

**МОДУЛЬ ТРЁХМЕРНОЙ ВИЗУАЛИЗАЦИИ ДЛЯ СИМ TRIAD.NET****К. В. Рябинин (Пермь)**

Визуализация результатов прогона модели является очень важной задачей при реализации современных систем имитационного моделирования. Наличие компонентов визуализации значительно увеличивают привлекательность системы для пользователя. Кроме того, визуализация процесса и результата моделирования упрощают такие этапы, как верификация и отладка модели, так как человеку проще оценить адекватность моделирования, опираясь на визуальный образ, а не на текстовые и числовые данные.

Визуальный образ, построенный на основе прогона имитационной модели, также является презентабельным, то есть пригоден для демонстрации другим людям, даже не являющимся специалистами в области имитационного моделирования. Это может быть полезно в том случае, если есть необходимость представить результаты моделирования людям, заинтересованным в изучении поведения моделируемой системы.

В работе [1] анимация (то есть визуализацию с отображением движений) выделяется как один из критериев сравнения систем имитационного моделирования. При этом отмечается, что модуль, отвечающий за анимацию, может быть как частью системы, так и внешним. В качестве критерия качества анимации отмечается не только её визуальная привлекательность и соответствие моделируемой предметной области, но и её интерактивность, то есть наличие у пользователя возможностей управления процессом визуализации. К таким возможностям относится, например, управление скоростью воспроизведения анимации.

**Система имитационного моделирования Triad.Net**

На кафедре математического обеспечения вычислительных систем Пермского государственного национального исследовательского университета была разработана система имитационного моделирования Triad [2]. Современной версией этой системы является Triad.Net [3], созданная на базе платформы Microsoft .Net. В состав Triad.Net входит графический редактор моделей, формирующий описания в виде XML-файлов.

**Модуль графического расширения Snail Engine**

Для визуализации процесса моделирования в системе Triad.Net было разработано приложение на базе модуля графического расширения Snail Engine [4, 5, 6].

Снимок экрана итоговой программы приведён на рис. 1.

Snail Engine является кроссплатформенным объектно-ориентированным модулем графического расширения, разработанным как надстройка над библиотекой стандарта OpenGL. Он предоставляет возможность работы с двумерной и трёхмерной сценами на уровне объектов, а не цепочек многоугольников, как это делается в OpenGL.

Иерархия классов Snail Engine описывает большое количество графических сущностей (например, камеру, трёхмерные модели и текстуры), а также инструменты управления этими сущностями: загрузки данных из файлов, организации очереди рендеринга и т. д. Модуль Snail Engine написан на языке C++ и подключается к прикладным программам на уровне исходного кода. Поэтому сами прикладные программы также должны разрабатываться с использованием C++.

В качестве примера для визуализации была взята классическая задача имитационного моделирования систем массового обслуживания – задача об автозаправочной станции.



Рис. 1. Визуализация имитационной модели автозаправочной станции

### Архитектурные и технические особенности модуля визуализации

Разработанная на базе Snail Engine программа визуализации представляет собой отдельный и полностью независимый модуль, коммуникация которого с системой Triad.Net осуществляется через XML-файл. Решение использовать такой подход обусловлено сразу несколькими причинами. Во-первых, система Triad.Net написана на С# с использованием платформы .Net, поэтому интеграция с ней программы, реализованной на С++, затруднительна и нецелесообразна. Во-вторых, Triad.Net предоставляет возможность удалённого прогона модели, то есть поддерживает клиент-серверную архитектуру. В этом случае визуализатор должен находиться на клиентской машине для обеспечения хорошей производительности при отображении сцены в высоком разрешении, а значит, должен существовать отдельно от основной системы. В-третьих, Triad.Net является универсальной системой моделирования, то есть не привязан к терминам какой-либо конкретной предметной области, тогда как визуализатор создаётся под конкретную задачу. Используемая в нём сцена (которая по большей части описывается на уровне исходного кода) наилучшим образом подходит только для демонстрации моделирования в выбранной предметной области (автозаправочной станции). В случае изменения предметной области следует также внести коррективы в исходный код визуализатора (хотя большая часть кода может быть переиспользована) и изменить набор мультимедийных ресурсов (трёхмерных моделей и текстур, хранящихся во внешних файлах).

Кроме того, созданный визуализатор является кроссплатформенным, тогда как Triad.Net, в силу того, что он основан на платформе .Net, может быть запущен только под управлением ОС Windows. Таким образом, пользователь, имея уже сгенерированные XML-файлы, может просмотреть весь процесс моделирования в визуализаторе, без необходимости установки и запуска системы Triad.Net.

Для коммуникации системы Triad.Net и визуализатора был разработан специальный протокол общения на базе XML. В рамках системы Triad.Net он был реализован аспирантом кафедры математического обеспечения вычислительных систем Артёмом Шафрановым. Код, реализующий протокол, был помещён в информационные процеду-

ры системы, которые служат для сбора данных о процессе и результатах моделирования.

Файлы, получаемые от системы Triad, представляют собой модифицированные (изменённые в рамках разработанного протокола общения) журналы событий и имеют вид последовательных шагов, сгруппированных по признаку времени. Так как Triad является событийно-ориентированной системой моделирования, модельное время фиксируется только тогда, когда происходит изменение состояния системы: приход очередной заявки, помещение заявки в очередь и т. д. Промежутки времени, когда система неизменна, проходят мгновенно. Таким образом, изменение времени нелинейно.

Нелинейность времени реализована и в визуализаторе: каждое следующее событие происходит сразу после предыдущего, даже если между ними находится значительный промежуток модельного времени. При этом изменение счётчика модельного времени имеет скачкообразный характер.

В процессе решения задачи визуализации возникает так называемая проблема нулевого времени: ситуация, когда действие, которое в оригинале осуществлялось мгновенно, требует при визуализации определённых временных затрат. Это происходит, например, если необходимо отобразить перемещение заявки от одного пункта системы массового обслуживания к другому. При моделировании средствами Triad.Net считается, что такое перемещение происходит мгновенно, но для повышения привлекательности визуализации следует воспроизводить соответствующую анимацию движения. Следовательно, необходимо затратить время, не предусмотренное сценарием, получаемым от системы Triad.Net.

Чтобы избежать временных накладок, вся анимация воспроизводится вне модельного времени, т. е. следующее событие считывается из журнала только тогда, когда полностью отрисовано предыдущее. Такой подход является наиболее простым решением проблемы нулевого времени и гарантирует корректное отображение любой последовательности событий.

Ещё одним важным преимуществом, которое даёт такое решение проблемы нулевого времени, является лёгкость управления скоростью визуализации. Достаточно только изменить скорость воспроизведения соответствующих анимаций, и весь процесс отображения процесса моделирования ускорится, либо замедлится. В разработанной программе присутствует возможность динамического изменения скорости, что позволяет проматывать «скучные» места в процессе моделирования быстрее и замедлять воспроизведение при возникновении интересных ситуаций, либо же достаточно быстро демонстрировать модели с большим числом событий.

Созданная программа, как и любое приложение, основанное на модуле графического расширения Snail Engine, поддерживает стереоскопическую визуализацию, пригодную для просмотра в шлеме виртуальной реальности eMagin Z800 или затворных очках eDimensional 3D Vision.

Управление камерой на трёхмерной сцене осуществляется введением так называемого 3D-курсора, т. е. объекта, к которому привязана камера как к цели слежения. Положением этого курсора а также поворотом камеры вокруг него можно управлять при помощи мыши и клавиатуры. В программе реализованы подсказки, которые можно включить или отключить. Из этих подсказок пользователь может получить всю необходимую информацию об управлении.

Трёхмерная сцена представлена ландшафтом, небом и набором трёхмерных моделей, которые отвечают за изображение мелких статических деталей сцены и самих заявок, роль которых в выбранной предметной области играют автомобили. Все эти объекты реализованы при помощи средств, предоставляемых модулем Snail Engine.

Так, например, ландшафт строится по карте высот, которая загружается из файла. Эта карта представляет собой двумерное изображение, кодирующее высоты ландшафта яркостью пикселей. Затем на построенную по карте высот поверхность ландшафта накладывается текстура, также загружаемая из файла. Небо представляет собой куб, описанный вокруг сцены, на грани которого наложены соответствующие изображения. Трёхмерные модели загружаются из файлов формата 3DS, а затем на них также накладываются текстуры.

Помимо трёхмерных объектов использованы и двумерные, отвечающие за отображение различных индикаторов (например – индикатор высоты камеры над ландшафтом) и текста (для отображения модельного времени и справочной информации). Эти объекты также созданы при помощи средств, предоставляемых модулем Snail Engine.

Визуализатор представляет собой группу классов, наиболее важными из которых являются менеджер, обслуживающее устройство и клиент. Внутри менеджера собраны все методы работы со сценарием, получаемым от системы Triad.Net и управления поведением клиентов в соответствии с этим сценарием.

Визуализатор поддерживает работу с многоканальными устройствами. При этом число каналов определяется из мета-информации, находящейся непосредственно перед описанием процесса моделирования в XML-файле сценария. Класс обслуживающего устройства введён для хранения всей необходимой информации о многоканальном устройстве а также для управления его отображением.

Класс клиента отвечает за отображение заявки. При создании экземпляра этого класса случайным образом выбирается и загружается одна из доступных моделей автомобилей, что вносит разнообразие в визуализацию. Также этот класс реализует методы анимации движения заявки и организации очереди.

Анимация действий заявок сводится к их перемещению по сцене. Нетривиальным вопросом является определение траектории движения заявки. Получив уведомление от менеджера о том, какое действие следует произвести, заявка формирует массив контрольных точек, через которые проходит траектория движения. Первой точкой в этом массиве является текущее положение заявки, а последней – положение места назначения. Местом назначения может быть либо пункт удаления заявок со сцены, координаты которого известны (заданы в коде программы), либо точка конца очереди, координаты которой также известны заранее, либо канал устройства, координаты которого сообщает само устройство. Промежуточные точки выбираются в соответствии с конфигурацией ландшафта.

Через контрольные точки проводится ломаная линия, углы которой сглаживаются при помощи кривых Безье [6]. Полученная кривая и используется в качестве траектории плавного движения автомобиля.

## **Выводы**

Разработанный программный продукт может быть использован для демонстрации возможностей системы Triad.Net и модуля графического расширения Snail Engine. Этот продукт является единственным на данный момент средством трёхмерной визуализации процесса имитационного моделирования в системе Triad.Net. Достоинствами по сравнению с аналогами для других систем имитационного моделирования являются свобода и межплатформенная переносимость созданного приложения, а также поддержка стереоскопической визуализации, совместимой со шлемом виртуальной реальности eMagin Z800 и затворными очками eDimensional 3D Vision. Недостатком же на настоящий момент является отсутствие каких-либо редакторов, которые позволили бы создавать сцены для визуализации задач в других предметных областях без про-

граммирования. Создание таких редакторов следует выделить в качестве направления дальнейшего развития средств визуализации для системы Triad.Net.

### Литература

1. **Jalal Nikoukaran, Vlatka Hlupic, Ray J. Paul.** Criteria For Simulation Software Evaluation. In Proceedings of the 1998 Winter Simulation Conference D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds. P. 399–406.
2. **Mikov A. I.** Simulation and Design of Hardware and Software with Triad// Proc.2nd Intl.Conf. on Electronic Hardware Description Languages, Las Vegas, USA, 1995. P. 15–20.
3. **Миков А. И., Замятина Е. Б.** Проблемы повышения эффективности и гибкости систем имитационного моделирования // Проблемы информатики/ Новосибирск, Институт вычислительной математики и математической геофизики СО РАН, 2010. № 4(8). С. 49–64.
4. **Рябинин К. В., Полотнянщиков И. С.** SNAIL ENGINE: кроссплатформенный модуль графического расширения. URL: <http://rae.ru/forum2011/104/293> (дата обращения: 13.06.2011).
5. **Рябинин К. В.** Разработка кроссплатформенного объектно-ориентированного модуля графического расширения на базе спецификации OpenGL // Современные проблемы математики и её прикладные аспекты. Перм. гос. ун-т., 2010. С. 55.
6. **Рябинин К. В.** Проект Snail Engine: кроссплатформенный объектно-ориентированный модуль графического расширения на базе стандарта OpenGL. Актуальные проблемы механики, математики, информатики. Перм. гос. ун-т., 2010. С. 189.