

# Interactive 3D scene modeling and visualization system

Roman Grebinnik  
Software Engineer, Infostroy Ltd.  
email: [r.grebinnik@infostroy.com.ua](mailto:r.grebinnik@infostroy.com.ua)

Alexander Lipanov  
PhD, Director of Infostroy Ltd., associate  
professor, chair of Informatics, KNURE  
email: [alex@infostroy.com.ua](mailto:alex@infostroy.com.ua)

## Abstract

*In this scientific report the visual system for 3D scene creation for the purpose of design and visualization of various objects' behavior scenarios on the stage, their interaction with each other and also with the participation of user is described. The following system can be used at the development of virtual simulator systems and at visual modeling of different situations to demonstrate the interaction of objects with each other on the stage. The system provides for the ability to interact with user during scene demonstration, and objects' behavior on stage is being managed according to the scenario, which is predesigned by user with the help of a simple script language. These system capabilities allow substantially reduce the time required for the development of some model of an interactive 3D scene; provide possibility for user to participate in actions on scene. The system can be used for training purposes, for analysis and demonstration of various situations, where processes visualization is necessary considering mathematical models and physical properties of objects. The system is developed using the up to date technologies .NET Framework 3.5 and XNA Game Studio 3.1.*

**Keywords:** 3D scenes modeling, interactive scene, XNA Studio, simulation system, 3D scene editor.

## Система моделирования и визуализации интерактивных 3D сцен

Роман Гребинник  
Инженер-программист,  
ООО «Инфострой»  
email: [r.grebinnik@infostroy.com.ua](mailto:r.grebinnik@infostroy.com.ua)

Александр Липанов  
к.т.н., директор ООО «Инфострой»,  
доцент кафедры Информатики, ХНУРЭ  
email: [alex@infostroy.com.ua](mailto:alex@infostroy.com.ua)

## Реферат

В докладе рассматривается система создания 3D сцен, предназначенная для моделирования и визуализации различных сценариев поведения объектов на сцене, их взаимодействия между собой, и с участием пользователя. Современные технологии предоставляют обширные возможности в области компьютерной графики, в тоже время зачастую разработка различных систем с использованием компьютерной графики является долгим и сложным процессом. Полнофункциональных систем, в которых сцена будет моделироваться на основе сценария описанного пользователем с возможностью его участия в действиях на сцене и с возможностью задания специальных моделей поведения объектов на сегодняшний день не существует. Данная система может быть применена при разработке систем виртуальных тренажеров, визуальном моделировании различных ситуаций для демонстрации взаимодействия объектов на сцене между собой. В системе предусматриваются возможности взаимодействия с пользователем во время демонстрации сцены, а поведением объектов на сцене управляет сценарий, который предварительно разрабатывает пользователь при помощи простого скриптового языка. Данные возможности системы позволяют существенно сократить время необходимое на разработку некоторой модели интерактивной 3D сцены, предоставить возможности участия пользователя в действиях на сцене. Систему можно использовать при обучении, при анализе и демонстрации различных ситуаций, где необходима визуализация процессов, с учетом математических моделей и физических свойств объектов. Система разрабатывается с использованием новейших технологий .NET Framework 3.5 и XNA Game Studio 3.1.

**Ключевые слова:** моделирование трехмерных сцен, интерактивная сцена, XNA Studio, система моделирования, редактор трехмерных сцен.

## 1. Введение. Современные средства построения интерактивных 3D сцен

Современный уровень развития вычислительной техники и программного обеспечения открывает широчайшие возможности в разработке различных систем компьютерной графики и анимации. Различные системы создания трехмерной графики широко применяются в промышленности при проектировании различных систем, индустрии компьютерных игр, кинематографе. В большинстве систем, таких как 3D MAX, Maya, Bryce 3D, AutoCAD, ProEngineer, Catia и т.д. в первую очередь решаются задачи построения 3D изображений, чертежей, анимации сцен. В то же время в этих системах отсутствует возможность создания сценариев, в которых сможет принимать участие пользователь. Это приводит к тому, что разработка виртуальных тренажеров или систем, которые бы моделировали динамические изображения и предоставляли возможность пользователю участвовать в этих сценах, является очень сложной задачей, и чаще всего подобные системы разрабатываются специально для какой-то конкретной задачи, например, тренажер для водителя. Такие подходы не дают возможности широкого применения интерактивной 3D графики в процессе обучения персонала или моделировании ситуаций, поскольку разработка таких систем является достаточно ресурсоемким процессом. В то же время наличие систем, способных визуализировать различные процессы и предоставлять пользователю возможность участвовать в этих процессах, позволит решать широкий круг задач, включая разработку тренажеров, систем визуализации сцен, систем визуального моделирования.

Учитывая актуальность задачи разработки таких систем, сравнительно недавно начало развиваться новое направление в системах трехмерной графики, которое можно характеризовать как направление разработки систем построения интерактивных трехмерных сцен. В таких системах предполагается, что пользователь может самостоятельно разрабатывать трехмерную сцену, сценарии для объектов сцены и сценарии взаимодействия пользователя с объектами на сцене. При помощи таких систем можно смоделировать управление автомобилем, самолетом, вертолетом, кораблем, технологической линией, ландшафт местности и показать, например, площадь затопления, а также продемонстрировать различные физические опыты, которые требуют лабораторных условий и специального оборудования. В то же время

визуальное моделирование для указанных ситуаций позволит осуществить не только демонстрацию какого-либо процесса, но и вовлечь пользователя в этот процесс и, что важно, отслеживать результаты его действий в различных ситуациях. Как следствие, появится возможность оценивать знания и навыки пользователя в работе с той или иной системой или его действиях в той или иной ситуации, поскольку действия будут продиктованы решениями, принятыми на основе знаний о той или иной области.

Введем определения терминов, которые будут широко использоваться в дальнейшем.

**Сцена** - это визуальное представление трехмерного пространства, на котором изображаются все объекты, включая фон, с которыми происходят различные действия.

**Сценарий** - это описание последовательности действий пользователя и объектов на сцене, включая взаимодействие объектов между собой и взаимодействие пользователя с объектами. Сценарий создается для каждой сцены и описывается при помощи специального скриптового языка, а также путем задания определенных параметров объектов сцены.

Несколько лет назад началась разработка подобных систем, однако существующие на сегодняшний день системы в первую очередь ориентированы на разработку игр, что существенно сужает возможности их использования в других сферах деятельности. Рассмотрим наиболее известные системы.

**Система OxGameEngine [1].** Это система с открытым исходным кодом с достаточно узким набором функциональности. Система ориентирована на разработку различных игр и 3D приложений для платформы XBox. Она имеет полностью компонентную архитектуру. Создание проекта пользователем осуществляется путем изменения характеристик в окне properties (свойства). Существует встроенная скрипт система, но пользователь не имеет к ней доступа. Из функций редактирования доступен визуальный редактор сцены и редактор объекта игры с ограниченными возможностями с абстрактной системой построения вершин и геометрии, а также возможность подключения видео и музыкальных файлов. Для управления камерой используется обобщенная система управления путем привязки камеры к объекту и выбора типа камеры из списка. Принцип построения освещения аналогичен управлению камерой. Управление физикой движения возможно лишь путем задания кривой движения в редакторе с помощью мыши или клавиатуры, а

также выбора одной из стандартных функций поведения. Система управления тенями реализована с использованием мягких теней. При генерации изображений используются шейдеры второго и третьего уровней. В системе предусмотрен незначительный набор предопределенных моделей, которые нельзя дополнять (на момент анализа системы).

**Система Visual3D.NET** [2]. Коммерческая система с различными вариантами лицензирования и широким набором функций. Система ориентирована на разработку игр для платформы Windows. В системе доступно значительное количество встроенных моделей, которые моделируют реальный мир, включая модели земного шара. Все объекты имеют значительное число параметров, и основной упор сделан на использование свойств объектов, а не на программирование с использованием скриптового языка, который имеет ограниченные возможности, и в основном ориентирован на малые изменения и дополнения. Принцип описания поведения объектов, света и камер аналогичен предыдущему проекту, но более функционален. Среди интересных возможностей следует отметить возможность модификации требований игры во время ее выполнения без перезапуска системы, встроенную сетевую подсистему, очень высокий уровень прорисовки графических объектов, также в системе широко используются многофункциональные визуальные обработчики. Система достаточно проста в применении широким кругом пользователей.

**Система Blade3d** [3]. Проект, ориентированный на разработку игр на платформах Windows и Xbox 360. В системе поддерживается импорт из 20 различных форматов файлов для хранения моделей, включая форматы dae (collada), .fbx, and .x, поддерживаются различные графические форматы, аудио файлы, поддерживаются стандартные HLSL шейдеры и эффекты, различные возможности анимации текстов. В дизайнера сцен реализован редактор виртуального мира со всеми необходимыми функциями. Система работы с материалами интегрирована с HLSL редактором с поддержкой синтаксической раскраски и обнаружения ошибок. Реализованы мощные механизмы отображения подсветки и теней с различными эффектами и настройками, широкие возможности создания поверхности на основе карты высот, мультитекстурных поверхностей и поверхностей, заданных нормальными, создание воды, включая волны, реки, дороги, облака, реализованы все необходимые эффекты реалистичного отображения водной поверхности.

В системе интегрирована физическая подсистема, которая позволяет реализовывать всевозможные законы механики, необходимые для реалистичного отображения процессов взаимодействия объектов. Скриптовый язык, реализованный в системе, позволяет выполнять самые различные сценарии взаимодействия объектов и поведения пользователя, в том числе предусмотрена возможность расширения языка самим разработчиком. В целом система является самой мощной среди всех рассмотренных систем.

Рассмотренные системы содержат необходимые функции для построения сцен и создания сценария поведения объектов на сцене. В тоже время в этих системах отсутствуют такие возможности:

- моделирование и визуализация ситуаций, которые выходят за рамки обычной компьютерной игры;
- учет действий пользователя в определенных ситуациях с целью их сохранения и анализа;
- реализация пользователем и использование им при построении сцен моделей необходимых процессов на основе определенных математических и физических законов.

## 2. Система моделирования интерактивных 3D сцен

На основе анализа, проведенного выше, определен набор требований к системе построения интерактивных 3D сцен:

- Наличие редактора сцены с необходимыми функциями построения сцен
- Скриптовый язык и редактор для описания сценариев поведения объектов
- Физическая платформа с необходимыми функциями моделирования процессов механики и взаимодействия объектов
- Возможности расширения системы
- Импорт моделей из файлов формата .X и файлов в графических форматах
- Задание параметров поверхности с использованием карты высот
- Возможность записи и последующего воспроизведения действий на сцене
- Возможность оповещения об ошибках и учета ошибок, совершенных пользователем в процессе работы со сценой
- Наличие предопределенных наборов сценариев и возможностей их комбинации в новой сцене

Система разрабатывается с использованием технологий .NET Framework 3.5 и XNA Game Studio 3.1 [4, 5], что позволяет использовать новейшие разработки в области компьютерной

графики. Благодаря использованию данной платформы появляется возможность использовать платформу моделей физических процессов предоставляемую системой XNA, систему шейдеров и многие другие функции, что

значительно сокращает временные затраты на разработку системы и ее дальнейшее развитие. Структурная схема системы приведена на рисунке 1.



Рисунок 1. Структура системы построения интерактивных сцен

Из приведенной схемы (рис 1.) следует, что система состоит из нескольких наиболее значащих частей, которые рассмотрим далее.

Редактор сцен используется для построения как самой сцены, так и сценария, по которому будет работать сцена. Данное приложение является главной частью системы, поскольку оно реализует все необходимые функции по

построению сцены, добавлению объектов и настройке их свойств, редактированию скрипта, созданию сценариев. Интерфейс приложения разработан в стиле Microsoft Office 2007 и является простым и интуитивно понятным для пользователя. Главное окно редактора сцен с построенной сценой показано на рисунке 2.



Рисунок 2. Интерфейс редактора сцен системы построения интерактивных сцен

Вся сцена, объекты, поведение объектов, действия пользователя записываются редактором в единый сценарий, который выполняется системой воспроизведения сцены. Встроенный анализатор скрипт-кода проверяет корректность его написания независимо от того, генерировался ли он автоматически редактором, либо пользователь вносил в него изменения.

Процесс создания пользователем сцены в Редакторе Сцен можно разбить на несколько этапов:

1. Определение с помощью скриптового языка, конструкторов, расположенных в главном меню, палитры компонентов, объектов, что будут представлены на сцене (ландшафт, деревья, здания и т.д.). А также указываются последовательности прорисовки этих объектов или задание момента их появления на сцене.

Особенность данного процесса состоит еще в том, что использование палитры компонентов и возможностей меню полезно для прорисовки статических объектов или объектов, которые нужно прорисовать только один раз. Прорисовку объектов, для которых задаются моменты появления, случайность или ритмичность появления целесообразно задавать в скрипт-системе. Например:

- случайное распределение и момент появления капель дождя на сцене (определяется с помощью скрипта)
- прорисовка объекта в момент какого-либо действия, например, машина доехала до определенного места на дороге, и выбежал заяц (определяется с помощью скрипта)
- последовательность прорисовки объектов на сцене с учетом поля зрения камеры (вывод осуществляется таким образом, что объекты отрисовываются непосредственно перед их отображением на экране)

Данная последовательность сохраняется в класс `ObjectList`, который технически является просто списком указателей на заданные пользователем объекты, чтобы в дальнейшем последовательно прорисовать указанные объекты.

2. Определение физических свойств объектов осуществляется с помощью скриптового языка, либо пункта меню `Physics`. Функции, доступные в данном меню, используются для задания стандартных физических свойств объектов (течение воды и отражение света, движение автомобиля и т.д.)

В этом же меню задаются варианты физического взаимодействия между объектами системы (закон импульса и трения и т.д.). Взаимодействия такого рода так же, как и иные

модели, основанные на законах физики и математических моделях процессов, которые необходимо использовать, следует реализовывать с применением скриптового языка.

Стандартные свойства объектов, одинаковые для всей сцены, такие как освещение и отражение света, также задаются в данном пункте меню. Причем в данном случае освещение будет одинаково для всей сцены, а характеристики отражения будут изменяться в зависимости от физических свойств объекта, т.е. вода, зеркало и т.д. Данные атрибуты также сохраняются в `ObjectList` и применяются в момент создания объектов по указанным пользователем классам.

3. После того как сцена, а следовательно, и сценарий ее выполнения готовы, осуществляется ее сохранение в файл, которое сводится к сохранению списка `ObjectList`, последовательности действий пользователя и его взаимодействие со сценой. После этого сценарий готов к использованию системой воспроизведения.

Система воспроизведения сцены предназначена для воспроизведения созданного сценария и сцены пользователем. Данная подсистема реализована в виде отдельного приложения, которое получает на вход сценарий и воспроизводит его пользователю. При этом происходит регистрация действий пользователя, а также передача их в подсистему анализа результатов работы пользователя.

### 3. Язык сценариев

Сценарий представляет собой набор команд на языке близком к языку `C#`, что позволяет быстро его изучить. Для выполнения сценария необходимо составить набор команд, которые впоследствии будут интерпретированы и обработаны основным движком программы, что позволит динамически (т.е. не перезагружая программу) наблюдать изменения на создаваемой сцене. Следует отметить, что созданная скрипт система используется для манипулирования объектами на сцене, а не для ее непосредственного создания. Скриптовый код разделен на несколько основных категорий:

1. Основные программные функции, которые применяются ко всем объектам сцены.
2. Команды для управления статическими объектами сцены
3. Команды для управления динамическими объектами сцены
4. Специальные команды – это команды для работы с конкретной задачей. Например: `RunCars(road_name, time)`; запустит автомобильное движение на

конкретной дороге в указанное время.  
Эта команда доступна только в модуле управления дорожным движением

Следует отметить, что специальные команды добавляются для решения каких-либо конкретных задач при моделировании для определенной предметной области.

Примером команды глобального уровня есть создания ландшафта:

```
Terrain terrain = new Terrain("Content/heightmap",  
"Content/grass");
```

где *heightmap* – это карта высот, *grass* – текстура.

В программировании логики применяется основные типы конструкций:

1. Операторы сравнения, суммирования, умножения, деления
2. Цикл `for i = (start_index, end_index)`  
и `for i = (start_index, end_index, iteration)`
3. условный оператор `if (x сравнение y)`

В системе воспроизведения сцены, предусмотрена возможность реализации подсистемы учета результатов работы. Данные результаты могут затем анализироваться с целью выявления ошибок при работе на интерактивной сцене. Данная функциональность необходима в случае использования системы для построения тренажеров. Анализ результатов предполагается выполнять при помощи отдельного приложения.

#### 4. Заключение

Сферой применения данной системы является ее использование при разработке различных тренажеров или тренировочных курсов для подготовки водителей автомобилей, пилотов, судоводителей, при моделировании различных ситуаций. Например, с использованием системы можно построить

ландшафт реальной местности и посмотреть зоны затопления, в случае наводнения моделируя уровень подъема воды, в случае разлива какой-либо жидкости из трубопровода или цистерны. Конечно, данная система не заменит обучение в реальных условиях, но понять и попробовать многие аспекты поведения реальных систем будет возможно. Тем более, что с помощью данной системы можно очень быстро менять или создавать сценарии различных ситуаций и, как следствие, проверять действия человека в этих ситуациях. Для повышения реалистичности картины монитор можно заменить проектором и осуществлять проецирование изображения сцены на экране.

Разработанная система является основой для системы построения интерактивных 3D сцен, так как в ней реализованы все необходимые компоненты для построения и воспроизведения сцены с участием пользователя и отслеживания его действий на сцене.

В дальнейшем предполагается развивать систему в направлении совершенствования системы построения сцен, наполнять систему готовыми сценами, объектами, создавать отдельные сценарии для демонстрации возможностей системы, развивать систему скриптового языка, а также осуществить подробное документирование системы.

[1] Интернет ресурс <http://oxgameengine.codeplex.com>

[2] Интернет ресурс <http://www.visual3d.net>

[3] Интернет ресурс <http://www.blade3d.com>

[4] Горнаков С.Г. Программирование компьютерных игр под Windows в XNA Game Studio Express.- М.: ДМК Пресс, 2008. – 384с.

[5] Alexandre Lobro, Bruno Evangelista, Jose Antonio Leal de Farias. Beginning XNA 2.0 Game Programming: From Novice to Professional. – NewYork.: Springer-Verlag, 2008. – 457p