

ПРОТОТИП ПЛАТФОРМЫ **AUTO2KILL** ДЛЯ АВТОМАТНОГО МОДЕЛИРОВАНИЯ И СОРЕВНОВАНИЙ VERSION **0.95** БЕТА, BUILD **3091**

*Леонид Столяров (9 класс) <Ind1212@rambler.ru>,
Федор Петряйкин (9 класс), Никита Уваров (7 класс)*

Научный руководитель: И.Р. Дединский (МФТИ) <mail@ded32.net.ru>

Заказчики и консультанты: проф. А. А. Шалыто, проф. В.Г. Парфенов (СПбГУ ИТМО)

КРАТКОЕ РУКОВОДСТВО

Пример сеанса работы с системой и разработки интеллекта для «Умного муравья»

1. Введение

Представим, что мы участвуем в соревнованиях по «автоматным боям» на платформе *Auto2Kill*. Согласно парадигме соревнований [1], управлять персонажем непосредственно участники не могут. Единственный способ победить – разработать интеллект, который справится с задачей лучше интеллектов других участников.

2. Предварительная проверка системы и установка

Для запуска платформы компьютер должен удовлетворять минимальным системным требованиям и на нём должно быть установлено необходимое программное обеспечение:

1. *Microsoft Visual C++ Runtime x86 2008 SP1 (файл vcredist_x86).*
2. *PhysX Runtime 9.09.0814 (файл PhysX_9.09.0814_systemSoftware).*
3. *Microsoft .NET 3.5 (файл dotNetFx35setup).*
4. *Microsoft Visio 2003.*

Дистрибутивы этих компонентов (кроме *MS Visio*) находятся в папке **Required**. Для проверки наличия этих компонентов на компьютере используется утилита **TestRequirements**, находящаяся в папке **Required/TestRequirements**. Если утилита завершается без каких-

либо сообщений, то все компоненты найдены, иначе появится сообщение об ошибке. В этом случае необходимо установить или переустановить необходимые компоненты.

Саму систему *Auto2Kill* специально устанавливать не нужно (достаточно просто скопировать в папку с разрешением по записи). В настоящее время не гарантируется работа из сетевой папки с путем UNC, а также с правами пользователя, отличными от прав администратора.

3. Запуск системы в несетевом режиме

Для запуска платформы необходимо запустить файл **Auto2Kill11** из папки **Bin** (см. рис. 1).

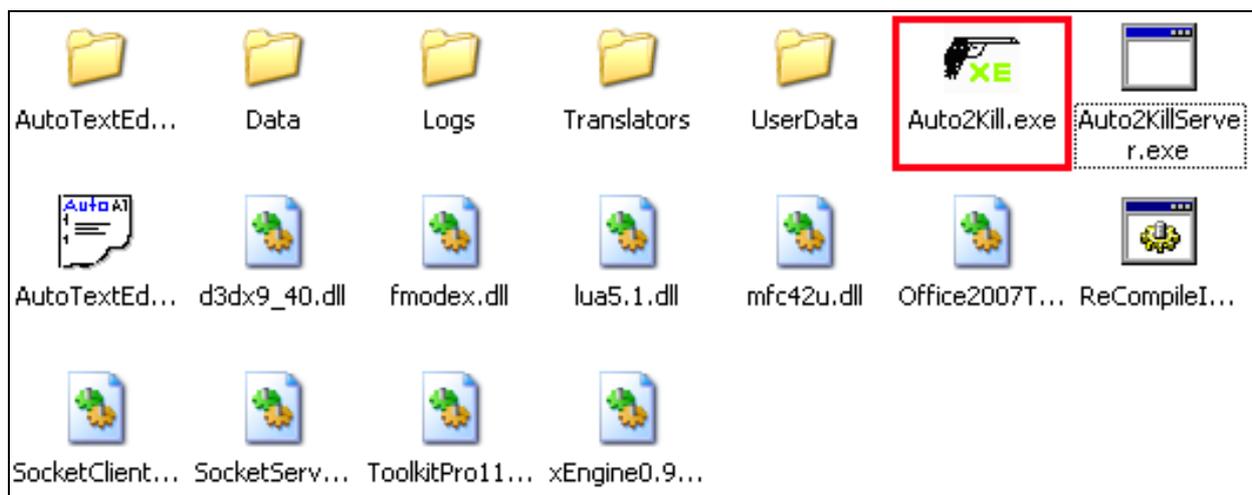


Рис. 1. Директория Bin.

Если интеллекты были изменены перед запуском платформы, то до запуска платформы необходимо запустить файл **ReCompileIntelligencies**, что приведёт к перекомпиляции интеллектов.

После запуска платформы появится главное меню. В нём присутствуют кнопки **[Start Task]** и **[Net mode]**, которые включают различные режимы работы платформы: индивидуальный или сетевой соответственно. Кнопка **[Exit]** предназначена для выхода из программы (см. рис. 2).



Рис. 2. Главное меню программы.

При выборе индивидуального режима будет предложено выбрать карту (см. рис. 3) с помощью меню карт. На данном этапе разработки доступно две карты, равнозначные с точки зрения соревнований интеллектов, однако карта **Prototype_Light** содержит меньше графического оформления, чем **Prototype**, и предназначена для компьютеров с низкой (или встроенной) видеокартой. Когда карта выбрана, ее можно запустить, нажав кнопку **[Start level]**. Во время загрузки карты на экране отображается окно загрузки с надписью «**Loading level...**». Когда карта будет загружена, можно начинать соревнования (см. рис. 4).

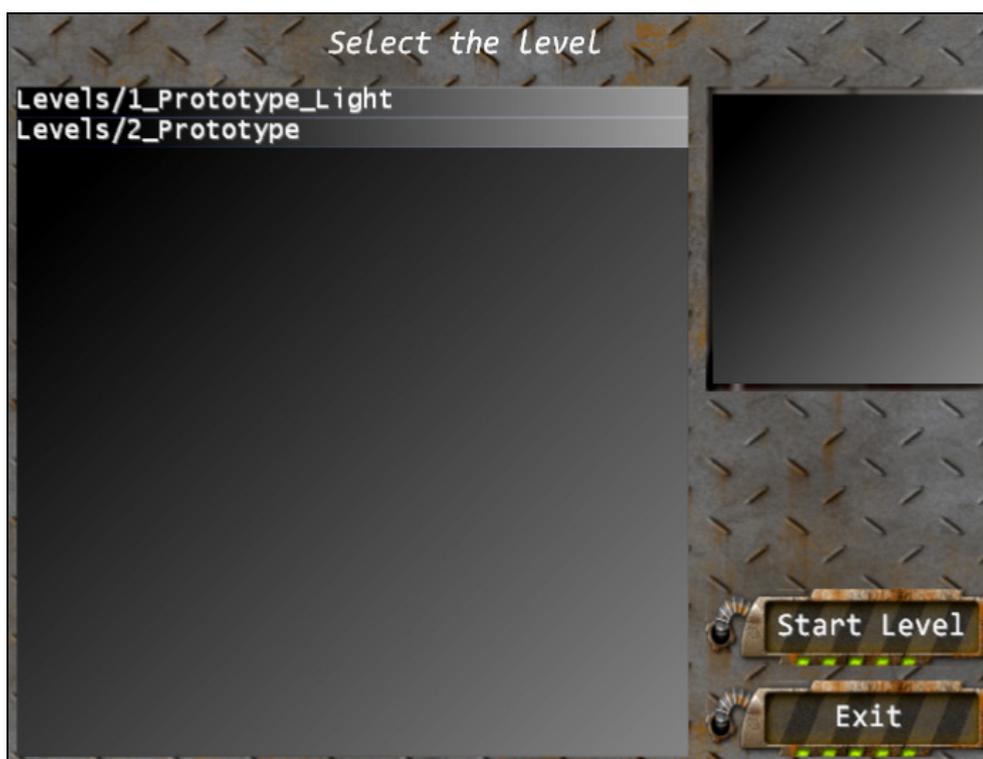


Рис. 3. Меню выбора карты.

4. Работа с системой

В правом верхнем углу экрана отображается панель управления муравьями. Цифры вверху этой панели предназначены для выделения муравья. Кнопки, расположенные ниже, совершают различные действия над выделенным муравьем. Для запуска муравья необходимо нажать кнопку **[Start]**. Кнопка **[Edit]** открывает интеллект муравья в соответствующем редакторе. Если интеллект был изменен, то необходимо обновить его, нажав на кнопку **[Update]**. Добавить муравьев можно нажатием кнопки **[Add ant]** на боковой панели (см. рис. 4). В полях **Apples eaten** и **Current step** отображается количество съеденных муравьем яблок и количество пройденных ходов. Над кнопками управления располагается меню вы-

бора муравья: при нажатии на номер муравья он выделяется. На данный момент поддерживается до четырех независимых муравьев.

Передвижение по игровому миру осуществляется с помощью клавиш [W] (вперед), [S] (назад), [A] (влево), [D] (вправо) и поворота мыши с зажатой левой клавишей (поворот камеры).



Рис. 4. Игровое поле.

5. Разработка интеллекта муравья

У муравья есть одно входное воздействие – **a**, обозначающее есть ли перед ним яблоко (*apple*). Три выходных воздействия **g**, **tl** и **tr** служат для перехода вперёд на одну клетку (*go*), поворота налево (*turn left*) и поворота направо (*turn right*) соответственно.

Рассмотрим базовый (изначальный) интеллект муравья, входящий в комплект системы (рис. 5).

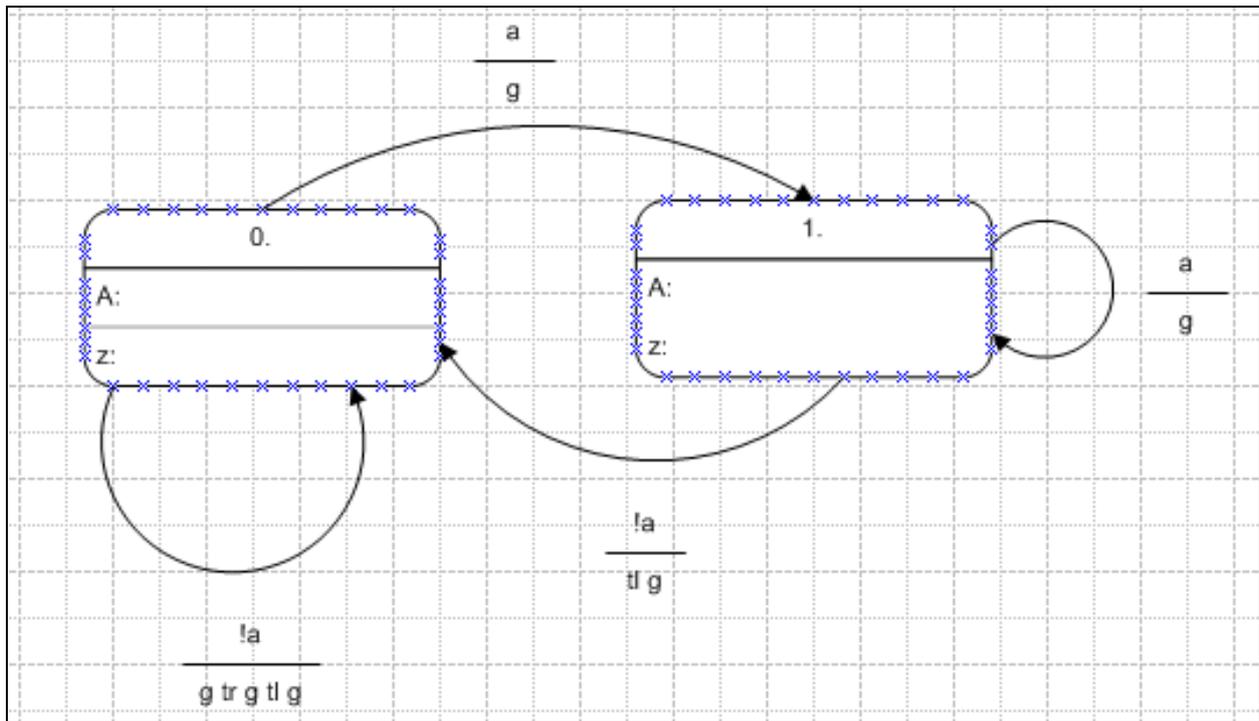


Рис. 5. Базовая версия интеллекта умного муравья.

Этот интеллект в среднем съедает 13 яблок за 200 ходов. Пусть мы хотим изменить этот интеллект. Для этого выделим нужного муравья и нажмём кнопку **[Edit]** на боковой панели. Для создания новой фигуры на схеме необходимо перетащить её с шаблона на лист. Изменённый интеллект приведён на рис. 6.

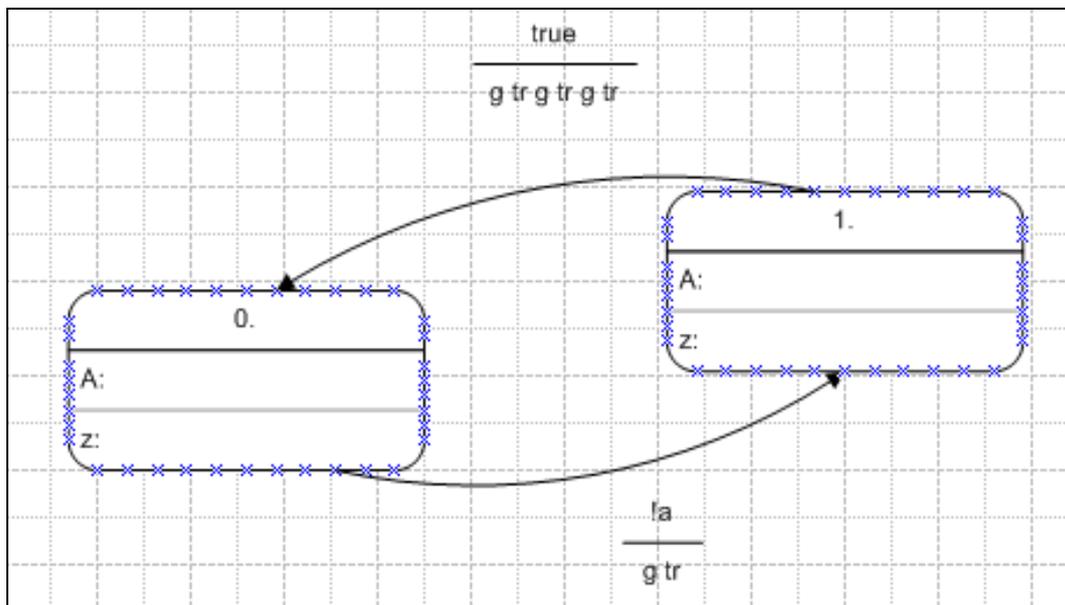


Рис. 6. Вторая версия интеллекта умного муравья.

После запуска этого интеллекта видно, что с ним возможно съесть максимально 3 яблока, т.к. муравей будет сразу останавливаться. Рассмотрим, почему это происходит:

- 1 случай: пусть изначально перед муравьём есть яблоко. В этом случае муравей не сдвинется с начальной клетки, т. к. он находится в состоянии **0**, а из этого состояния нет переходов под условием **a**.
- 2 случай: пусть изначально перед муравьём нет яблока. Тогда муравей пройдёт на следующую клетку, повернётся направо и перейдёт в состояние **1**. После этого он в любом случае вернётся в начальное состояние (это обеспечивает команда перехода из **1** в **0**). В клетке перед ним нет яблока (по предположению). Значит, муравей ходит по кругу, в котором яблоки могут лежать только в трёх клетках.

Получилось, что интеллект мы ухудшили. Изменим интеллект ещё раз (рис. 7).

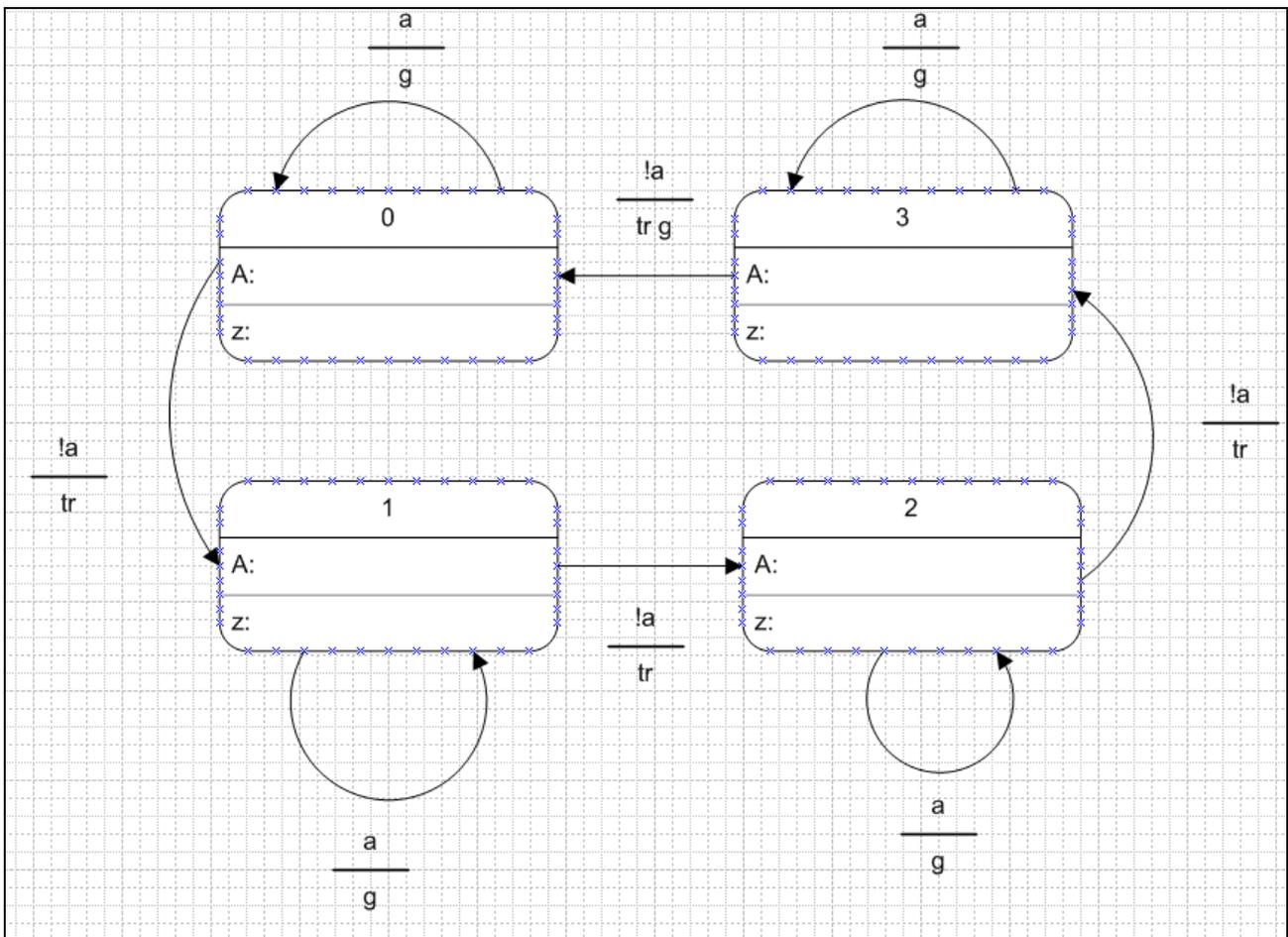


Рис. 7. Третья версия интеллекта умного муравья.

Этот интеллект лучше первых двух версий. Он позволяет съесть 20 яблок, т. к. в нем учитывается то, что нужно осматривать не только клетку, а еще и смежные с ней.

6. Просмотр результатов

Предположим, что мы запустили нескольких муравьёв и хотим сравнить их результаты. Для этого нужно нажать на клавишу **[Tab]**. В отображенной таблице будут показаны свод-

ные сведения о каждом муравье: количество шагов и съеденных на текущий момент яблок (см. рис. 8).



ANT	APPLES	STEPS	STATE
ant#1	5	81	
ant#2	4	95	
ant#3	3	73	
ant#4	3	68	

Рис. 8. Сводная таблица текущих результатов.

7. Запуск и работа с системой в сетевом режиме

Теперь предположим, что мы хотим провести соревнования по сети. Тогда на одном из компьютеров нужно запустить программу сервера **Auto2KillServer** из папки **Bin** (см. рис. 1). У каждого участника в файле **Bin/UserData/Settings/Auto2KillServer.xcfg** в параметре **ip** должен быть указан IP-адрес компьютера с запущенным сервером, доступный с компьютеров участников соревнований.

Затем каждый участник запускает на своём компьютере файл **Auto2Kill** из папки **Bin**, как и для несетевого варианта. Однако в главном меню (см. рис. 2) теперь нужно выбрать **[Net mode]**. В этом режиме окно выбора карты отображено не будет, а на общее поле будет автоматически добавлен новый муравей. В данной версии в сетевом режиме количество ходов не ограничено, а яблоки появляются в течение всего времени соревнования. Участник может управлять только своим муравьем и ему нельзя добавлять новых муравьев. Он также не может выделять других муравьев и в его меню выбора муравьев доступен всегда только один «его собственный» муравей.

Условия победы в сетевом соревновании определяются регламентом состязаний.

8. Литература

1. Столяров Л., Петряйкин Ф., Уваров Н. Платформа для автоматного моделирования и соревнований. I. Разработка движка для автоматного моделирования. Тезисы Балтийского научно-инженерного конкурса. СПб, 2010.