на распространение вируса
* Проверка на эффективность той или иной вакцины
* Создание визуализации модели непрерывно по времени

Список использованных источников

1. Пахомов Б.И. C/C++ и MS Visual C++ 2010 для начинающих. – СПб.: БХВ-Петербург, 2011. – 736с.
2. Википедия [Электронный ресурс]. – Режим доступа: https://wikichi.ru/wiki/Rule\_30, свободный.
3. Сайт CyberForum [Электронный ресурс]. Режим доступа: https://www.cyberforum.ru/, свободный.

Приложение A. Снимки экранных форм   
пользовательского интерфейса

*Рисунок A. 1*

Появились переносчики. Они же люди, находящиеся в инкубационном периоде. Они ничего не знают о своей болезни, но уже распространяют вирус вокруг себя. При нажатии кнопки шаг появляются люди, заболевшие с симптомами.

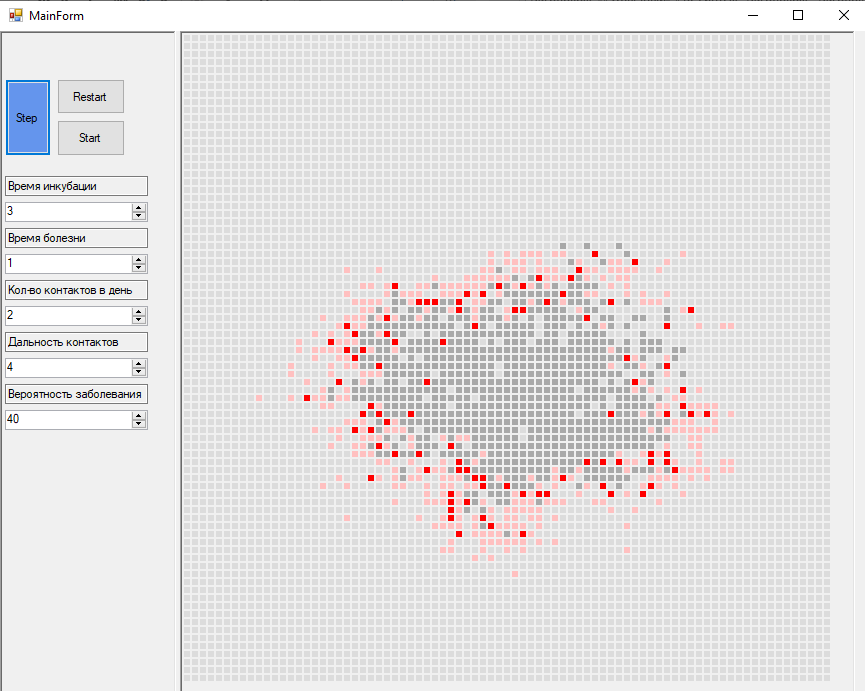


Рисунок A. 2

Как показывает модель самое большое влияние на распространение болезни оказывает – кол-во контактов в день. Так при вероятности заболевания всего в 10 процентов, но с разницей количества контактов в 6 человек (2 и 8), кол-во переболевших радикально отличается.

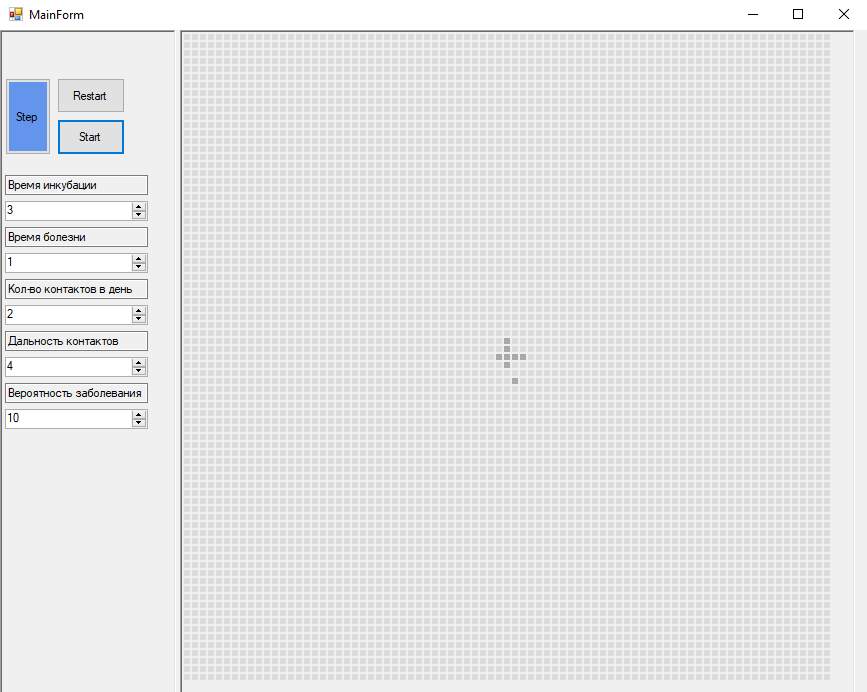


Рисунок A. 3



Рисунок A. 4

Таким образом даже самое не заразное заболевание при частом контакте, очень быстро распространяется, даже при малом совокупном времени инкубации и болезни.

Приложение Б. Исходный код

Исходный код программы находится на GitHub (<https://github.com/BF8DF7D/Kursovaya>)