§1 ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Сморкалов А.Ю., Кирсанов А.Н.

СРЕДСТВА ПРОГРАММИРОВАНИЯ ПОВЕДЕНИЯ БОТОВ В ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ

Аннотация: *В последнее время виртуальные миры неуклонно расширяют сферу своего применения в образовании. Тренажеры, симуляции, ролевые и серьезные игры являются наиболее удачными для обучения в виртуальных средах. Важной частью вышеперечисленных подходов к обучению являются педагогические агенты (боты), которые участвуют в процессе обучения и помогают студенту выполнить учебное задание. В виртуальном мире vAcademia поддерживается реализация активных форм обучения с помощью языка vJS, однако использование и программирование ботов до настоящего времени было недоступно. В статье рассматривается система управления ботами, которая позволяет каждому пользователю vAcademia размещать и настраивать ботов, а также задавать их поведение с помощью расширенного языка vJS. Программирование поведения ботов реализовано на основе использования объектно-ориентированного подхода, автосинхронизируемых функций, возможности задания последовательности выполнения асинхронных действий, а также организации взаимодействия с пользователем на основе озвученных текстовых диалогов с выбором варианта ответа. Взаимодействие с запрограммированными ботами может быть записано в виде 3D-записи для последующего просмотра, что имеет большое значение в образовательной сфере.*

Ключевые слова: *виртуальные миры, виртуальные среды, образовательные инструменты, виртуальность, боты, языки программирования, скрипты, встроенные языки программирования, синхронизация, аватары*

Введение.

В последнее время виртуальные миры занимают всё большее место в обучении. Они являются местом встреч самых престижных университетов и некоммерческих академий по всему миру. Их использование многогранно и подходит для удовлетворения различных потребностей в обучении. Они могут являться площадкой для проведения дискуссий. Некоторые преподаватели используют виртуальные миры в качестве площад-

DOI: 10.7256/2305-6061.2014.2.12679

ки для встреч студентов. Слушатели могут прослушать не только лекцию, но и увидеть выложенные преподавателем видео, иллюстрации, книги и 3D модели.

Неотъемлимой частью виртуальных миров являются боты. Бот—управляемый в автоматическом режиме некоторой программой трехмерный персонаж, предназначенный для выполнения или имитации действий, обычно выполняемых человеком или управляемым пользователем аватаром.

Один из вариантов использования ботов в виртуальных мирах – боты-помощники. Например, в мультимедийном продукте "Химия. 8-11 класс. Виртуальная лаборатория" [1] бот подсказывает пользователю, что в текущий момент нужно делать и сообщает пользователю, если он допустит ошибку (рис. 1).

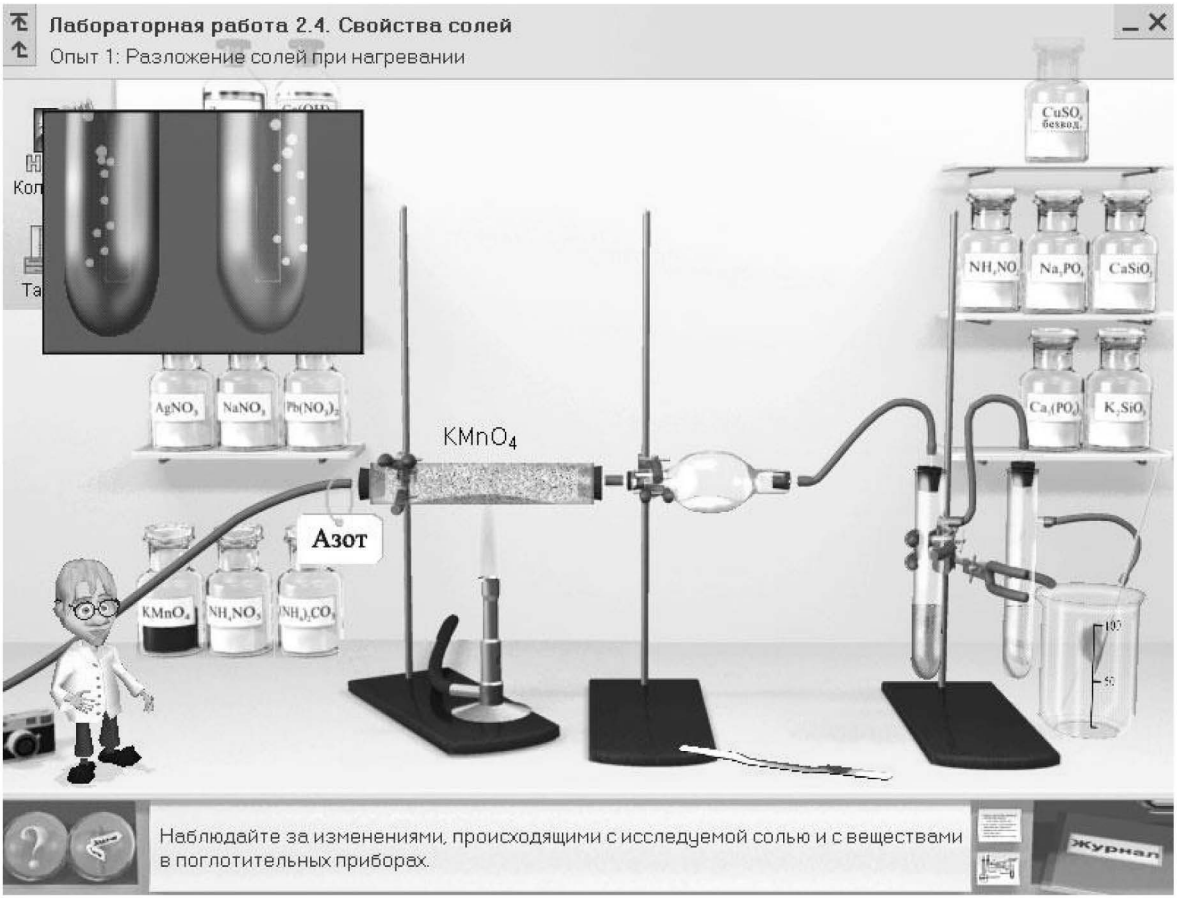

Рис. 1. Бот-помощник в обучающей программе "Химия. 8-11 класс. Виртуальная лаборатория"

Часто боты используются и как полноценное действующее лицо в образовательных сценариях, например, как один из собеседников в языковых тренажерах или ролевых играх. Например, в модулях ЭОР совместной образовательной деятельности для поддержки коллективной работы учащихся [2] боты участвуют в языковых диалогах, продолжая общение только тогда, когда пользователь выбирает правильный с точки зрения грамматики ответ (рис. 2).

DOI: 10.7256/2305-6061.2014.2.12679

Рис. 2. Модуль ЭОР по английскому языку

Исходя из всего вышесказанного, разработка средств программирования поведения ботов для образовательного виртуального мира vAcademia является актуальной задачей.

Обзор языков программирования в виртуальных мирах.

Скриптовые языки программирования легки в освоении, поэтому они нашли свое применение в том числе для взаимодействия с виртуальной средой. Использование таких языков значительно облегчает программирование поведения ботов.

Язык LSL [3] , использующийся в виртуальном мире Second Life, является один из самых популярных языков программирования в виртуальных мирах. Это скриптовый язык программирования похожий по синтаксису на C. В языке реализована поддержка машины состояний. Каждое состояние именовано и содержит описание действий, которые выполняются, когда объект находится в этом состоянии. LSL содержит в себе около 300 функций для управления 3D-объектами и ботами.

LSL не поддерживает объектно-ориентированное программирование, не поддерживает типы данных, удобные для работы в 3D-пространстве, например, массивы или повороты, задаваемые в удобной форме (предоставляется возможность использовать кватернионы, однако задание поворотов в форме кватернионов является сложной задачей для программистов, чьей прямой специализацией не является программирование

DOI: 10.7256/2305-6061.2014.2.12679

компьютерной графики). Кроме того, LSL-скрипт исполняется на стороне сервера, что может привести к проблемам функционирования виртуального мира из-за неверно написанных пользователями скриптов. Большим недостатком является то, что LSL не поддерживает повторное использование кода. Таким образом, LSL имеет определенные функции для управления ботами, однако из-за ряда недостатков использование этого языка оказывается затруднительным.

Improv [4] – это система для создания поведения анимированных персонажей. Improv состоит из двух подсистем. Первая подсистема – анимационный движок. Второй – подсистема поведения. Подсистема поведения программируется скриптовым языком. Язык содержит в себе команды для выполнения атомарных действий (например, "Идти" или "Помахать рукой"), которые выполняются анимационной подсистемой. Для изображения эмоций и движений анимационная подсистема деформирует модель обьекта. Анимационная система деформирует модель пошагово для создания плавных переходов. Недостатком системы является отсутствие адаптации для виртуальных миров, в частности необходимость поддержки синхронизации внешними средствами.

vAcademia [5] – это образовательная платформа, предоставляющая сервисы, с помощью которых можно проводить и посещать учебные курсы, совещания, презентации, тренинги для групп от одного до нескольких десятков пользователей одновременно. Каждый посетитель представлен специальным визуальным объектом – аватаром. Аватары пользователей синхронизируются – все действия каждого из них видны остальным в реальном времени.

Язык vJS [6] предназначен для программирования поведения трехмерных объектов внутри одной локации виртуального мира vAcademia. Язык представляет собой расширенный JavaScript, поддерживает все языковые конструкции базового JS и ориентирован на объектно-ориентированное программирование. vJS-программа исполняется локально на каждом клиенте виртуального мира, находящемся в данной локации.

Язык позволяет управлять всеми пользовательскими объектами внутри одной локации, обращаясь к ним по имени. По имени объекта можно получить ссылку на него, а имея ссылку поставить обработчики на любые события.

Работа программы не начинается до тех пор, пока все объекты, используемые программой, не будут загружены. Это серьезно упрощает программирование в виртуальных мирах, где стандартом «де-факто» считается стриминг объектов и ресурсов. vJS содержит в себе как автоматически синхронизируемые функции и свойства, так и функции, которые работают локально, что также упрощает процесс задания поведения объектов.

vJS-программа не имеет смысла без объектов, которыми она управляет. Поэтому, для того, чтобы сохранить программу, необходимо сохранить шаблон локации. В шаблон локации сохраняется как сама vJS-программа, так и все расставленные в данный момент объекты (со всеми их свойствами). Такой подход позволяет с максимальным удобством повторно использовать vJS-программы.

Таким образом, существующие языки, позволяющие осуществлять управление ботами, имеют существенные недостатки. Язык vJS позволяет управлять только 3D-объектами,

DOI: 10.7256/2305-6061.2014.2.12679

однако не содержит в себе описанных недостатков. В данной статье описывается расширение языка vJS для задания поведения ботов.

Система управления ботами и их настройки.

Язык vJS предназначен для управления заранее размещенными и настроенными 3D-объектами. Такой подход избавляет от необходимости рутинного программирования создания объектов и задания их основных свойств, таких как координат, поворота, масштаба и других. Кроме того, язык vJS управляет только именованными объектами – т.е. объектами, для которых было задано имя для программирования.

Для интеграции управления ботами в vJS по аналогии был разработан инструмент настройки и размещения ботов (см. рис. 3). Он предоставляет следующие возможности:

- Создание нового бота в выбранной точке
- Удаление выбранного бота
- Визуальное перемещение и настройка поворота бота
- Визуальная настройка масштаба отображения бота
- Ввод имени бота для программирования и имени бота для отображения над ботом.
- Визуальная настройка внешности бота

Рис. 3. Инструмент настройки и размещения ботов.

Таким образом, с помощью инструмента расстановки и настройки ботов можно визуально разместить ботов и настроить внешность и основные свойства. Подготовленная 3D-сцена с ботами может управляться языком vJS с целью реализации какого-либо образовательного сценария.

DOI: 10.7256/2305-6061.2014.2.12679

Основные подходы к управлению ботами на vJS.

Управление ботами осуществляется на основе объектно-ориентированного подхода. Каждый бот представлен экземпляром класса бота. С помощью метода getBotByName глобального объекта scene получаются ссылки на экземпляры классов бота по имени бота. Имя для программирования хранится в синхронизируемой переменной экземпляра бота, что позволяет обращаться к ботам в локально выполняющейся vJS-программе на каждом клиенте виртуального мира. vJS-программа не начинает работу до тех пор, пока не будут созданы все боты, что существенно упрощает логику построения vJS-программ. Как и у 3D-объекта, у бота доступны автоматически синхронизируемые свойства position, rotation, scale.

У экземпляра класса бота возможно переопределение обработчиков всех событий, доступных для 3D-объектов, а также специфичных для бота событий. Определением обработчиков событий достигается взаимодействие между пользователем и ботом. Кроме того для взаимодействия пользователя и бота предназначена несинхронизируемая функция отображения диалогового окна с вопросом и выбором варианта ответа (см. рис. 4) showDialog. Как вопрос, так и ответы на него могут быть озвучены. Вопрос озвучивается в момент появления окна с вопросом. Ответ озвучивается при его выборе пользователем. Если окно появляется локально (на одном клиенте виртуального мира), то звуки проигрываются синхронизированно – на всех клиентах виртуального мира, находящихся в текущей локации, что имеет большое образовательное значение, т.к. другие посетители могут слышать происходящий диалог как будто бы он происходит «в живую», с общением через голосовую связь.

Во все методы и функции работы с ботом интегрирована обобщенная проверка параметров вызова. Если параметры вызова функции или метода имеют неверные типы или не входят в диапазоны допустимых значений, то выводится подробное сообщение об ошибке в отладочную консоль, а вызов функции не осуществляется. Это значительно облегчает процесс отладки.

Таким образом, реализация поддержки управления ботами в vJS основана на объектно-ориентированном подходе, именовании ботов и получении ссылки на экземпляр бота по имени, обработке событий взаимодействия с пользователем и организации диалога с пользователем на основе выбора вариантов.

Автосинхронизируемые методы.

Автоматически синхронизируемые методы бота представляют собой методы класса бота, результат действия которых автоматически синхронизируется между всеми клиентами виртуального мира. Это значит, в частности, что достаточно вызвать определенное действие бота на одном клиенте виртуального мира, чтобы оно воспроизвелось у всех посетителей текущей локации. Кроме того, результат действия всех автосинхронизируемых методов может быть записан в форме 3D-записи [7, 8], а затем многократно просмотрен, в

DOI: 10.7256/2305-6061.2014.2.12679

том числе с использованием механизма изменения текущего проигрываемого фрагмента.

Реализация каждого автосинхронизируемого метода делится на три части:

1. Непосредственно метод, который устанавливает значения синхронизируемых переменных для синхронизированного воспроизведения действий у всех пользователей. Метод вызывается на одном из клиентов виртуального мира.
2. Обработчик изменения синхронизируемых переменных, который непосредственно выполняет требуемое действие. Обработчик выполняется на всех клиентах виртуального мира.
3. Инициализация значений синхронизируемых переменных, а также их обработка события необходимости сброса их значения в значения по умолчанию. Инициализация значений нужна для корректной перемотки на начало записи, которая была бы невозможна, если бы на начальный момент записи не было бы установленных значений синхронизируемых переменных, а сброс значений синхронизируемых переменных необходим для повторного использования 3D-объектов при переходе в 3D-запись сразу после записи события.

Автосинхронизируемые методы предоставляют функциональность, которая эквивалентна возможностям аватара, управляемого пользователем, в том числе:

- Метод doGesture позволяет выполнить жест по его имени.
- Метод doGestureInCycle позволяет выполнить жест по его имени несколько раз подряд.
- Метод clearAllGestures позволяет остановить все жесты, перевести бота в положение по умолчанию.
- Метод playSound позволяет проиграть звук из серверного ресурса с заданным уровнем громкости. Бот на протяжении проигрывания звука воспроизводит скелетную и лицевую анимацию произношения речи.
- Метод stopSounds позволяет остановить проигрывание всех звуков.
- Метод goToPos позволяет направить бота идти шагом в заданную точку.
- Метод teleport позволяет перенести бота в заданную точку моментально.
- Метод sit позволяет посадить бота на 3D-объект с размеченными местами для сидения на место с заданным номером.
- Метод writeToChat позволяет написать от имени бота текстовое сообщение в чат виртуального мира текущей локации.
- Метод applyMimic позволяет отобразить на лице бота заданную эмоцию.
- Метод pointerShow позволяет показать из руки бота указку, направленную в заданную точку.
- Метод pointerHide позволяет убрать указку из руки бота.
- Метод pointerShowForTime позволяет показать из руки бота указку на заданное время.

Особый интерес представляет синхронизация длительных действий, например, проигрывание звука или осуществление жеста. Т.к. синхронизируемые переменные хранят в себе текущее состояние объекта, то единожды установленные синхронизируемые

DOI: 10.7256/2305-6061.2014.2.12679

переменные будут означать состояние как в текущий момент времени, так и в любой другой до смены их значений. Это может привести к воспроизведению автосинхронизируемого метода в любой последующий момент после действия, например, при перемотке 3D-записи или входе пользователя в локацию виртуального мира.

Поэтому длительные действия сохраняются в виде набора синхронизируемых переменных, в который обязательно включаются:

1. Время начала длительного действия T_1 относительно времени сервера виртуального мира. Время сервера виртуального мира периодически посылается на каждый клиент виртуального мира, а текущее значение получается сложением значения пришедшего времени и разницы локального времени клиента между запросом времени и последним получением времени с сервера.
2. Длительность синхронизируемого действия T.

Подобный подход позволяет определить:

1. Необходимость воспроизведения длительного действия. Действие необходимо воспроизвести, если текущее время T_T входит в диапазон $[T_1, T_1 + T]$.
2. Временной сдвиг выполнения действия при необходимости воспроизведения длительного действия не с начала, а с некоторого момента. Абсолютный временной сдвиг равен $\Delta T_A = T_T - T_1$ (1)
   а относительный временной сдвиг измеряемый в диапазоне [0, 1] равен
   $\Delta T_A = (T_T - T_1) / T$ (2)

Таким образом, для языка vJS был предложен набор автоматически синхронизируемых методов бота, который обеспечивает аналогичный функционал, который доступен при управлении аватаром пользователя. Автоматически синхронизируемые методы обеспечивают корректное воспроизведение как мгновенных, так и длительных действий, в том числе как в реальном времени, так и в момент просмотра 3D-записи.

Последовательность асинхронных действий бота.

Образовательные сценарии часто требуют выполнения ботом не одного, а сразу нескольких действий, одного за другим, например, «подойти в нужную точку, помахать рукой, сказать фразу». Каждое из этих действий является асинхронным, т.е. время выполнения напрямую зависит от аргументов функции и чаще всего не может быть определено до времени исполнения vJS-скрипта.

Такая последовательность действий может быть реализована за счет обработки окончания длительных действий:

- onSoundPlayed – бот закончил проигрывание звука
- onBotMoved – бот закончил движение
- onGestureDone – бот сделал жест
- и других

В этом случае скрипт будет представлен множеством обработчиков вложенных друг в друга событий. Структура кода при таком подходе становится чрезвычайно запутанной,

DOI: 10.7256/2305-6061.2014.2.12679

а число строк, необходимых для реализации последовательности действий, непомерно большим. С целью устранения этого недостатка для класса бота была предложена глобальная функция playSequence последовательного исполнения асинхронных действий с ожиданием текущим действием выполнения предыдущего до своего запуска. Метод playSequence принимает параметром специальный массив действий с сопоставленным действию ботом. Таким образом, вызов playSequence может управлять не только одним ботом, но и несколькими. Пример массива действий приведен в листинге 1.

```
var actionsArray = createActionsPool();
actionsArray.addAction(bot1, ‘doGesture(“agree”) ‘);
actionsArray.addAction(bot1, ‘playSound(speech1, 100)’);
actionsArray.addAction(bot2, ‘goToPos(62385, 23600)’);
```

Листинг 1. Пример задания массива действий.

Таким образом, для управления последовательными длительными действиями бота был предложен механизм последовательности действий. Последовательность действий позволяет осуществлять выполнение один за другим как мгновенно выполняемых, так и длительных действий одного или нескольких ботов без необходимости обработки событий окончания длительных действий.

Заключение.

Рис. 4. Языковой тренажер по английскому языку.

DOI: 10.7256/2305-6061.2014.2.12679

Были разработаны расширения языка vJS для управления ботами и инструмент визуального размещения и настройки ботов. Совокупность программных средств позволяет реализовывать образовательные сценарии с использованием ботов. Как пример использования vJS для программирования ботов был разработан языковой тренажер по английскому языку (см. рис. 4), где бот выполнял роль собеседника, который реагирует только на грамматически правильные ответы.

Как средство программирования ботов был предложен набор автоматически синхронизируемых функций, которые существенно сокращают сложность программирования в многопользовательской среде и дают доступ к аналогам всех функций управляемого пользователем аватара. Программирование поведения ботов реализовано на основе использования объектно-ориентированного подхода, возможности задания последовательности выполнения асинхронных действий, а также организации взаимодействия с пользователем на основе озвученных текстовых диалогов с выбором варианта ответа. Взаимодействие с запрограммированными ботами может быть записано в виде 3D-записи для последующего просмотра, что имеет большое значение в образовательной сфере.

Библиография :

1. Morozov M., Tanakov A., Gerasimov A., Bystrov D., Cvirco V. "Virtual Chemistry Laboratory for School Education." The 4th IEEE International Conference on Advanced Learning Technologies (ICALT). 30 August-1 September 2004, Joensuu, Finland. IEEE Computer Society 2004, ISBN 0-7695-2181-9, pp.605-608.
2. Морозов М.Н., Герасимов А.В., Курдюмова М.Н. Совместная образовательная деятельность школьников на основе компьютерных сетей // Школьные технологии. 2009. №4. С. 78-88.
3. Michael Rymaszewski, Wagner James Au, Mark Wallace, Catherine Winters, Cory Ondrejka, Benjamin Batstone-Cunningham, Philip Rosedale: "Second Life: The Official Guide" December 2006, ISBN: 978-0-470-09608-6
4. Ken Perlin, Athomas Goldberg: :Improv: a system for scripting interactive actors in virtual worlds: // SIGGRAPH '96 Proceedings of the 23rd annual conference on Computer graphics and interactive techniques — 1996. — pp. 205-216.
5. Mikhail Morozov, Alexey Gerasimov, Mikhail Fominykh, and Andrey Smorkalov: "Asynchronous Immersive Classes in a 3D Virtual World: Extended Description of vAcademia," in Marina Gavrilova, Chih Jeng Kenneth Tan and Arjan Kuijper Eds., Lecture Notes in Computer Science (LNCS) – Transactions on Computational Science (TCS), Volume 7848, Issue XVI, 2013, Springer, Series ISSN: 0302-9743, ISBN: 978-3-642-38802-6, Extended version of CW 2012. DOI: 10.1007/978-3-642-38803-3_5
6. Сморкалов А.Ю.. Дизайн и архитектура среды выполнения языка программирования виртуальной реальности // Программные системы и вычислительные методы.-2014.-№ 1.-С. 104-107. DOI: 10.7256/2305-6061.2014.1.11328
7. Mikhail Morozov, Alexey Gerasimov, and Mikhail Fominykh: "vAcademia-Educational Virtual World with 3D Recording," in Arjan Kuijper and Alexei Sourin ed. the 12th International Conference on Cyberworlds (CW), Darmstadt, Germany, September 25-27, 2012, IEEE, ISBN: 978-0-7695-4814-2/12, pp. 199-206. doi>10.1109/CW.2012.35

DOI: 10.7256/2305-6061.2014.2.12679

8. М.Е. Рыженков Редактирование трехмерного образовательного контента // Программные системы и вычислительные методы. - 2013. - 1. - С. 95 - 105. DOI: 10.7256/2305-6061.2013.01.8.

References:

1. Morozov M., Tanakov A., Gerasimov A., Bystrov D., Cvirco V. "Virtual Chemistry Laboratory for School Education." The 4th IEEE International Conference on Advanced Learning Technologies (ICALT). 30 August-1 September 2004, Joensuu, Finland. IEEE Computer Society 2004, ISBN 0-7695-2181-9, pp.605-608.
2. Morozov M.N., Gerasimov A.V., Kurdyumova M.N. Sovmestnaya obrazovatel'naya deyatel'nost' shkol'nikov na osnove komp'yuternykh setei // Shkol'nye tekhnologii. 2009. №4. S. 78-88.
3. Michael Rymaszewski, Wagner James Au, Mark Wallace, Catherine Winters, Cory Ondrejka, Benjamin Batstone-Cunningham, Philip Rosedale: "Second Life: The Official Guide" December 2006, ISBN: 978-0-470-09608-6
4. Ken Perlin, Athomas Goldberg: :Improv: a system for scripting interactive actors in virtual worlds: // SIGGRAPH '96 Proceedings of the 23rd annual conference on Computer graphics and interactive techniques — 1996. — pp. 205-216.
5. Mikhail Morozov, Alexey Gerasimov, Mikhail Fominykh, and Andrey Smorkalov: "Asynchronous Immersive Classes in a 3D Virtual World: Extended Description of vAcademia," in Marina Gavrilova, Chih Jeng Kenneth Tan and Arjan Kuijper Eds., Lecture Notes in Computer Science (LNCS) – Transactions on Computational Science (TCS), Volume 7848, Issue XVI, 2013, Springer, Series ISSN: 0302-9743, ISBN: 978-3-642-38802-6, Extended version of CW 2012. DOI: 10.1007/978-3-642-38803-3_5
6. Smorkalov A.Yu.. Dizain i arkhitektura sredy vypolneniya yazyka programmirovaniya virtual'noi real'nosti // Programmnye sistemy i vychislitel'nye metody.-2014.-№ 1.-S. 104-107. DOI: 10.7256/2305-6061.2014.1.11328
7. Mikhail Morozov, Alexey Gerasimov, and Mikhail Fominykh: "vAcademia-Educational Virtual World with 3D Recording," in Arjan Kuijper and Alexei Sourin ed. the 12th International Conference on Cyberworlds (CW), Darmstadt, Germany, September 25-27, 2012, IEEE, ISBN: 978-0-7695-4814-2/12, pp. 199-206. doi>10.1109/CW.2012.35
8. M.E. Ryzhenkov Redaktirovanie trekhmernogo obrazovatel'nogo kontenta // Programmnye sistemy i vychislitel'nye metody. - 2013. - 1. - C. 95 - 105. DOI: 10.7256/2305-6061.2013.01.8.