

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО АВТОНОМНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
КАФЕДРА «ИНФОРМАЦИОННЫЕ СИСТЕМЫ»

Направление подготовки 09.03.04
«Программная инженерия»

Утверждаю
Заведующий кафедрой ИС

_____ Р.А.Валиев

_____ г.

КУРСОВАЯ РАБОТА

по дисциплине:

«Программирование»

на тему:

«Разработка прикладной программы с использованием объектно-ориентированной
технологии»

Автор:
студент группы 2201122

_____ М.Р.Миниахметов

Оценка: _____

Руководитель:
к.т.н., доцент кафедры ИС

_____ Е.В. Зубков

Дата защиты:
_____ г.

Набережные Челны
2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
НАБЕРЕЖНОЧЕЛНИНСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО АВТОНОМНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
КАФЕДРА «ИНФОРМАЦИОННЫЕ СИСТЕМЫ»
Направление подготовки 09.03.04
«Программная инженерия»

Утверждаю
Заведующий кафедрой ИС

_____ Р.А.Валиев

_____ г.

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студент

Миниахметов Марсель Ришатович

1 Тема

«Разработка прикладной программы с использованием объектно-ориентированной технологии»

2 Срок представления к защите

_____ г.

3 Исходные данные

- Правила развития искусственной жизни

4 Перечень подлежащих разработке вопросов

Написать программу «Cell Life Application» (Приложение клеточной жизни). Программа должна реализовывать симуляцию искусственной клеточной жизни, а также иметь понятный графический интерфейс.

Задание выдано

_____ г.

_____.

Е.В. Зубков

Задание принято

_____ г.

_____.

М.Р.Миниахметов

СОДЕРЖАНИЕ

Введение	4
1. Проектирование программного продукта	5
1.1 UML-диаграмма классов.....	5
1.2 UML-диаграмма последовательности.....	6
1.3 UML-диаграмма состояния.....	7
1.4 UML-диаграмма вариантов использования.....	8
2.Листинг программы.....	9
3.Результат выполнения программы.....	12
Заключение.....	15

Введение

Мною была создана программа «Cell Life Application» (Приложение клеточной жизни), которая даёт возможность наблюдения за развитием искусственной жизни.

Класс MainWindow (ГлавноеОкно) содержит в себе объект класса PictureBox (для показа изображения), объект класса Field (поле содержащее в себе клетки), объект класса FieldPrinter (формирует изображение поля), 2 таймера, 2 label (текстовые метки), полосу меню, всплывающие подсказки, 2 файловых окна, скрытую панель настройки параметров:

- screen – объект класса PictureBox показывающий изображение поля с клетками.
- field – объект класса Field, содержащий двумерный массив клеток и выполняющий генерацию нового поколения.
- printer – объект класса FieldPrinter, формирующий картинку.
- renderTimer – таймер, за один тик которого обновляется изображение.
- updateTimer – таймер, за один тик которого генерируется следующее поколение.
- F3info – текстовая метка показывающая некоторые важные параметры (данную текстовую метку можно скрыть или показать нажатием клавиши F3).
- pauseLabel – текстовая метка появляющаяся при паузе и скрывающаяся при возобновлении симуляции.
- полоса меню предоставляет пользователю дополнительные возможности взаимодействия с приложением, такие как сохранение и загрузка поля и его параметров, приближение/отдаление, регулировка скорости и т. п.
- всплывающие подсказки указывают на что влияет определённый параметр.
- скрытая панель настройки параметров появляется при нажатии клавиши “ё” или при выборе нужного пункта в полоске меню.

1.1 UML диаграмма классов

Данная диаграмма классов (рисунок 1) отображает основную часть классов программы, принадлежащие им атрибуты, операций и взаимосвязь между классами.

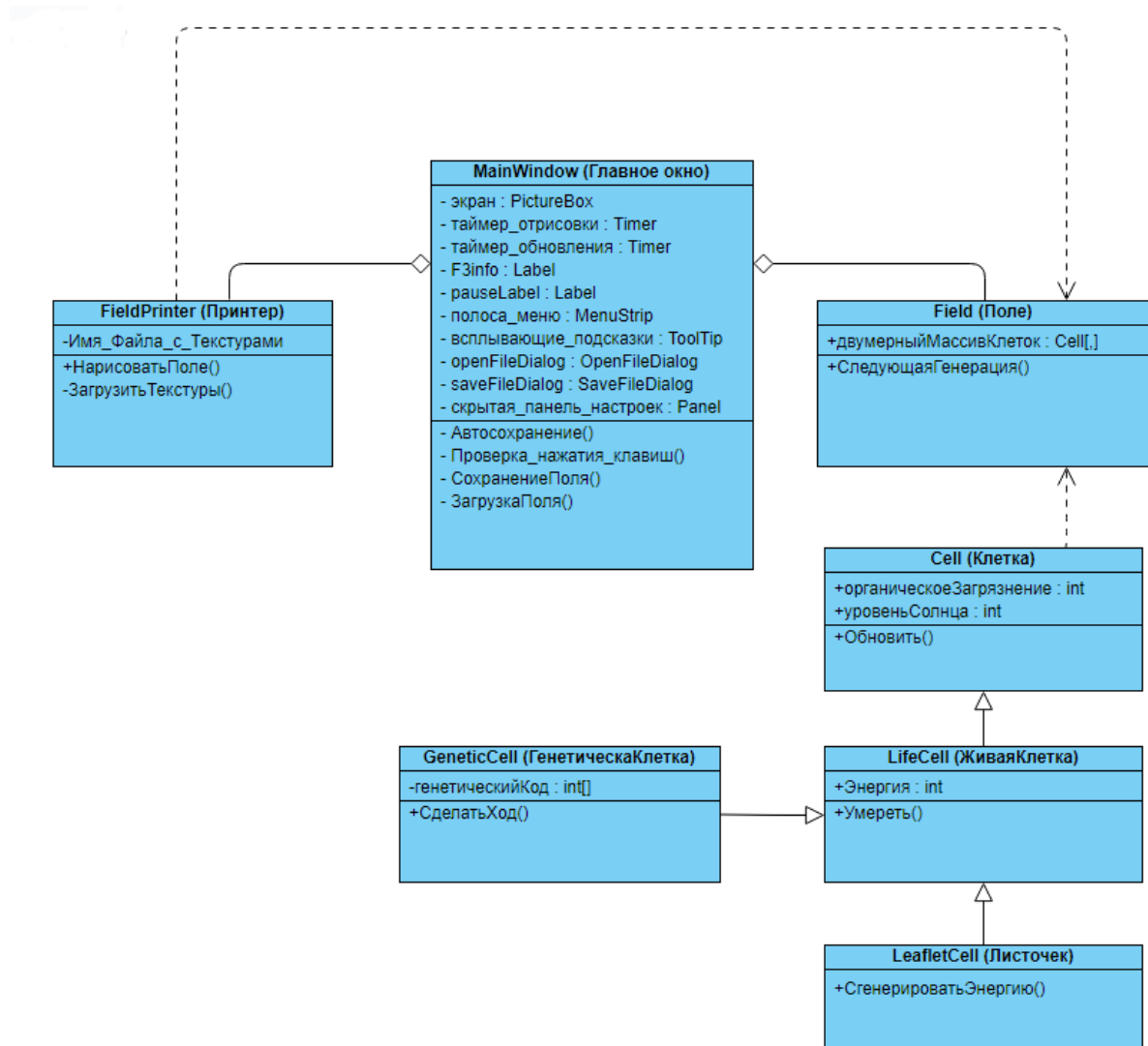


Рисунок 1 UML диаграмма классов

1.2 UML диаграмма последовательности

Диаграмма последовательностей (рисунок 2) отображает поведение системы и взаимодействие объектов системы во времени. Диаграмма также содержит объекты, которые взаимодействуют в рамках сценария, сообщения, которыми они обмениваются, а также результаты, связанные с сообщениями.

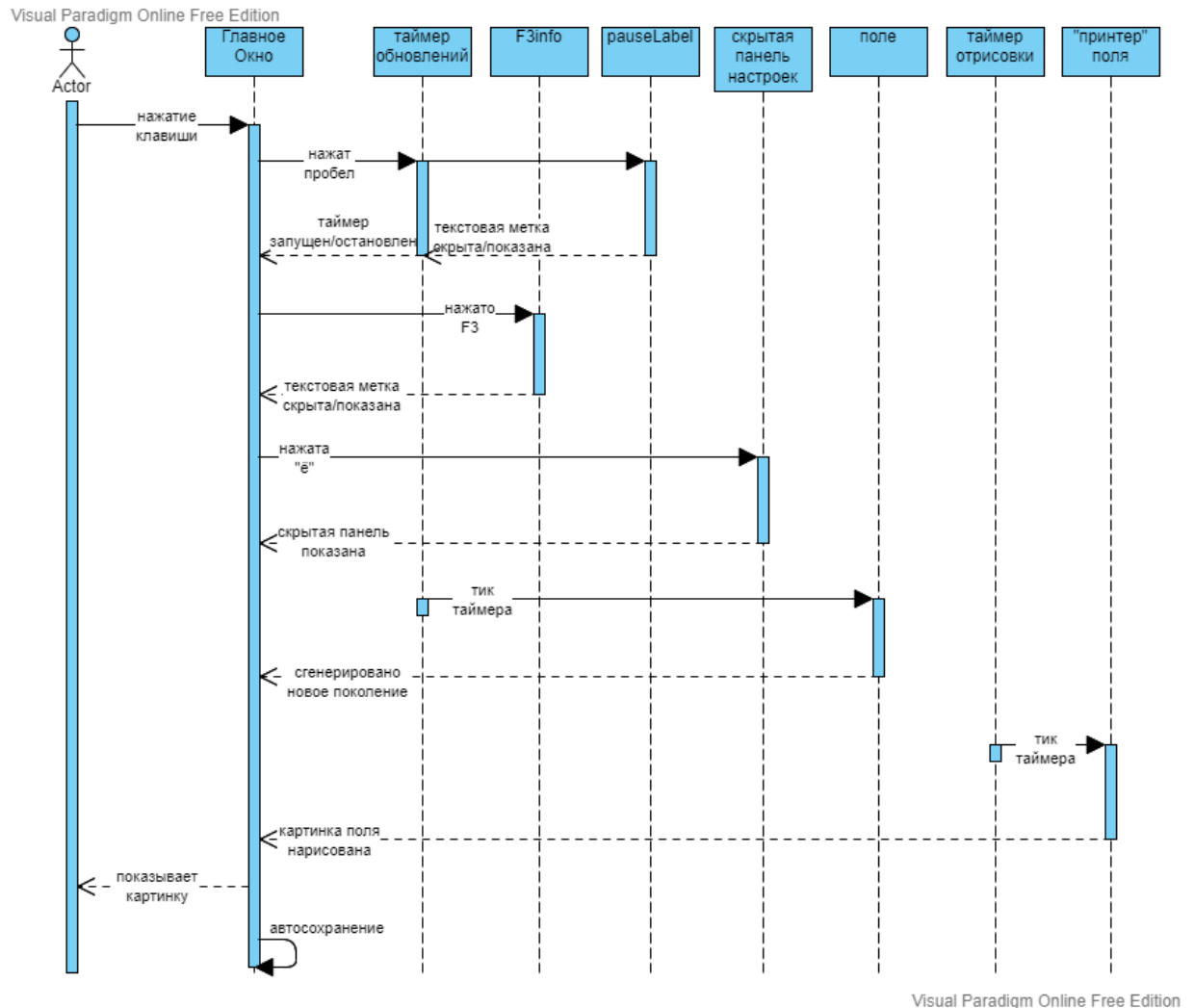
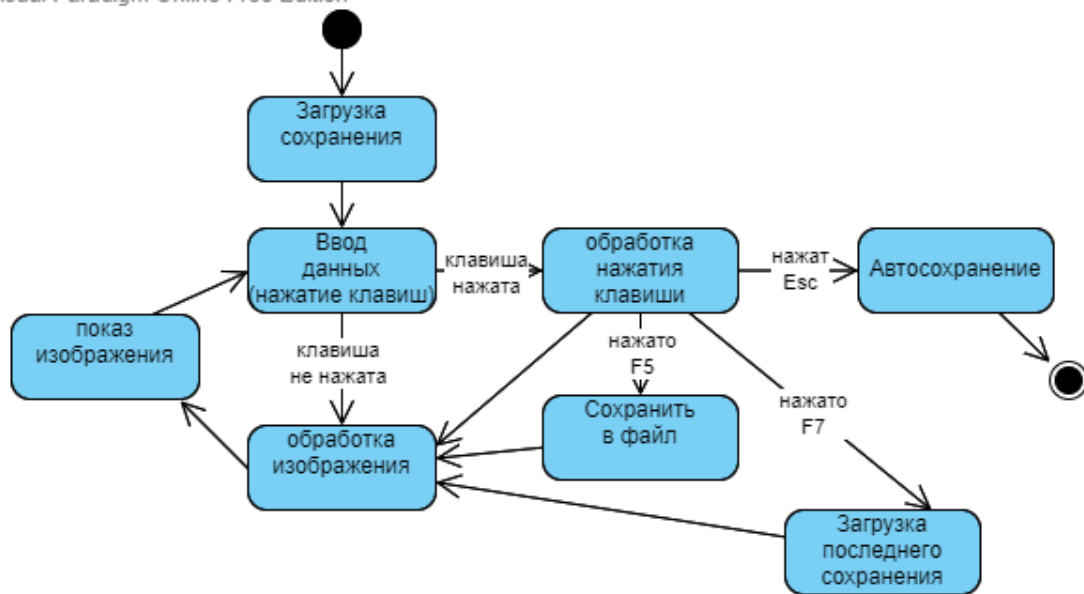


Рисунок 2 UML диаграмма последовательности

1.3 UML диаграмма состояния

Диаграмма состояния (рисунок 3) с помощью стрелок описывает переход объекта из одного состояния в другое.

Visual Paradigm Online Free Edition



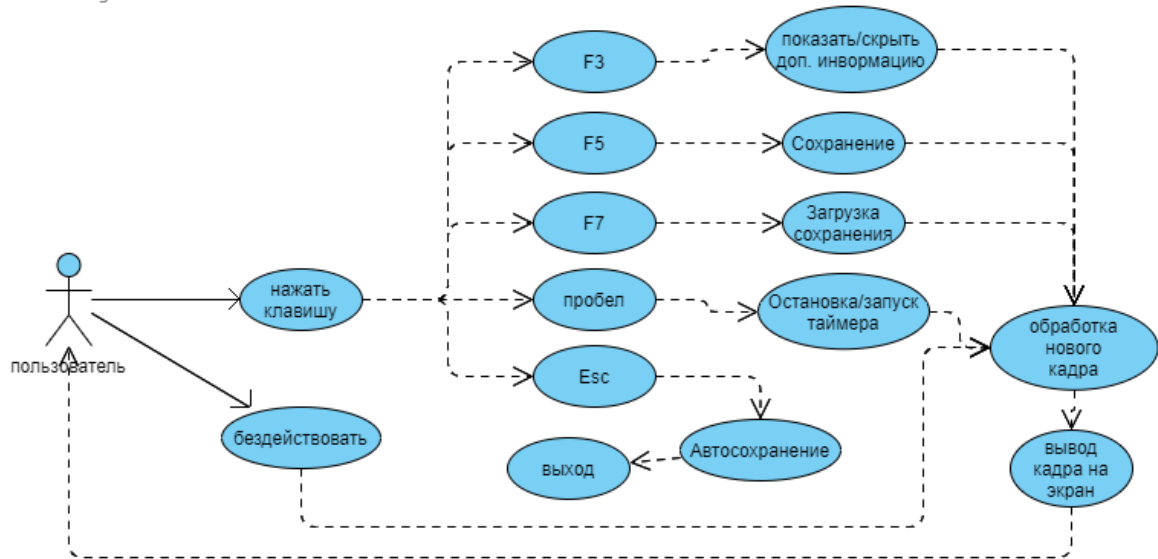
Visual Paradigm Online Free Edition

Рисунок 3 UML диаграмма состояния

1.4 UML-диаграмма вариантов использования

Диаграмма прецедентов (рисунок 4) отражает отношения между актерами и прецедентами. В данном случае показана взаимосвязь между актером и главным окном приложения.

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Рисунок 4 Диаграмма вариантов использования

2 Листинг программы

```
using System;
using System.Collections.Generic;
using System.Drawing;

namespace CellLifeApp.FieldClasses
{
    [Serializable]
    class Field
    {
        private Cell[,] field; // двумерный массив клеток
        // список генетических клеток на поле
        private HashSet<GeneticCell> geneticCells = new
HashSet<GeneticCell>();
        // список генетических клеток, подлежащих удалению
        internal HashSet<GeneticCell> removeList = new
HashSet<GeneticCell>();
        // список новых генетических клеток
        internal HashSet<GeneticCell> addList = new
HashSet<GeneticCell>();
        // объект класса Random для создания случайных чисел
        private Random random = new Random();
        // свойство, где CellParameters - класс, хранящий
параметры поля
        public CellParameters Parameters { get; private set; }
    }

    // счётчик поколений
    public int Generation { get; set; }
    // количество столбцов в поле
    public int Cols { get; private set; }
    // количество строк в поле
    public int Rows { get; private set; }
    // свойство возвращающее количество генетических
клеток на поле
    public int GeneticCellCount { get { return
geneticCells.Count; } }
    // обращение к полю по координатам (x, y) -
возвращает соответствующую клетку
    public Cell this[int x, int y]
    {
        get
        {
            if (Cols == 0 || Rows == 0)
```

```

        return null;
    x %= Cols;
    y %= Rows;
    if (x < 0)
        x += Cols;
    if (y < 0)
        y += Rows;
    return field[x, y];
}
set
{
    if (Cols == 0 || Rows == 0)
        return;
    x %= Cols;
    y %= Rows;
    if (x < 0)
        x += Cols;
    if (y < 0)
        y += Rows;
    field[x, y] = value;
}
}

// конструктор принимающий параметры поля
public Field(CellParameters parameters)
{
    Parameters = parameters;
    Cell.field = this;
    TestPattern();
}

// метод изменения размеров поля
private void SetFieldSize(int rows, int cols)
{
    Cols = cols;
    Rows = rows;
    field = new Cell[cols, rows];
}

// метод очистки поля
public void SetClearField()
{
    for (int x = 0; x < Cols; x++)
        for (int y = 0; y < Rows; y++)
            field[x, y] = new Cell(x, y);
}

```

```

        geneticCells.Clear();
        removeList.Clear();
        addList.Clear();
        GeneticCell.AllMutations = 0;
        Generation = 0;
    }
    // метод удаления ссылок на данную клетку у ее
соседей
    public void DelReferences(LifeCell cell, bool
shouldSetNewCell)...
        // метод повышения повышения загрязнения в клетке и
ее окружении
    public void IncreaseToxic(Cell cell)...
        // метод генерации нового поколения
    public void UpdateField()...
        // активация шаблона
    public void TestPattern() => TestPattern1();
        // первый шаблон
    public void TestPattern1()...
        // метод выстраивающий стены из "токсичных" клеток
    public void SetWalls()...
        // метод выстраивающий сектора из "токсичных" клеток
    public void SetSectorWalls(Size sectorSize, Point
sectorPos)...)
    }
}

```

3 Результат выполнения программы

- Описание интерфейса программы (рисунок 5)

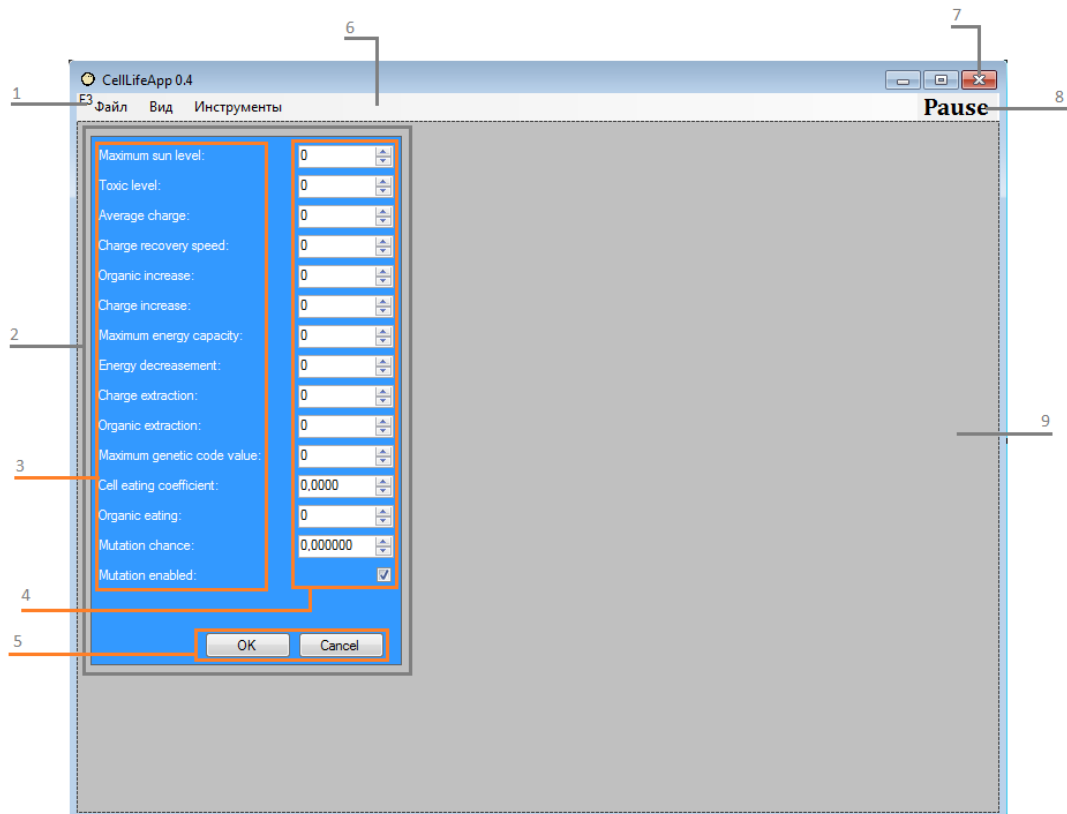


Рисунок 5 Внешнее описание

- 1) скрывающаяся текстовая метка, показывающая дополнительную информацию
- 2) скрытая панель настроек (показывается при выборе соответствующего пункта в полосе меню)
- 3) текстовые метки с подсказками (сообщает пользователю на что повлияет изменение данного параметра)
- 4) поля изменения настроек (параметров)
- 5) кнопки подтверждения/отмены изменений в параметрах
- 6) полоса меню
- 7) закрыть приложение
- 8) скрывающаяся текстовая метка (если видна, значит симуляция остановлена)
- 9) экран, на который выводится картина поля

- Запуск приложения, подгружается последнее сохранение (рисунок 6)

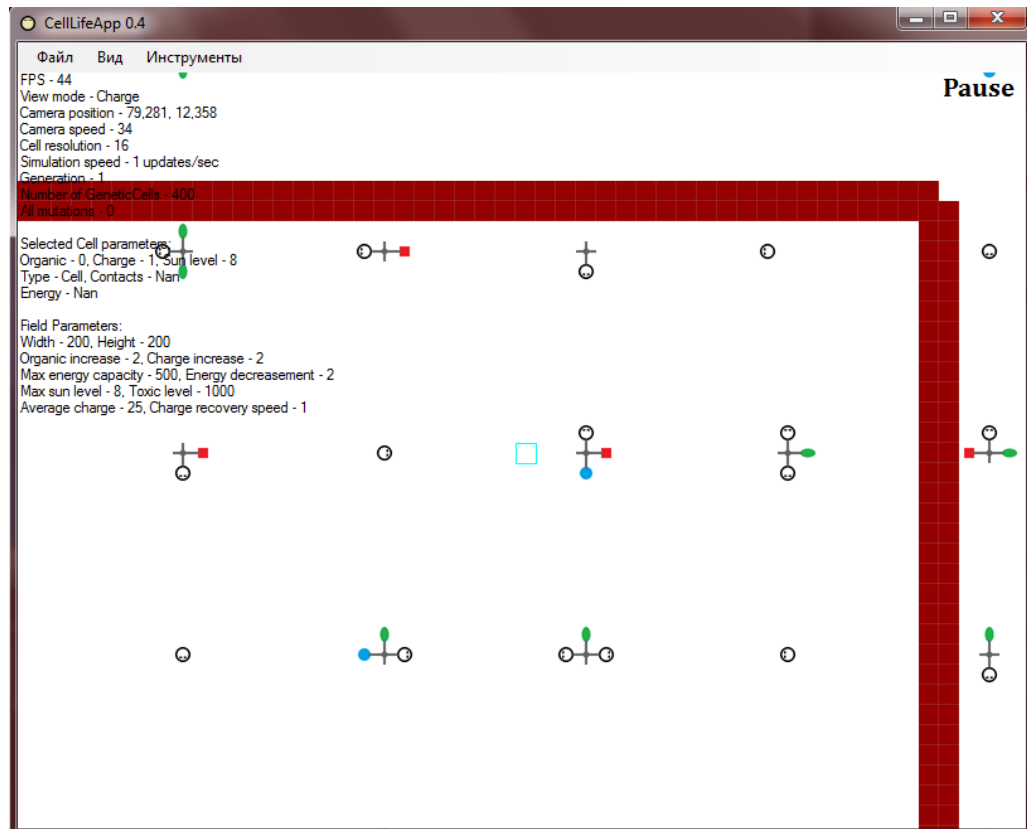


Рисунок 6 Запуск приложения

- Откроем панель настроек и изменим 1 параметр (рисунок 7)

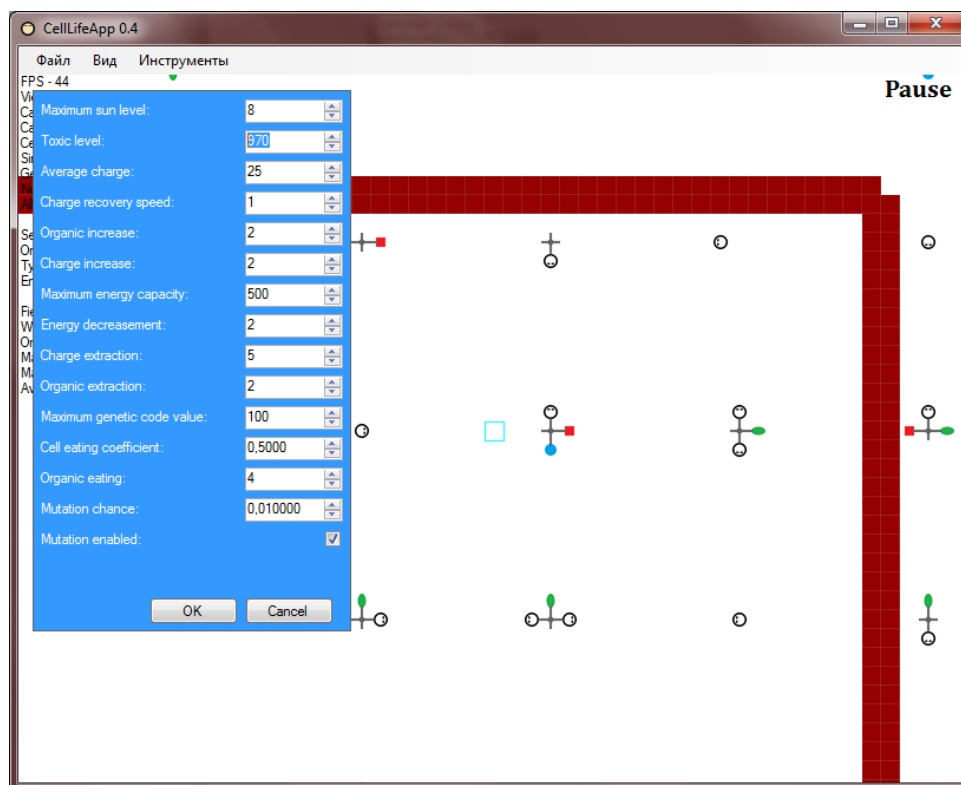


Рисунок 7 Изменение настроек

- Подтвердим изменения и увидим изменения в доп. информации (рисунок 8)

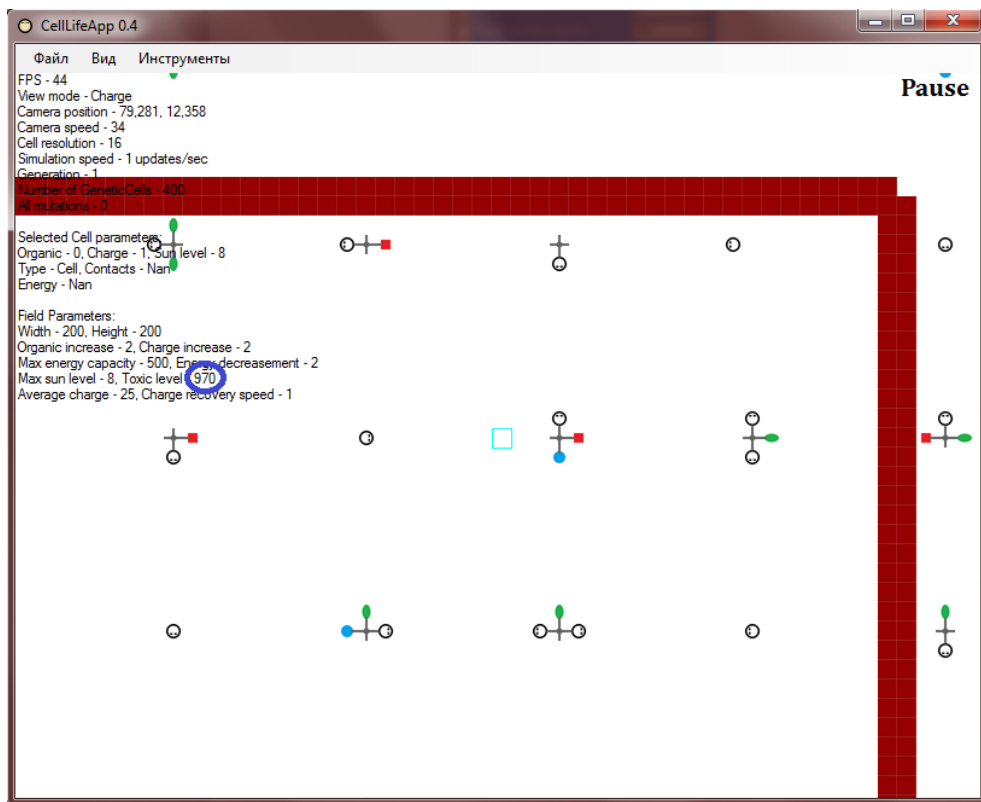


Рисунок 8 Результат изменений

- Скроем дополнительную информацию и запустим симуляцию (рисунок 9)

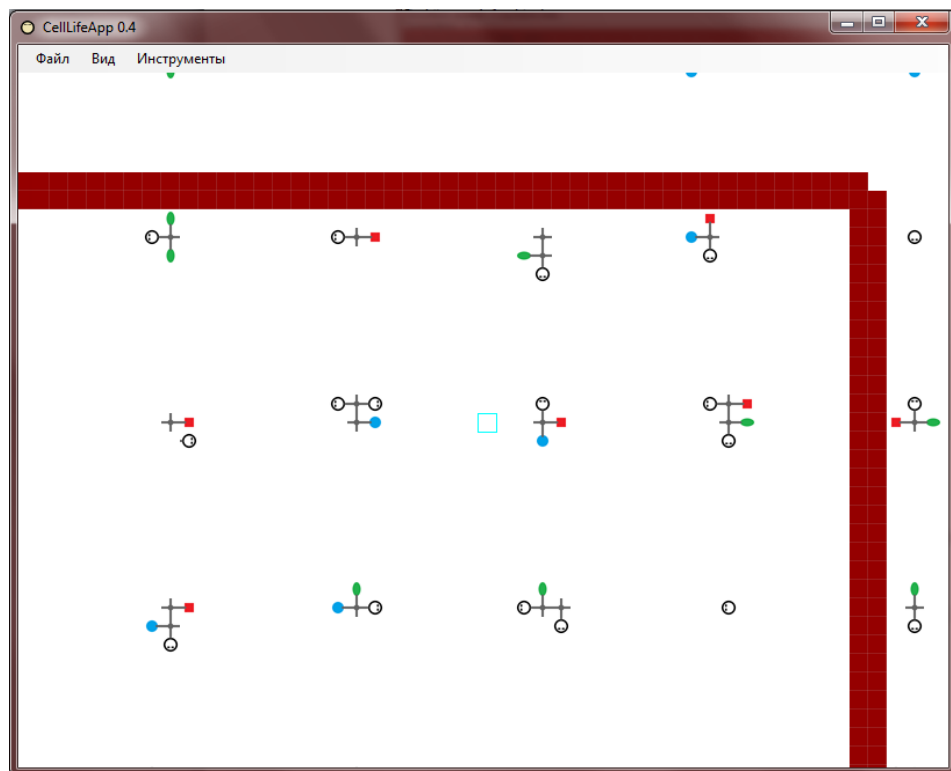


Рисунок 9 Симуляция

Заключение

В результате выполнения курсовой работы, было разработано приложение «Приложение клеточной жизни» (Cell Life Application). В программе реализована симуляция искусственной жизни на основе определённых правил. Также имеется удобный и понятный интерфейс взаимодействия с приложением. Программа позволяет наблюдать за искусственной жизнью, которая развивается путём естественного отбора (выживает наиболее приспособленный). Имеется возможность сохранять всё поле, чтобы открыть его потом и продолжить симуляцию. В будущем планируется добавить в приложение множество новых возможностей, таких как сохранение генетического кода отдельной клетки, редактирование поля (вставка генетической клетки с сохранённым генетическим кодом, удаление клеток или построение стен).

Исходя из этого была разработана программа «Приложение искусственной жизни», которая позволяет пользователю наблюдать за развитием искусственной жизни.