

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

КАФЕДРА КОМПЬЮТЕРНЫХ СИСТЕМ И ПРОГРАММНЫХ ТЕХНОЛОГИЙ

Отчёт по лабораторной работе №5

Курс: «Многоагентные системы»

Выполнил студент:

Ерниязов Т.Е.

Группа: 13541/2

Проверил:

Сазанов А.М.

Санкт-Петербург
2018 г.

Содержание

1	Лабораторная работа №5	2
1.1	Цель работы	2
1.2	Программа работы	2
1.3	Ход работы	3
1.3.1	Определение и краткая классификация многоагентных интеллектуальных систем. Преимущества и в чем недостатки многоагентного подхода.	3
1.3.2	Приведите наиболее полную классификацию таких систем, кратко поясните по какому признаку дана эта классификация.	4
1.3.3	Какие задачи решаются с помощью многоагентного подхода. Приведите не менее ДВУХ примеров задач, с кратким описанием.	5
1.3.4	Проанализируйте преимущества и недостатки многоагентного подхода на примере задач из предыдущего пункта.	5
1.3.5	Анализ системы NetLogo и обзор основных функциональных возможностей	5
1.4	Вывод	18
1.5	Список литературы	20

Лабораторная работа №5

1.1 Цель работы

Познакомиться с принципами работы многоагентных систем, а также изучить основные возможности основных платформ для разработки таких систем.

1.2 Программа работы

1. Дайте определение и краткую классификацию многоагентных интеллектуальных систем. В чем преимущества и в чем недостатки многоагентного подхода?
2. Приведите наиболее полную классификацию таких систем, кратко поясните по какому признаку дана эта классификация.
3. Какие задачи решаются с помощью многоагентного подхода. Приведите не менее ДВУХ примеров задач, с кратким описанием.
4. Проанализируйте преимущества и недостатки многоагентного подхода на примере задач из предыдущего пункта. Сформулируйте общие проблемы \недостатки и общие преимущества \достоинства такого подхода.
5. Проанализировать основные платформы для разработки многоагентных систем: выбрать одну из платформ (Например, NetLogo, StarLogo, Repast Simphony, Eclipse AMP, JADE, Jason либо другую) и провести обзор основных функциональных возможностей:
 - Процесс установки ПО, ссылки на сайты, ссылки на необходимые драйвера и т.д. (если нужно)
 - Процесс создания простого проекта.
 - Анализ одного примера (ссылка на пример, описание алгоритма работы и процесса моделирования и т.д.)
 - Опишите основные возможности данного ПО применительно для создания многоагентных систем. Опишите замеченные недостатки или наоборот опишите достоинства данного ПО.
6. Написать выводы. В выводах отразить, помимо своих мыслей, возникших в ходе работы, ответы на приведенные ниже вопросы:
 - В чем Плюсы и минусы многоагентного подхода?
 - Какие еще среды или языки программирования использует для создания многоагентных систем?
 - Как по-вашему стоит ли развивать данное направление, если нет, то почему, если да, то в какую сторону?
 - Корректно ли по-вашему моделирование многоагентных систем на одной вычислительной машине (рассмотреть два варианта, итеративный перебор агентом в цикле, и создание для каждого агента своего процесса)
 - Приведите области \примеры в которых применение многоагентного подхода дает максимально положительные результаты.

1.3 Ход работы

1.3.1 Определение и краткая классификация многоагентных интеллектуальных систем. Преимущества и в чем недостатки многоагентного подхода.

Многоагентные системы – это направление искусственного интеллекта, которое для решения сложной задачи или проблемы использует системы, состоящие из множества взаимодействующих агентов [1]

Агент — это аппаратная или программная сущность, способная действовать в интересах достижения целей, поставленных пользователем. Агенты описываются рядом свойств, которые характеризуют понятие агента:

- **Автономность** – способность функционировать без прямого вмешательства людей или компьютерных средств и при этом осуществлять самоконтроль над своими действиями и внутренними состояниями;
- **Реактивность** – способность воспринимать состояние среды (физического мира, пользователя – через пользовательский интерфейс, совокупности других агентов, сети Internet, или сразу все этих компонентов внешней среды);
- **Активность** – способность агентов не просто реагировать на стимулы, поступающие из среды, но и осуществлять целенаправленное поведение, проявляя инициативу.

В отличие от классической теории искусственного интеллекта, где агент (ИС) имеет в своем распоряжении все необходимые данные и вычислительные ресурсы, в теории многоагентных систем один агент владеет всего лишь частичным представлением о глобальной проблеме, а значит, он может решить лишь некоторую часть общей задачи.

Общая классификация агентов

В наиболее общем смысле, многоагентные системы принято различать по их архитектуре:

- **Делиберативная архитектура** – Подход на основе символической модели мира, в которой решения принимаются путем логических рассуждений. Такие агенты называются агентами, основанными на знаниях. Для представления знаний используются хорошо известные в системах искусственного интеллекта методы - логический, продукционный, семантический, фреймовый [2]
- **Реактивная архитектура** – Подход основывается на том, что интеллектуальное поведение может быть обеспечено без использования символической модели мира, путем формирования реакций агента на события внешнего мира [2]
- **Гибридная архитектура** – объединяет преимущества как делиберативной, так и реактивной архитектур.

Кроме того, в качестве общей классификации многоагентных систем следует выделить области, в которых эти системы применяются:

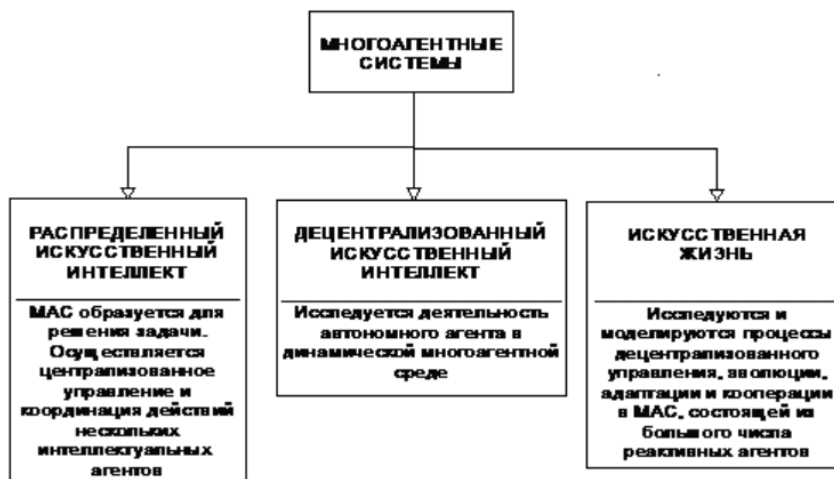


Рис. 1.1: Классификация многоагентных систем по областям применения [3]

Преимущества и в чем недостатки многоагентного подхода

Основные преимущества использования многоагентных систем, это: **распределенные вычисления, масштабирование и автономность**. Кроме того агенты могут общаться между собой, обмениваться информацией и кооперироваться для выполнения задач.

Из недостатков следует выделить усложнение структуры за счет количества агентов, что впоследствии ведет к **плохой наблюдаемости и непредсказуемости** системы.

1.3.2 Приведите наиболее полную классификацию таких систем, кратко поясните по какому признаку дана эта классификация.

Приведем наиболее подробную классификацию агентов:

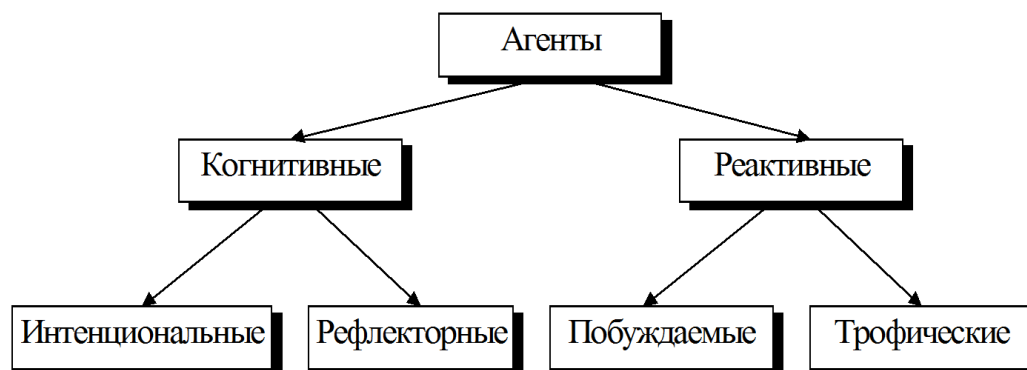


Рис. 1.2: Классификация агентов

Интеллектуальные агенты обладают хорошо развитой и пополняемой символьной моделью внешнего мира, что достигается благодаря наличию у них базы знаний, механизмов решения и анализа действий. Небольшое различие между этими типами интеллектуальных агентов связано с расстановкой акцентов на тех или иных интеллектуальных функциях: либо на получении знаний о среде, либо на рассуждениях о возможных действиях. У коммуникативных агентов внутренняя модель мира превращается главным образом в модель общения, состоящую из моделей участников, процесса и желаемого результата общения. Наконец, база знаний ресурсного агента содержит в основном знания о структуре и состоянии ресурсов, определяющих различные формы поведения [5]

Далее, по типу поведения интеллектуальные агенты делятся на **интенциональных** и **рефлекторных**. Большинство интеллектуальных (когнитивных) агентов можно отнести к числу интенциональных. Подобные агенты наделены собственными механизмами мотивации. Это означает, что в них так или иначе моделируются внутренние убеждения, желания, намерения и мотивы, порождающие цели, которые и определяют их действия. В свою очередь, модульные или рефлекторные агенты не имеют внутренних источников мотивации и собственных целей, а их поведение характеризуется простейшими (одношаговыми) выводами или автоматизмами [5]

Реактивные агенты не имеют ни сколько-нибудь развитого представления внешней среды, ни механизма многошаговых рассуждений, ни достаточного количества собственных ресурсов. Отсюда вытекает еще одно существенное различие между интеллектуальными и реактивными агентами, связанное с возможностями прогнозирования изменений внешней среды и, как следствие, своего будущего. В силу вышеуказанных недостатков реактивные агенты обладают очень ограниченным диапазоном предвидения. Они практически не способны планировать свои действия, поскольку реактивность в чистом виде означает такую структуру обратной связи, которая не содержит механизмов прогноза. Тогда как интеллектуальные агенты, благодаря богатым внутренним представлениям внешней среды и возможностям рассуждений, могут запоминать и анализировать различные ситуации, предвидеть возможные реакции на свои действия, делать из этого выводы, полезные для дальнейших действий и, в результате, планировать свое поведение [5]

По сложности этих реакций и происхождению источников мотивации реактивные агенты подразделяются на **побуждаемых** и **трофических** агентов. В случае трофических агентов поведение определяется простейшими трофическими связями. Типичной моделью подобных агентов являются клеточные автоматы, где основными параметрами выступают: радиус восприятия агента, количество условных единиц питания во внешней среде и энергетическая стоимость единицы. Между тем, побуждаемые агенты также могут иметь примитивный механизм мотивации, толкающий их на выполнение задачи, например, удовлетворение набора жизненных потребностей. Речь идет о поддержании энергетического баланса или, в более широком плане, об условиях выживания агента [5]

Таблица 1.1: Сравнительная характеристика когнитивных и реактивных агентов [5]

Характеристики	Когнитивные агенты	Реактивные агенты
Внутренняя модель внешнего мира	Развитая	Примитивная
Рассуждения	Сложные	Простые
Мотивация	Развитая	Простая
Память	Есть	Нет
Реакция	Медленная	Быстрая
Адаптивность	Низкая	Высокая
Модельная архитектура	Есть	Нет
Состав многоагентной системы	Небольшое	Большое

В соответствии с этой классификацией структурируем данные о когнитивных и реактивных агентах в виде таблицы:

1.3.3 Какие задачи решаются с помощью многоагентного подхода. Приведите не менее ДВУХ примеров задач, с кратким описанием.

Примеры использования многоагентных систем:

- Многоагентные системы широко используются при моделировании естественнонаучных и социальных процессов с целью выявления макрозакономерностей.
- Многоагентные системы хорошо зарекомендовали себя в сфере сетевых и мобильных технологий, для обеспечения автоматического и динамического баланса нагрузки, расширяемости и способности к самовосстановлению.

1.3.4 Проанализируйте преимущества и недостатки многоагентного подхода на примере задач из предыдущего пункта.

Достоинства использования многоагентного подхода при моделировании естественнонаучных и социальных процессов очевидны: как правило в этих областях моделируется большое количество различных распределенных агентов (атомы, клетки, нейроны, животные, люди и т.д.), каждый из которых имеет представление только о своей задаче и может взаимодействовать с другими агентами. Таким образом все преимущества многоагентного подхода, указанные выше справедливы и для этих областей.

Из недостатков многоагентного подхода при моделировании естественнонаучных и социальных процессов следует выделить непредсказуемость этих систем, так как наращивание количества таких распределенных агентов может быть очень быстрым в зависимости от начальных критериев.

То же самое справедливо для сферы сетевых и мобильных технологий, где количество агентов чаще всего не такое большое, но каждый из объектов обладает собственным поведением, не знает о состоянии всей системы и может взаимодействовать с другими агентами. Для данного примера не будет недостатка связанного с неконтролируемым наращиванием количества агентов, так как они чаще всего уже заданы изначально.

1.3.5 Анализ системы NetLogo и обзор основных функциональных возможностей

Среда программирования NetLogo служит для моделирования ситуаций и феноменов, происходящих в природе и обществе. NetLogo удобно использовать для моделирования сложных, развивающихся во времени систем. Создатель модели может давать указания сотням и тысячам независимых агентов действующим параллельно. Это открывает возможность для объяснения и понимания связей между поведением отдельных индивидуумов и явлениями, которые происходят на макро уровне [6]

Язык NetLogo достаточно прост и ученики и учителя могут создавать в этой среде свои собственные авторские модели. В то же время это достаточно мощный язык и среда для проведения исследовательских работ. Библиотека NetLogo содержит множество готовых моделей по биологии, математике, химии, социологии. С этими моделями могут ознакомиться и поиграть ученики.

Процесс установки ПО

NetLogo распространяется бесплатно и может быть загружен с официального сайта [7] для операционных систем Windows, Linux, Mac OS. Перед скачиванием установщика предлагается заполнить следующую форму:

Download NetLogo

Most computers can run NetLogo (see [system requirements](#)). If you would like to run NetLogo on a Chromebook or in a web browser, please see [NetLogo Web](#) which meets your needs.

Multiple versions of NetLogo can be installed on the same computer; installing a new one doesn't remove the old one.

Version: [More versions here](#)
For help using old models with new versions see the [Transition Guide](#)

Name:

Organization:

E-mail:

Comments:


We read these but don't respond directly. For a response, write feedback@netlogo.cwi.nl

Download trouble? Write bugs@netlogo.cwi.nl

Форма заполнения

NetLogo 6.0.4 Downloads

June 14, 2018

	Mac OS X	Download (209 MB)
	Windows (32-bit)	Download (181 MB)
	Windows (64-bit)	Download (183 MB)
	Linux (32-bit)	Download (203 MB)
	Linux (64-bit)	Download (200 MB)

Выбор ОС

После этого откроется страница, с ссылками на файл загрузчик для разных операционных систем.

Дальнейший процесс установки тривиален: для Windows и Mac OS типичный файл загрузчик, а для Linux распаковываемый архив.

Процесс создания простого проекта.

Проект может быть создан или открыт комбинациями клавиш Ctrl+N (cmd+N), Ctrl+O (cmd+O) соответственно или из меню приложения. Кроме того проект может быть импортирован из библиотеки демонстрационных проектов комбинацией клавиш Ctrl+M (cmd+M) или из меню приложения (Файл-> Библиотека моделей).

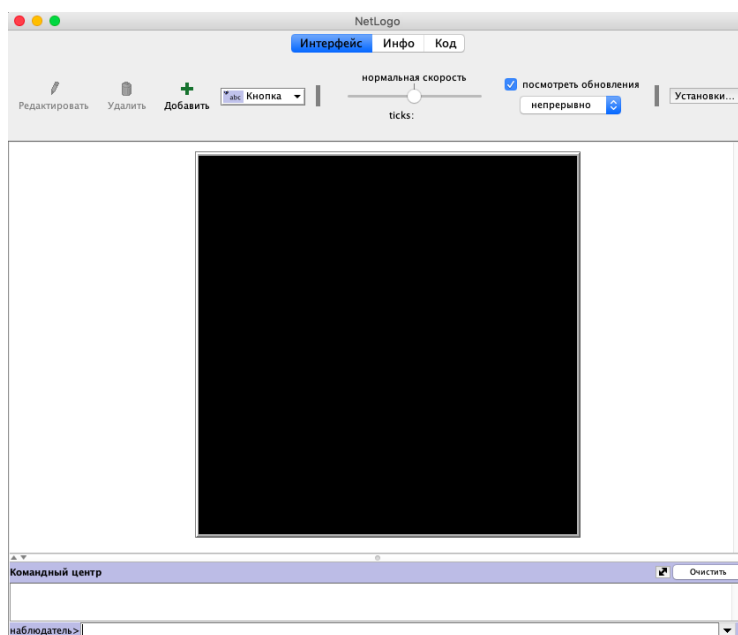


Рис. 1.3: Пустой проект

После этого не нужно предпринимать каких-либо дополнительных действий и проект будет доступен для редактирования.

Анализ одного примера

В качестве примера возьмем демонстрационный проект из сферы биологии демонстрирующий передачу и сохранение вируса в изолированной системе. Данный проект может быть найден в библиотеке демонстрационных проектов в папке SampleModels -> Alternative Visualizations -> Virus - Circle Visualization.

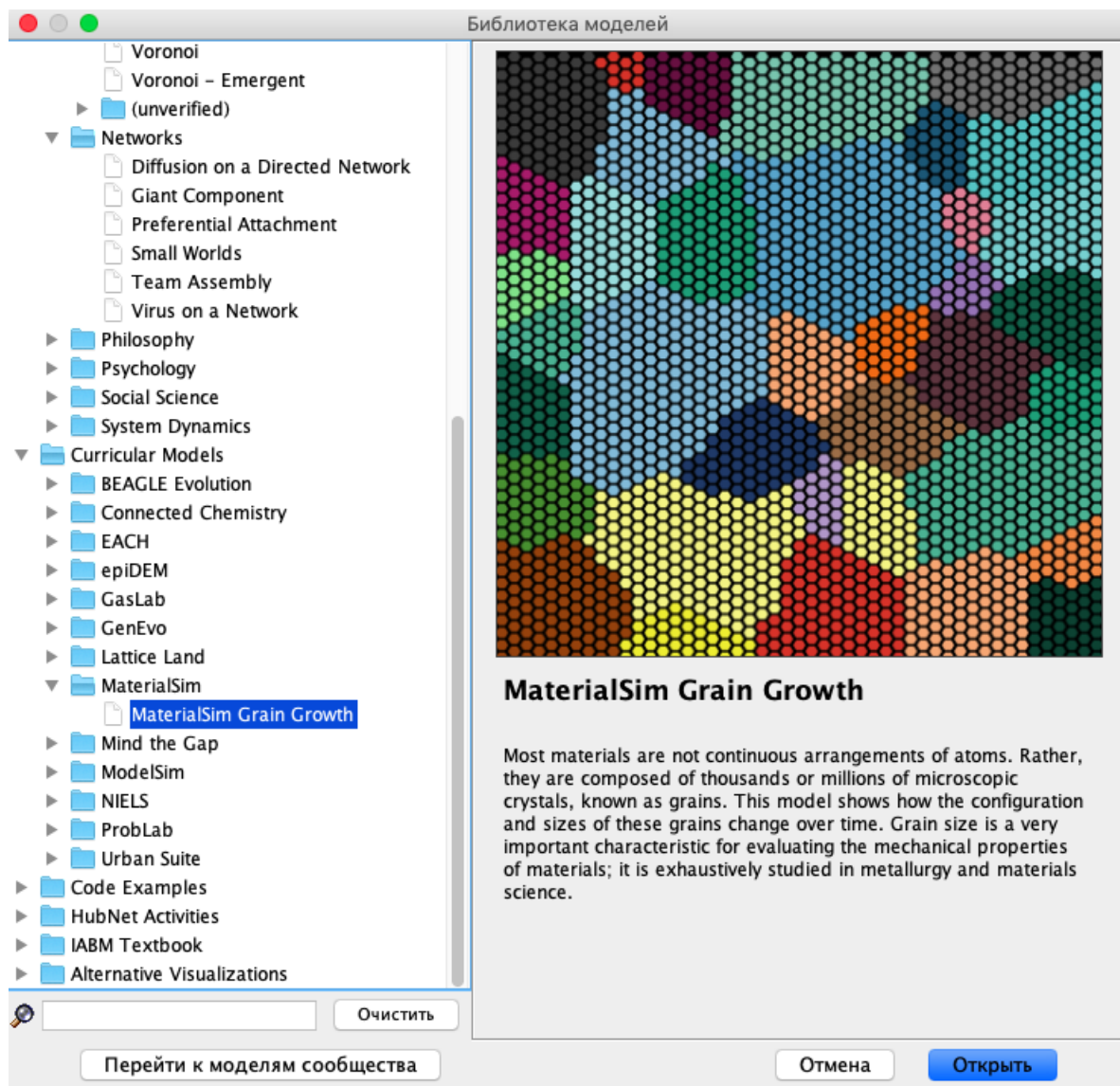


Рис. 1.4: Библиотека демонстрационных проектов

Стоит отметить, что данная библиотека имеет множество примеров использования многоагентных систем во многих научных и общественных сферах: биология, химия, искусство, психология и т.д.

После загрузки проекта в глаза бросается пользовательский интерфейс. NetLogo позволяет использовать кнопки, выпадающие списки, переключатели, рычаги и текстовые поля в качестве входных данных, а также экраны, графики, метки, диалоговые окна в качестве выходных данных.

Параметрами выбранной модели является количество популяции людей, процент заразности вируса, шанс выздороветь (в процентах) и длительность выздоровления.

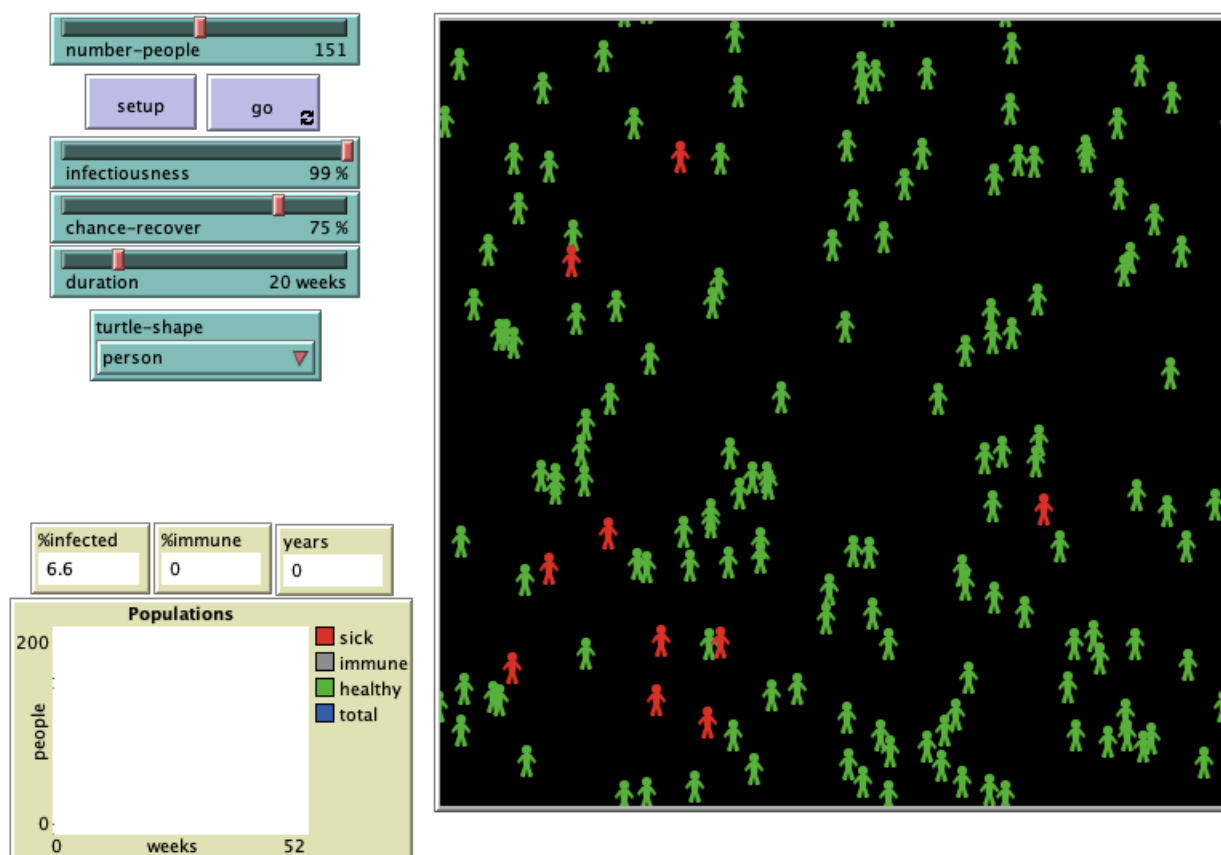


Рис. 1.5: Демонстрационный проект

Также существует режим отображения, где люди показаны кругами:

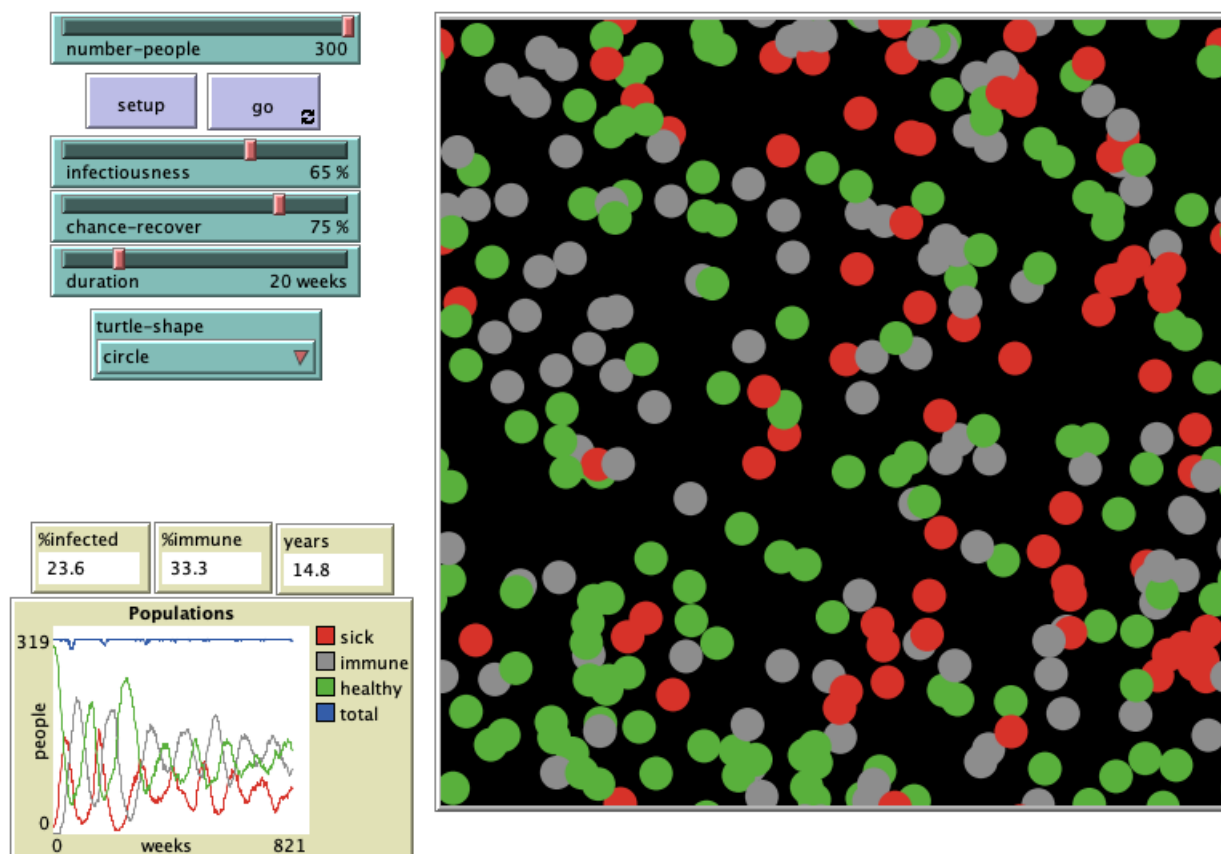
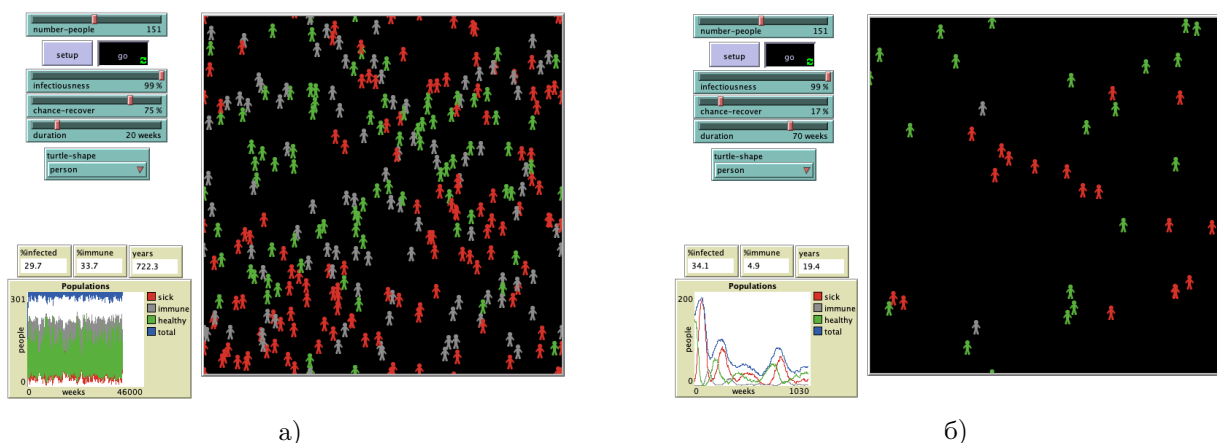


Рис. 1.6: Второй режим отображения

Данный пример промоделировал поведение вируса на протяжении около 16 лет. Как видно на диаграмме, постоянно варьируется количество заболевших, здоровых и имеющих иммунитет к болезни людей.

Далее изменяем параметры и смотрим на результаты моделирования:



На рисунке а):

- Количество людей: 151
- Процент заражения: 99 %
- Шанс выздороветь: 75 %
- Процесс выздоровления: 20 недель

Результат:

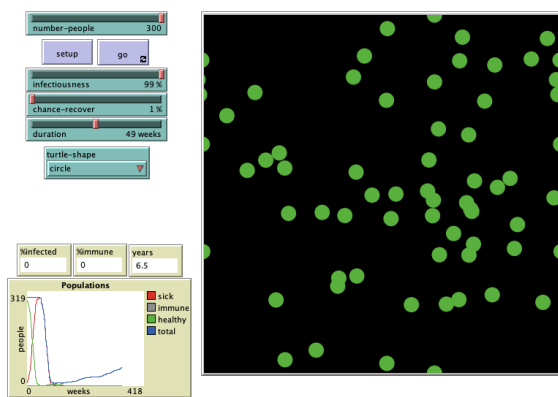
- Промоделировали систему на довольно большом отрезке времени (8822 лет).

На рисунке б):

- Количество людей: 151
- Процент заражения: 99 %
- Шанс выздороветь: 17 %
- Процесс выздоровления: 70 недель

Результат:

- Наблюдаем значительное падение популяции (практически в 2 раза).



в)

На рисунке в):

- Количество людей: 300
- Процент заражения: 99 %
- Шанс выздороветь: 1 %
- Процесс выздоровления: 49 недель

Результат:

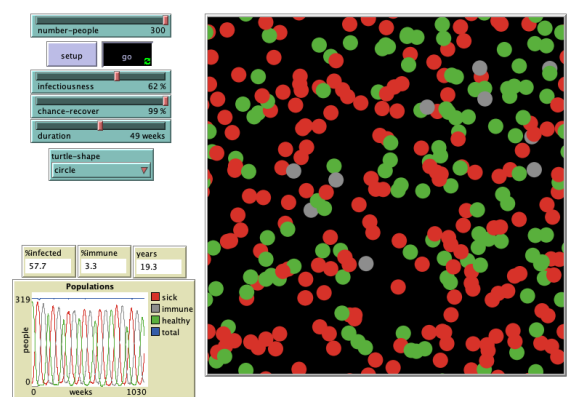
- На графике видно, что с данными параметрами, людей зараженных вирусом не осталось быстрее, чем за год, хотя в какой-то момент зараженные в системе были практически все.

На рисунке г):

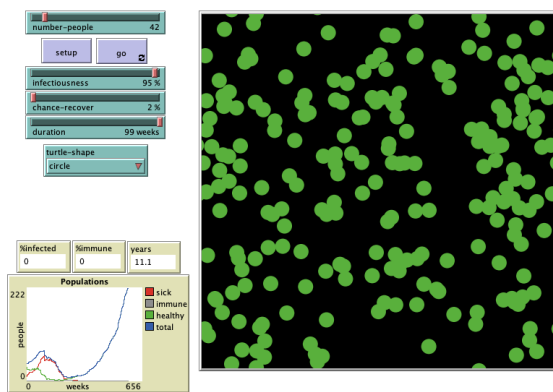
- Количество людей: 300
- Процент заражения: 62 %
- Шанс выздороветь: 99 %
- Процесс выздоровления: 49 недель

Результат:

- Наблюдаем циклические рост и падение популяции людей переносимых вирус, здоровых и имеющих иммунитет.



г)



д)

На рисунке д):

- Количество людей: 42
- Процент заражения: 95 %
- Шанс выздороветь: 2 %
- Процесс выздоровления: 99 недель

Результат:

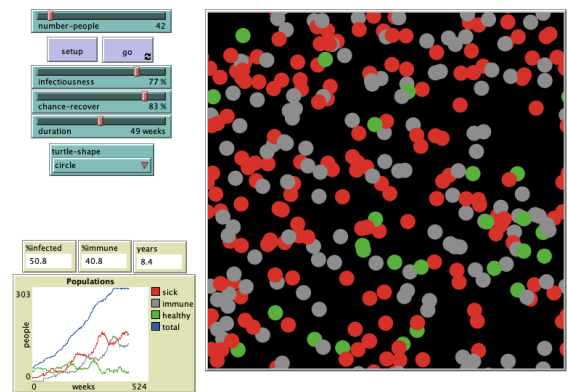
- Значительный рост популяции.

На рисунке е):

- Количество людей: 42
- Процент заражения: 77 %
- Шанс выздороветь: 83 %
- Процесс выздоровления: 49 недель

Результат:

- С ростом популяции, наблюдаем и рост зараженных людей.



е)

Анализ кода

Код пишется на одноименном языке NetLogo, который является продолжением языка Лого – первого языка, созданного еще в 1968 году объединенными усилиями Массачусетского Технологического Института и корпорации BBN (Bolt Beranek & Newman) с целью обучать детей при помощи компьютера [6]

NetLogo является процедурным, агентно-ориентированным языком программирования, который позволяет очень лаконично описывать мультиагентные системы.

Рассмотрим исходный код инициализации агентов:

```
turtles-own
[ sick?                ;; if true, the turtle is infectious
  remaining-immunity    ;; how many weeks of immunity the turtle has left
  sick-time             ;; how long, in weeks, the turtle has been infectious
  age ]                ;; how many weeks old the turtle is

globals
[ %infected            ;; what % of the population is infectious
  %immune              ;; what % of the population is immune
  lifespan             ;; the lifespan of a turtle
  chance-reproduce       ;; the probability of a turtle generating an offspring each tick
  carrying-capacity     ;; the number of turtles that can be in the world at one time
  immunity-duration ]  ;; how many weeks immunity lasts

;; The setup is divided into four procedures
to setup
  clear-all
  setup-constants
  setup-turtles
  update-global-variables
  update-display
  reset-ticks
end

;; We create a variable number of turtles of which 10 are infectious,
;; and distribute them randomly
to setup-turtles
  create-turtles number-people
  [ setxy random-xcor random-ycor
    set age random lifespan
    set sick-time 0
    set remaining-immunity 0
    set size 1.5 ;; easier to see
    get-healthy ]
  ask n-of 10 turtles
  [ get-sick ]
end

to get-sick ;; turtle procedure
  set sick? true
  set remaining-immunity 0
end

to get-healthy ;; turtle procedure
  set sick? false
  set remaining-immunity 0
  set sick-time 0
end

to become-immune ;; turtle procedure
  set sick? false
  set sick-time 0
  set remaining-immunity immunity-duration
```

```
end
```

```
;; This sets up basic constants of the model.
to setup-constants
  set lifespan 50 * 52      ;; 50 times 52 weeks = 50 years = 2600 weeks old
  set carrying-capacity 300
  set chance-reproduce 1
  set immunity-duration 52
end
```

В первую очередь описываются глобальные переменные командами `infected`, `immune`, `lifespan`, `chance-reproduce`, `carrying-capacity`, `immunity-duration`.

Процедуры, в том числе и обслуживающие элементы управления, описываются следующим синтаксисом:

```
to setup
  ...
end
```

Первой операцией процедуры `setup` является `clear-all`, которая очищает основной экран. Далее инициализируется константа и инициализируются способ отображения, количество популяции (в зависимости от режима).

После этого инициализируется заданное количество людей переносимых вирус и здоровых, со случайным показателем возраста и в случайных местах.

Рассмотрим код обработчика тактов:

```
to go
  ask turtles [
    get-older
    move
    if sick? [ recover-or-die ]
    ifelse sick? [ infect ] [ reproduce ]
  ]
  update-global-variables
  update-display
  tick
end
```

В первую очередь изменяется возраст людей (`get-older`), далее, они совершают движение (`move`) и происходит проверка на заболевание (`if sick?`), если человек болен вызывается функция, где решается дальнейшая судьба человека умрет или выздоровеет (`recover-or-die`), а если здоров проверяем заболеет ли (`infect`) или размножится (`reproduce`). В конце процедуры вызывается процедура изменения глобальных переменных (`update-global-variables`), прорисовки экрана (`update-display`) и иницируется новый такт (`tick`).

Рассмотрим код используемых процедур:

```
to update-global-variables
  if count turtles > 0
    [ set %infected (count turtles with [ sick? ] / count turtles) * 100
      set %immune (count turtles with [ immune? ] / count turtles) * 100 ]
end

to update-display
  ask turtles
    [ if shape != turtle-shape [ set shape turtle-shape ]
      set color ifelse-value sick? [ red ] [ ifelse-value immune? [ grey ] [ green ] ] ]
end

;;Turtle counting variables are advanced.
to get-older ;; turtle procedure
  ;; Turtles die of old age once their age exceeds the
  ;; lifespan (set at 50 years in this model).
  set age age + 1
```

```

    if age > lifespan [ die ]
    if immune? [ set remaining-immunity remaining-immunity - 1 ]
    if sick? [ set sick-time sick-time + 1 ]
end

;; Turtles move about at random.
to move ;; turtle procedure
    rt random 100
    lt random 100
    fd 1
end

;; If a turtle is sick, it infects other turtles on the same patch.
;; Immune turtles don't get sick.
to infect ;; turtle procedure
    ask other turtles-here with [ not sick? and not immune? ]
    [ if random-float 100 < infectiousness
      [ get-sick ] ]
end

;; Once the turtle has been sick long enough, it
;; either recovers (and becomes immune) or it dies.
to recover-or-die ;; turtle procedure
    if sick-time > duration ;; If the turtle has survived past the virus' duration
    [ ifelse random-float 100 < chance-recover ;; either recover or die
      [ become-immune ]
      [ die ] ]
end

;; If there are less turtles than the carrying-capacity
;; then turtles can reproduce.
to reproduce
    if count turtles < carrying-capacity and random-float 100 < chance-reproduce
    [ hatch 1
      [ set age 1
        lt 45 fd 1
        get-healthy ] ]
end

to-report immune?
    report remaining-immunity > 0
end

to startup
    setup-constants ;; so that carrying-capacity can be used as upper bound of number-people slider
end

```

Разбиение кода на независимые процедуры сделало его намного более понятным и простым, хотя и без этого на языке NetLogo подобные сложные системы описываются достаточно просто и интуитивно понятно.

Опишите основные возможности данного ПО применительно для создания многоагентных систем

Среда NetLogo является мощной и в то же время интуитивно понятной средой для проектирования многоагентных систем. NetLogo подходит для использования детьми (для которых и была разработана) или же серьезных ученых.

В качестве иллюстрации многогранности решаемых задач, приведем еще пару примеров.

Центральная предельная теорема Эта модель демонстрирует отношения между распределениями населения и их выборочными средними распределениями, а также влияние размера выборки на это отношение. В этой модели население распределяется некоторой переменной. Население распределяется случайным образом - не обязательно «нормально» - но образцы средств из этой популяции тем не менее накапливаются

в распределении, которое приближается к нормальной кривой. Программа позволяет проводить повторную выборку отдельных экземпляров в популяции.

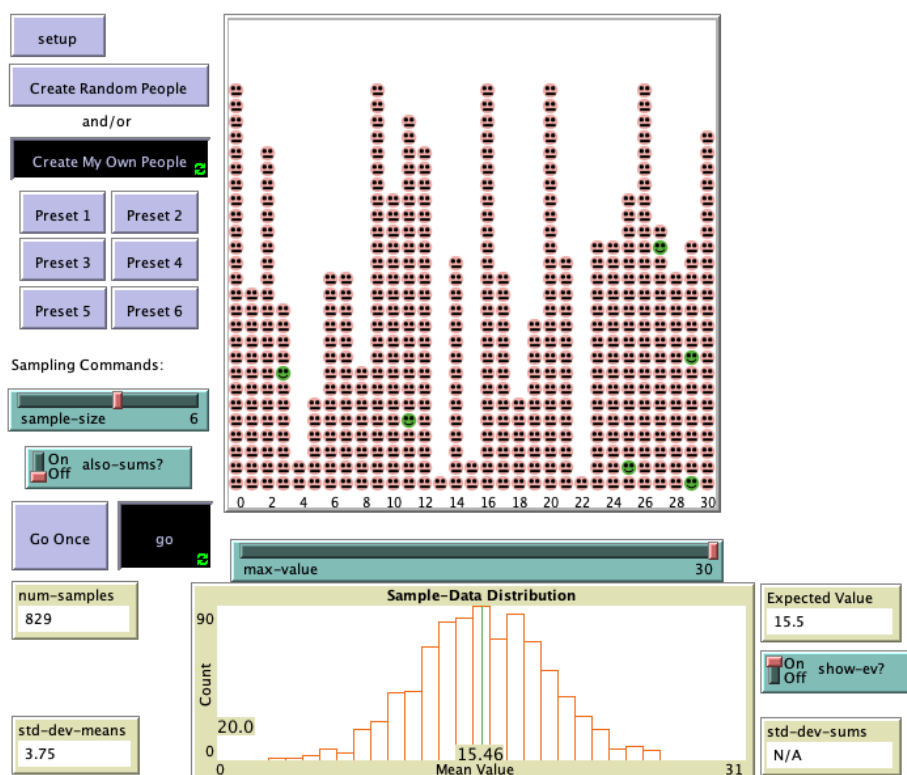


Рис. 1.7: Центральная предельная теорема

Голосование Эта модель представляет собой простой клеточный автомат, который имитирует распределение голосов путем того, что каждый патч принимает «голос» своих восьми соседних соседей, а затем, возможно, изменит свое голосование в соответствии с результатом.

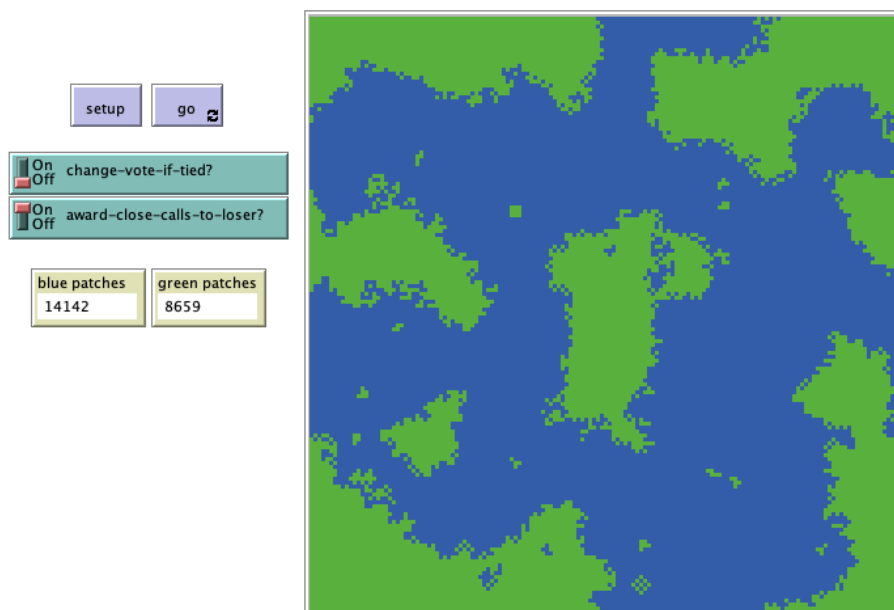


Рис. 1.8: Voting

Слухи Эта программа моделирует распространение слуха. Слух распространяется, когда человек, который знает этот слух, рассказывает одному из своих соседей. Другими словами, пространственная близость является определяющим фактором того, как скоро (и, возможно, как часто) данный человек услышит слух.

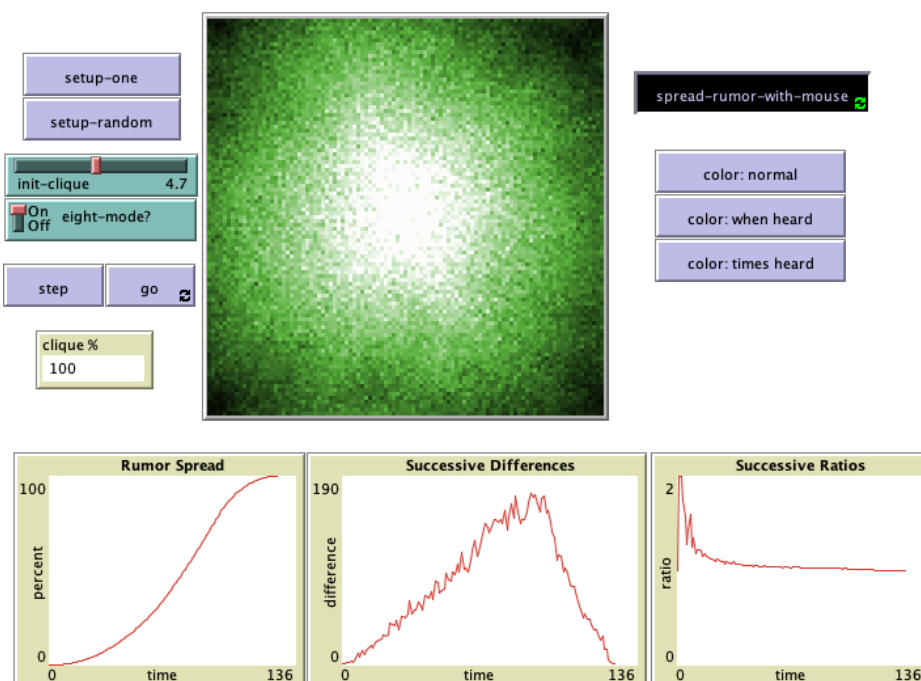


Рис. 1.9: Rumor Mill

Маленький мир Эта модель исследует формирование сетей, которые приводят к феномену «малого мира» - идея о том, что человек - всего лишь пара соединений с любым другим человеком в мире.

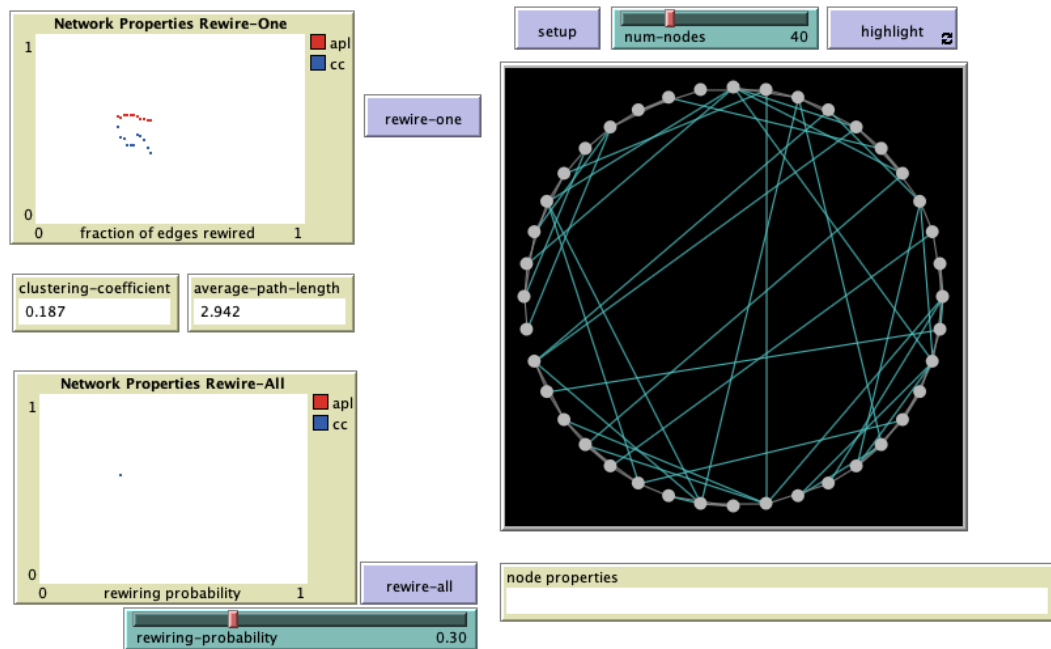


Рис. 1.10: Small Worlds

Пакман Эта игра представляет всемирно известную игру "Пакман".

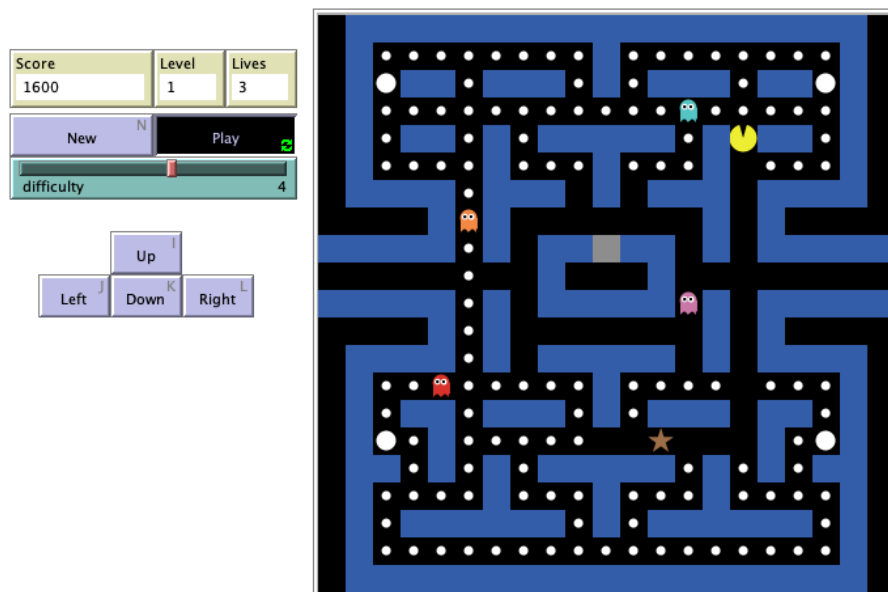


Рис. 1.11: Пакман

Таким образом NetLogo может быть использована для самого широкого спектра многоагентных задач, от демонстрации работы генетического алгоритма до компьютерных игр.

1.4 Вывод

В ходе работы были изучены принципы работы многоагентных систем. Был произведен анализ преимуществ и недостатков данного подхода, рассмотрена классификация, а также изучена среда для разработки многоагентных систем NetLogo.

В чем Плюсы и минусы многоагентного подхода?

Основные преимущества использования многоагентных систем, это: **распределенные вычисления, масштабирование и автономность**. Кроме того агенты могут общаться между собой, обмениваться информацией и кооперироваться для выполнения задач.

Возможно, недостатком можно посчитать то, что с увеличением параметров такой системы снижается ее легкая отслеживаемость.

Какие еще среды или языки программирования использует для создания многоагентных систем?

Для разработки многоагентных системы наилучшим образом подойдут следующие программные средства: Python, Matlab, Logo, NetLogo.

Как по-вашему стоит ли развивать данное направление, если нет, то почему, если да, то в какую сторону?

Определенно стоит! Это интересно и полезно. Они прекрасно подходят для выявления макрозакономерностей в таких важнейших науках, как биология, физика, химия, социология и др. Поэтому, я считаю, нужно продолжать двигаться в этом направлении и постепенно увеличивать наши знания о мире во всех отраслях.

Корректно ли по-вашему моделирование многоагентных систем на одной вычислительной машине?

Как и для любого программного средства, ответ на этот вопрос зависит от сложности задачи. Для простой многоагентной системы вполне достаточно и одного компьютера, в чем мы собственно и убедились при описании программы NetLogo.

Приведите области в которых применение многоагентного подхода дает максимально положительные результаты.

- Выявление макрозакономерностей в естественных науках и социальных исследованиях.
- Исследование нагруженности сетевых и мобильных сервисов.
- Исследование предпочтений пользователей.
- Доказательство математических теорем.
- Обучение детей.

1.5 Список литературы

- [1] Многоагентные системы AI Portal [Электронный ресурс]. — URL: <http://www.aiportal.ru/articles/multiagent-systems/multiagent-systems.html> (дата обращения 18.11.2018).
- [2] ИНТЕЛЛЕКТУАЛЬНЫЕ МИКРОРОБОТОТЕХНИЧЕСКИЕ АГЕНТЫ И МНОГОАГЕНТНЫЕ МИКРОСИСТЕМЫ [Электронный ресурс]. — URL: <http://www.microsystems.ru/files/publ/205.htm> (дата обращения 18.11.2018).
- [3] Агенты, многоагентные системы, виртуальные [Электронный ресурс]. — URL: <http://pandia.ru/text/78/362/252-2.php> (дата обращения 18.11.2018).
- [4] Мультиагентные системы [Электронный ресурс]. — URL: <http://mabi.vspu.ru/portfolio/multiagentnyie-sistemyi/> (дата обращения 18.11.2018).
- [5] Классификация агентов [Электронный ресурс]. — URL: <http://www.aiportal.ru/articles/multiagent-systems/agent-classification.html> (дата обращения 18.11.2018).
- [6] NetLogo Letopisi [Электронный ресурс]. — URL: <http://letopisi.org/index.php/NetLogo> (дата обращения 18.11.2018).
- [7] Download NetLogo [Электронный ресурс]. — URL: <https://ccl.northwestern.edu/netlogo/download.shtml> (дата обращения 18.11.2018).