

# Клеточные автоматы. Игра «Жизнь».

# Основные понятия

- Пространство – покрытие одинаковыми **КЛЕТКАМИ**

(могут быть пронумерованы)

- Для каждой клетки определены **СОСТОЯНИЯ**

(в соответствие ставим число – целое или действительное)

# Клеточные автоматы

- под клеточными автоматами понимаются сети элементов, меняющих свое состояние в последовательные дискретные моменты времени по определенному закону в зависимости от того, каким было состояние рассматриваемого элемента и его соседей в предыдущий дискретный момент времени

# Применение

- Моделирование систем типа «реакция-диффузия»
- Моделирование процессов в нанотехнологиях
- Моделирование дорожного движения
- Модели теории перколяции (просачивания)

# «САМЫЕ ЗНАМЕНИТЫЕ»

- Исторически первый – автомат Винера – Розенблюта (опубликован в статье Винера, Розенблюта и Бигелоу)

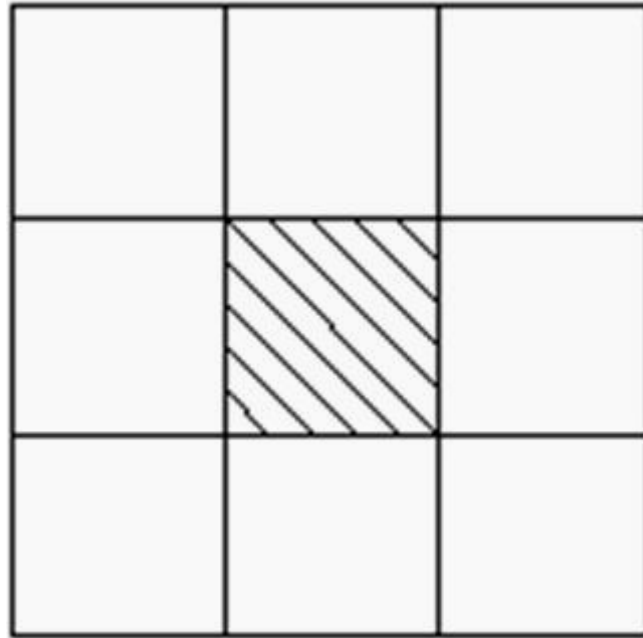
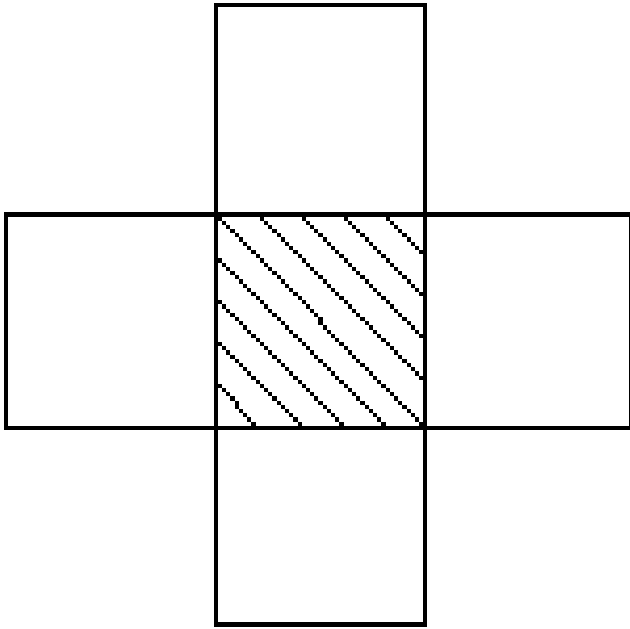
Автомат на плоскости

Состояния: 0 – покой,

1 – возбуждение, 2 – рефрактерность.

Окрестность - Неймана

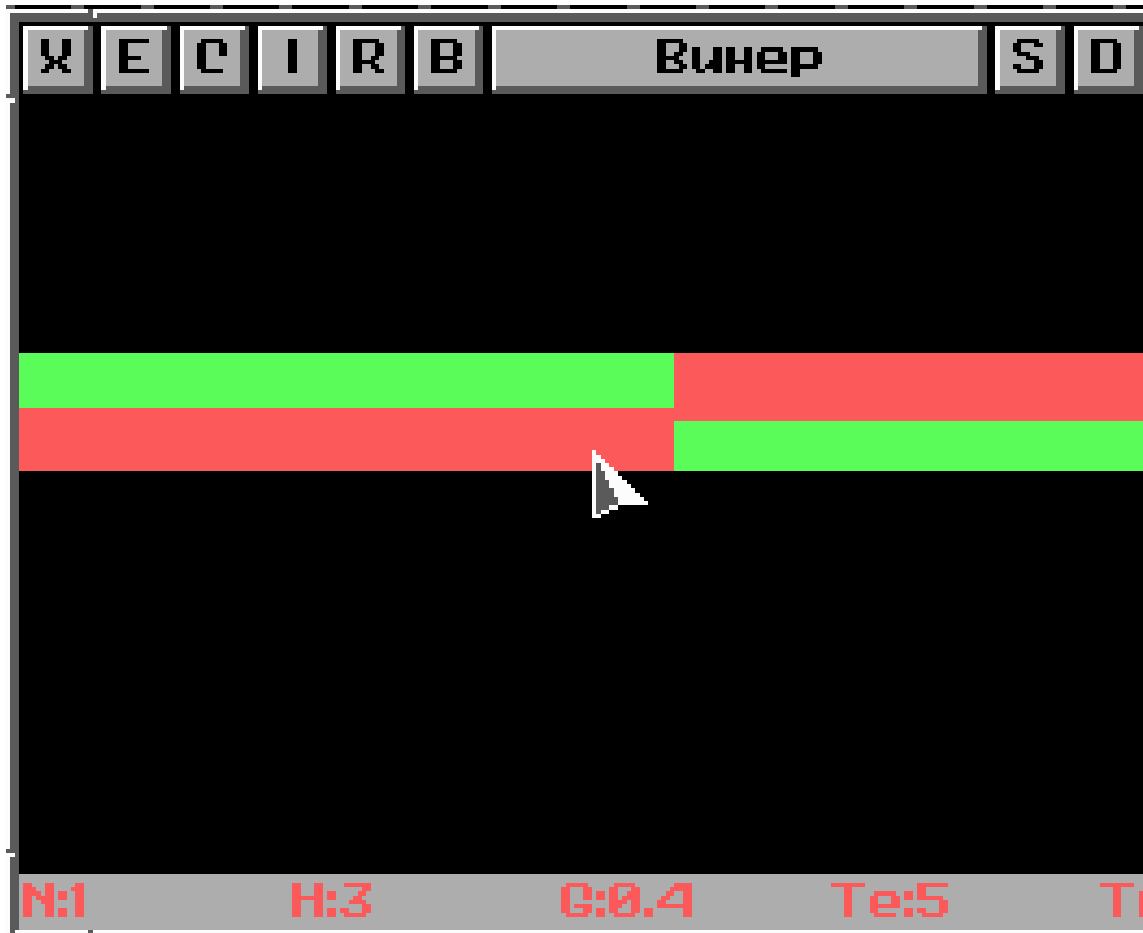
# Окрестности Неймана и Мура



# Автомат Винера-Рознблюта

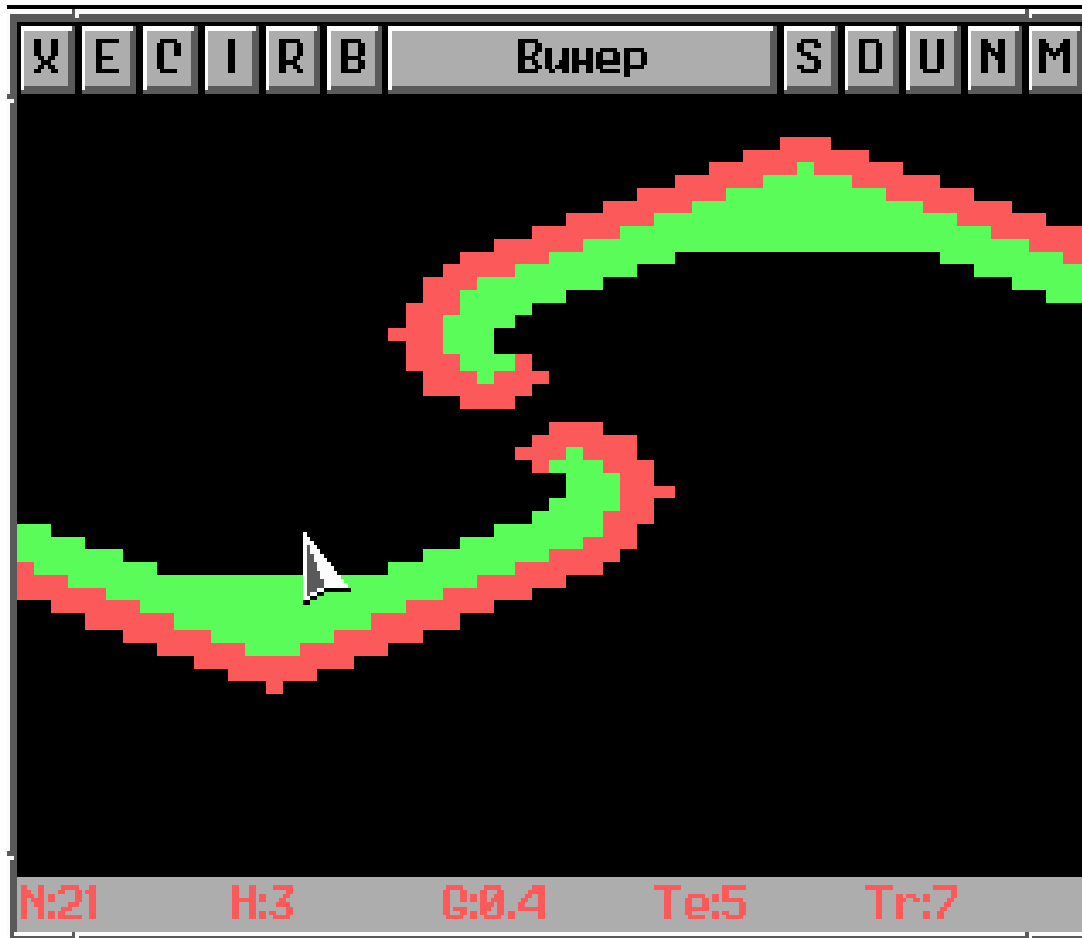
- Окрестность Неймана
- Правила перехода
- *1. Возбудиться может только невозбужденная клетка 0 (рефрактерные клетки 2 не возбуждаются), если в ее окрестности в момент времени  $t$  есть возбужденная клетка 1*
- *2. Через время  $t_1$  возбужденные клетки переходят в состояние рефрактерности.*
- *3. Через время  $t_2$  рефрактерные клетки переходят в состояние покоя.*

# Спиральные волны

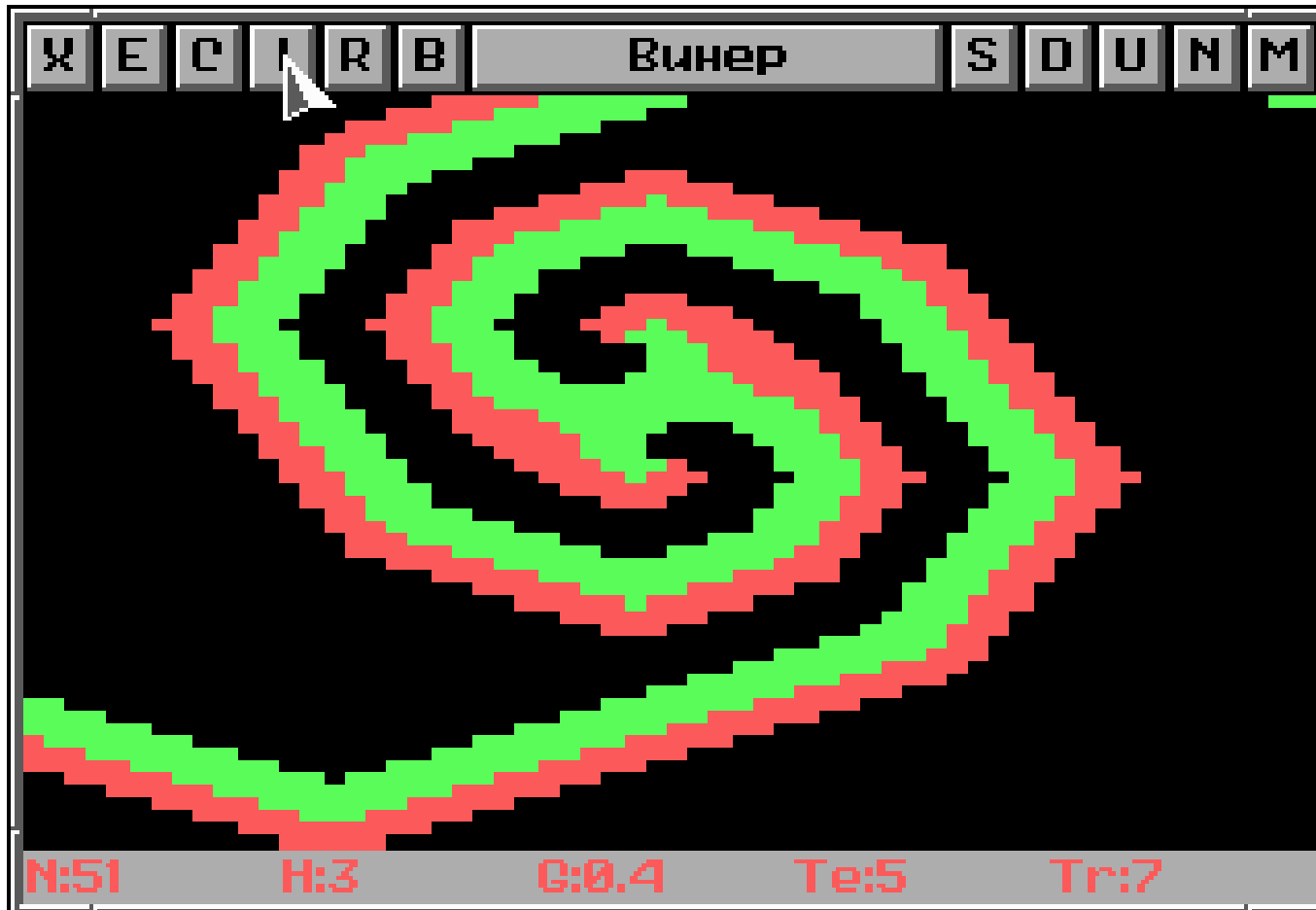




# Спиральные волны



# Спиральные волны



# Самый-самый знаменитый...

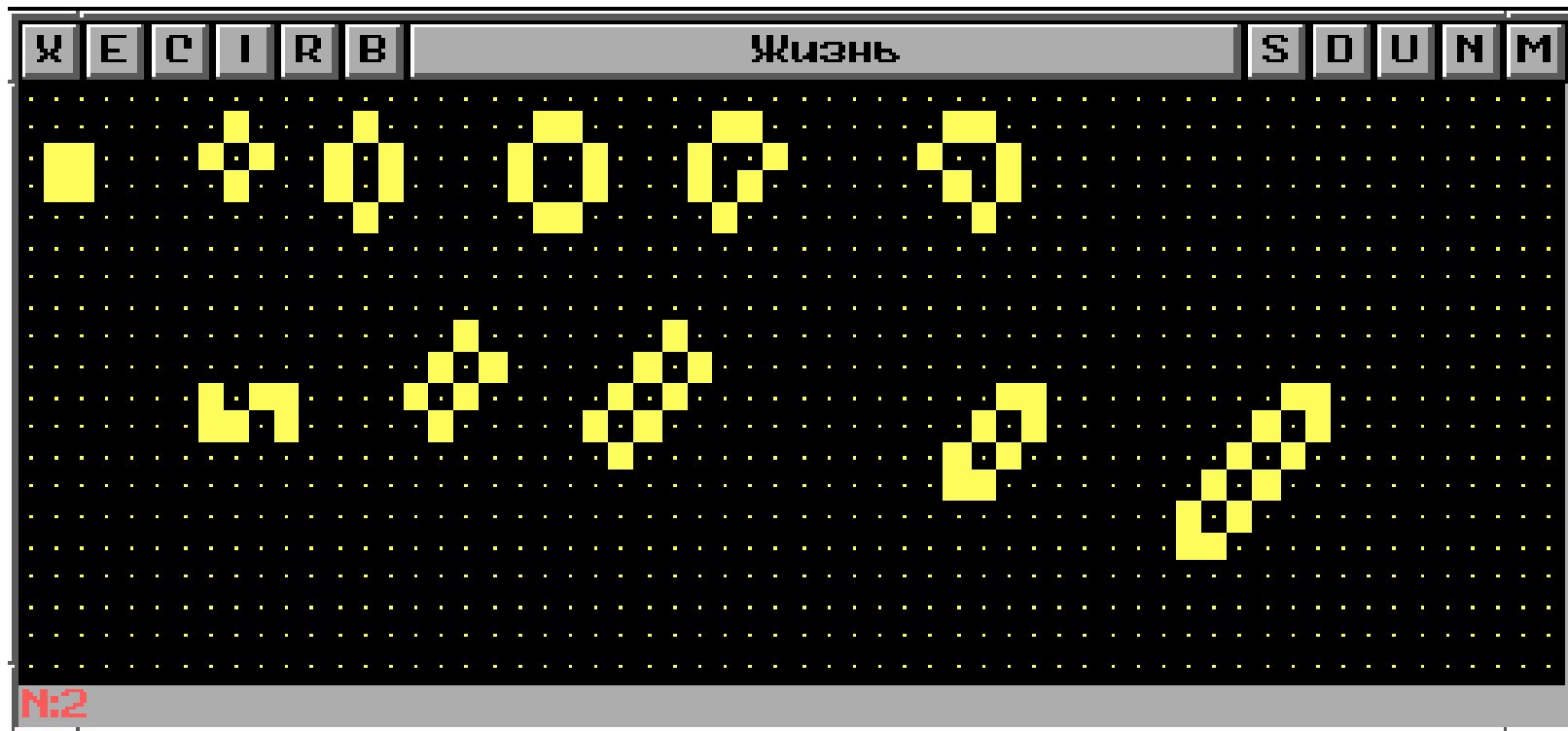
- элемент может находиться в состоянии покоя (0) или активности (1). Из состояния покоя в активное в следующем поколении элемент переходит, если рядом с ним в текущем поколении оказалось ровно три активных элемента. Состояние активности сохраняется, если среди ближайших соседей находится два или три активных элемента.

# Самый-самый знаменитый

- Правила перехода на следующий слой могут быть записаны в канонической форме  $B3/S2,3$ .
- B – born
- S - save

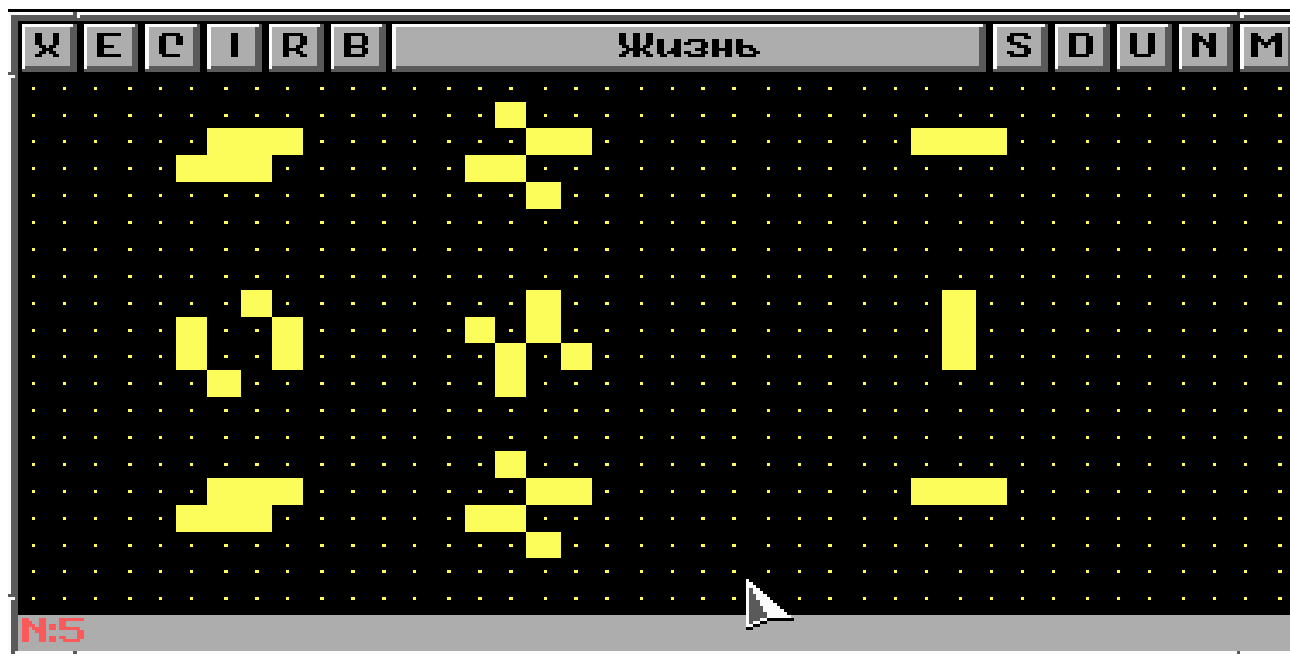
# Игра Дж. Конвея «Жизнь»

- . Стационарные структуры в игре «Жизнь». Структура в верхнем левом углу состоит из четырех клеток



# Игра Дж. Конвея «Жизнь»

- 2-циклы в три последовательных момента времени (сверху вниз). Цикл в третьей колонке — семафор



Зачетный проект

# Проект трехмерная «Жизнь»

- Трехмерная кубическая решетка с значениями 0 или 1
- Размер поля задан в начале и остается постоянным на все время расчета
- Граничные условия — периодические (тор)
- Входные данные — файл со списком «живых» клеток, размер поля
- Выходные данные — файл заданного формата со списком «живых» клеток на каждом временном шаге



# ЖИЗНЬ

Возможные значения клеток:  $\{0, 1\}$

Рождение клетки – переход клетки из состояния 0 в 1 – происходит при числе соседей 6 или 7

Сохранение клетки – переход из 1 в 1 – происходит при числе соседей 4, 5, 6 или 7

В других случаях клетка умирает – переходит в 0

$$S(x, y, z, t) = \sum_{i=x\pm 1, j=y\pm 1, k=z\pm 1} F(i, j, k, t)$$

$$F(x, y, z, t+1) = \begin{cases} 1, & (F(x, y, z, t) = 1 \wedge 4 \leq S(x, y, z, t) \leq 7) \vee (F(x, y, z, t) = 0 \wedge 6 \leq S(x, y, z, t) \leq 7) \\ 0, & \text{иначе} \end{cases}$$

# Формат входного файла

**WX WY WZ T**

**w1 x1 y1 z1**

**...**

**wm xm ym zm**

**0 0 0 0**

WX, WY, WZ – размеры поля, натуральные числа

T – тип симуляции:

1 – “Жизнь”

w<sub>i</sub> – тип i-й клетки (для “Жизни” - 1, для “К-О” - 1 или 2, где 2 обозначает клетки типа M)

x<sub>i</sub>, y<sub>i</sub>, z<sub>i</sub> – координаты i-й клетки

# Формат выходного файла

**w11 x11 y11 z11**

...

**w1m x1m y1m z1m**

**0 0 0 0**

...

**wn1 xn1 yn1 zn1**

...

**wnk xnk ynk znk**

**0 0 0 0**

$w_{ji}$  – тип  $i$ -й клетки на  $j$ -м шаге (для “Жизни” - 1)

$x_{ji}$ ,  $y_{ji}$ ,  $z_{ji}$  – координаты  $i$ -й клетки на  $j$ -м шаге

# Пример входных данных

20 20 20 1

1 5 7 5

1 5 8 5

1 6 8 5

1 7 5 5

1 8 5 5

1 8 6 5

1 5 7 6

1 5 8 6

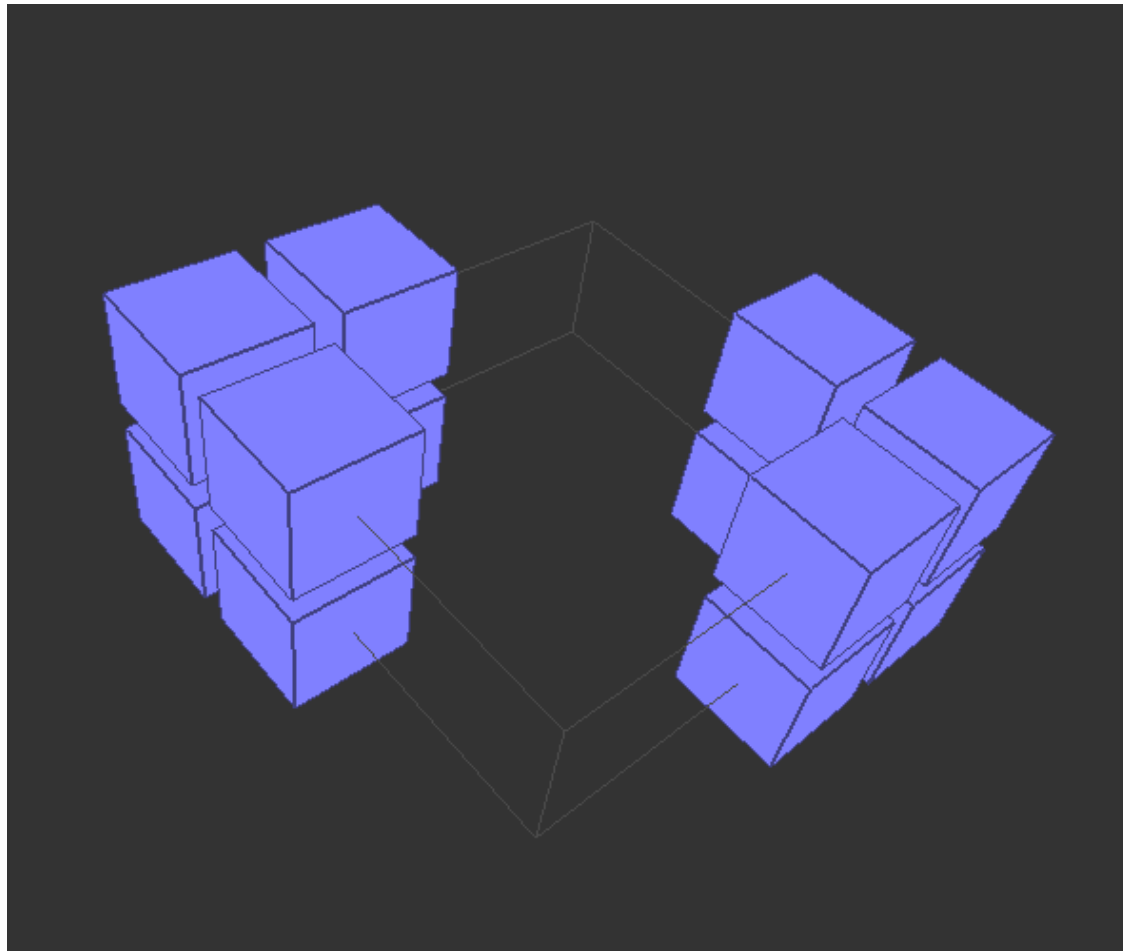
1 6 8 6

1 7 5 6

1 8 5 6

1 8 6 6

0 0 0 0



# Параметры запуска

**mpirun -np 3 ./life test.in 100 5 test.out**

- test.in — файл с входными данными
- 100 — общее число шагов
- 5 — какие шаги сохранять, определяется из условия  $i \% 5 == 0$ , при цикле от 0
- test.out — файл с результатами расчета, если файл существует, все данные перезаписываются
- Программа должна работать при **любом** числе потоков и выдавать одинаковый результат

# Параметры тестирования

- Несколько тестовых прогонов на различных входных данных
- Различное число шагов
- Различные шаги сохранения
- Замер времен работы в несколько потоков и наличие ускорения
- Идентичность работы при любом числе шагов

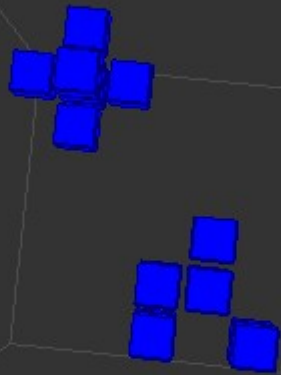
# Критерий оценки

- Хранение расчетной области (локальное или распределенное)
- Пересылка всей сетки или только приграничных областей
- Типы используемых пересылок
- Деление по одной из оси или по нескольким
- Масштабируемость при большом числе потоков
- Общая скорость работы

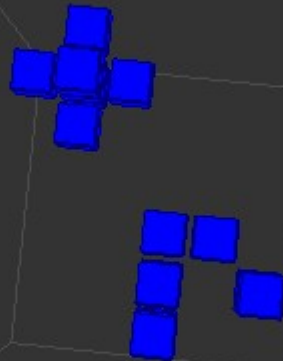
Пример работы



34Hz



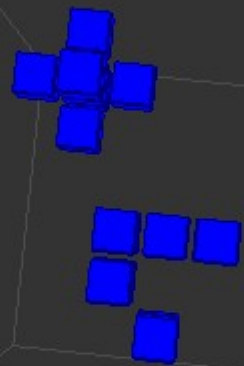
34Hz



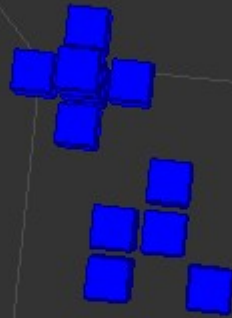
34Hz



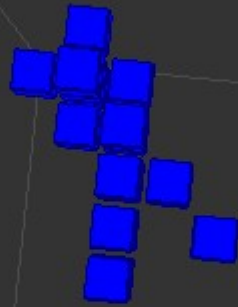
34Hz



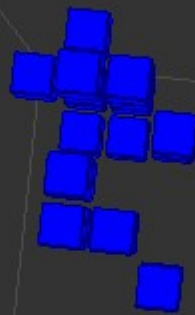
34Hz



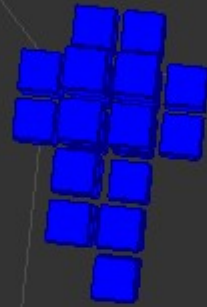
34Hz



34Hz

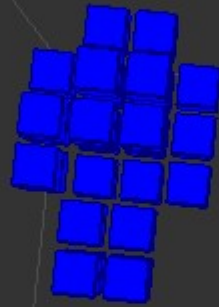


34Hz

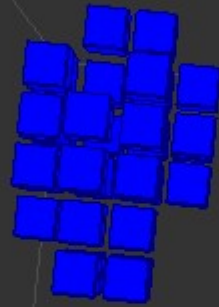




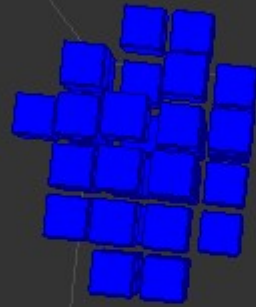
34Hz



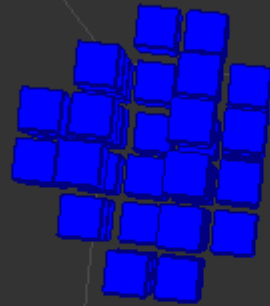
28Hz



28Hz



28Hz



Вопросы