

**ЧЕТВЕРТАЯ ОТКРЫТАЯ КОНФЕРЕНЦИЯ
ИССЛЕДОВАТЕЛЬСКИХ
И ПРОЕКТНЫХ РАБОТ ШКОЛЬНИКОВ
ПАМЯТИ Р.К. БЕГА**

**СБОРНИК МАТЕРИАЛОВ
СЕКЦИИ COMPUTER SCIENCE**

ЧЕТВЕРТАЯ ОТКРЫТАЯ НАУЧНАЯ КОНФЕРЕНЦИЯ

СОДЕРЖАНИЕ

График работы конференции.....	5
Вниманию участников и научных руководителей.....	5
СЕКЦИЯ COMPUTER SCIENCE	7
1. Бартунов Сергей, Зайцев Виктор (10 ^Г). Разработка системы удаленного взаимодействия объектов программно-аппаратного робототехнического комплекса.	7
2. Журавлев Андрей, Будков Александр (9 ^В). Программа для построения фракталов.	8
3. Варенцов Михаил (8 ^В). Логика совмещенного использования структур и массивов в игровом программировании.	8
4. Власюк Владимир, Кеворков Артем (8 ^В). Перенос программного игрового проекта из среды MS-DOS в MS Windows.	8
5. Воинов Андрей (10 ^А). Генерация водной 3D поверхности на OpenGL и трехмерное программирование под MS-DOS.	9
6. Волынский Андрей (9 ^А). Разработка движка для создания RPG-игр.	9
7. Гречаник Сергей (11 ^Г). Дифференцирование функций на этапе компиляции при помощи шаблонов C++.	10
8. Костин Григорий, Швец Павел (10 ^Г). Реализация ролевой игры на базе скриптового движка.	10
9. Кузнецов Иван (8 ^В). Алгоритм поворота объекта на C++ и другие геометрические преобразования тела.	10
10. Кузьмин Алексей (10 ^Г). Реализация системы частиц для визуализации графических эффектов.	11
11. Левкович-Маслюк Федор, Гречаник Сергей (11 ^Г). Моделирование газогидратов метана методом молекулярной динамики.	11
12. Остроумов Олег (10 ^А). Редактор HTTP-запросов.	12
13. Переславцев Алексей (10 ^В). Разработка генератора синтаксических LR1 анализаторов и алгоритмов восстановления после синтаксических ошибок LR1 разбора.	13
14. Рудой Георгий (9 ^В). Применение концепции MVC (Model-View-Controller) для портирования DOS -приложений под ОС Windows.	13
15. Федоров Владимир (11 ^Г). Использование компонентной архитектуры для создания модульных приложений.	14

16. Хохлова Марина (10^Г). Реализация программы построения сечений фигур. 14
17. Чикунов Егор, Устинов Владислав (10^Г). Редактор карт для трёхмерной игры
«Kayak Extreme»..... 15
18. Шарко Федор, Богданов Евгений (10^Г). Физическое моделирование. 16

МАТЕРИАЛЫ ЧЕТВЕРТОЙ ОТКРЫТОЙ НАУЧНОЙ КОНФЕРЕНЦИИ

СЕКЦИЯ COMPUTER SCIENCE

7-8 ДЕКАБРЯ 2005 ГОДА

ГРАФИК РАБОТЫ КОНФЕРЕНЦИИ

7 ДЕКАБРЯ, СРЕДА

- 9⁰⁰ – 9³⁰ — Открытие конференции
(Актовый зал, 1 этаж)
- 10⁰⁰ – 12⁰⁰ — Работа секций.
Регистрация участников.
- 12⁰⁰ – 12²⁰ — Перерыв.
- 12²⁰ – 14²⁰ — Работа секций.
- 14²⁰ – 14⁴⁰ — Перерыв.
- 14⁴⁰ – 17⁴⁰ — Работа секций.

8 ДЕКАБРЯ, ЧЕТВЕРГ

- 15⁰⁰ — Подведение итогов. Закрытие конференции.

ВНИМАНИЮ УЧАСТНИКОВ И НАУЧНЫХ РУКОВОДИТЕЛЕЙ

Все материалы докладов необходимо представить в Оргкомитет *заблаговременно, до начала конференции* для публикации в сборнике трудов конференции и на сайте школы в Интернете.

Материалы принимаются в электронной форме ответственными за секции.

По поводу использования проекционной техники и/или особых условий доклада просьба консультироваться с ответственными за секции.

Расписание докладов будет вывешено дополнительно во вторник, 6 декабря.

ИЗ ПОЛОЖЕНИЯ ОБ ОТКРЫТОЙ НАУЧНОЙ КОНФЕРЕНЦИИ ЛИЦЕЯ

Одной из важнейших задач специализированных школ является задача развития самостоятельных навыков исследовательской работы у школьников. Для поддержки ребят, заинтересованных в такой форме обучения, в лицее в 2002 году была проведена пробная научная конференция школьников. На ней было заслушано десять докладов по физике, математике и программированию. Хотя конференция была проведена в неудобное для школьников время (конец мая) и докладчики из других школ не приглашались, конференция, по нашим оценкам, прошла успешно: пришедшие школьники проявили большой интерес и активно участвовали в обсуждении содержания докладов. В связи с этим у организаторов конференции возникла мысль о необходимости более систематической работы со школьниками, проявившими интерес к научной работе, в течение всего года.

В связи с этим ежегодно в конце ноября — начале декабря устраивать традиционную научную конференцию школьников, на которой заслушивать доклады, отобранные оргкомитетом конференции. Так как эта конференция будет проводиться в сроки, предшествующие срокам проведения традиционных научных конференций школьников (зима-весна), то на неё будут допускаться также докладчики с незавершёнными исследованиями с целью активизации дальнейшей работы над темой и подготовки докладчиков к выступлениям на «традиционных» конференциях. Обязательным условием допуска докладов на Конференцию будет являться наличие в исследовании новых результатов и оценок (реферативные доклады организаторами поддерживаться не будут).

Важной особенностью нашей конференции является принципиальный отказ от идеи состязательности докладчиков. Этому есть две причины. Во-первых, формальная оценка качества докладов жюри создаёт неудобные условия для школьников, не успевших завершить свои исследования к моменту проведения конференции и ставит под вопрос их участие в подобном мероприятии. Вторая причина — каждое разделение докладов по качеству создаёт ненужные барьеры в общении школьников между собой и с членами жюри (к тому же оценки научных работ членами жюри всегда субъективны).

Другой важной особенностью конференции является ее открытость. К участию приглашаются (в качестве докладчиков и слушателей) все заинтересованные школьники г. Москвы, а также преподаватели и сотрудники школ и институтов.

Члены оргкомитета конференции лицея
А. В. Белов, И. Р. Дединский

СЕКЦИЯ COMPUTER SCIENCE

(Предс. И. Р. Дединский)

1. Бартунов Сергей, Зайцев Виктор (10^Г). Разработка системы удаленного взаимодействия объектов программно-аппаратного робототехнического комплекса.

Цель проекта – создать *framework* для ускоренной разработки, отладки и внедрения искусственного интеллекта, независимого от аппаратной среды робота.

Архитектура системы оперирует минимальной единицей комплекса – *модулем*. Основу функциональности системы представляют собой *сервисы*, объекты, подключенные к ядру с помощью модулей, и таким образом привязанные к ним. Модуль, подключая сервис к ядру, выступает в роли фабрики объектов, и имеет контроль над его внутренним состоянием и операциями. Базовая реализация сервиса предоставляет взаимодействие с другими объектами через обработку запросов.

Механизм расширения комплекса предоставляет полный доступ к ядру и позволяет бесконечно видоизменять систему, вплоть до замены ядра. Независимость от платформы обеспечивается использованием технологии .NET, которая уже была перенесена не только на мобильные операционные системы Windows, но и на Linux и MacOS.

Взаимодействие объектов внутри системы осуществляется на уровне *репрезентаций*, – разных проявлений объекта, имеющих общий внешний интерфейс, которое может происходить между разными процессами или даже компьютерами. Возможности распределенной системы предоставляет модуль соединения, который перенаправляет вызовы репрезентаций. Таким образом, операции с конкретным слоем производятся через репрезентации данного слоя, позволяя разделить различные задачи между несколькими взаимосвязанными компонентами.

Программа-пульт выполняет аппаратно-зависимые функции ввода-вывода, автоматически создавая GUI на основе предоставленного интерфейса репрезентаций, и устанавливает внутреннюю связь между графическим интерфейсом и аппаратной репрезентацией объекта, при необходимости передавая данные через модуль соединения. Для всего комплекса принят универсальный формат описания объектов – модификация языка XAML (Extensible Application Markup Language), основанного на XML, еще не имеющего официальной реализации, которая появится только в Windows Vista.

Базовое внедрения ИИ осуществляется созданием соответствующей репрезентации и подключения ее к модулю визуализации (иначе, *полигону*), который

симулирует *реальное* взаимодействие робота и внешнего мира. Подобное решение позволяет разрабатывать и отлаживать интеллект робота, не имея его аппаратной реализации, а также, облегчать ее, перенося вычислительно-арифметические задачи на внешний компонент. Данный модуль использует масштабируемый конвейер графического движка, работающий на основе технологии массивов вершин, абстрагируя от формата представления графических объектов, в качестве основного используя формат редактора 3DS Max.

2. Журавлев Андрей, Будков Александр (9^B). Программа для построения фракталов.

Самоподобные структуры, они же фракталы интересная область геометрии и алгебры. Изображения фракталов очень красивы и разнообразны. Фракталы удобны для построения природных объектов (деревьев, облаков и т.д.) так как с помощью простых формул можно задать сложные объекты. Также они используются для сжатия изображений и в некоторых других областях науки. Данная программа предназначена для изучения изображений некоторых фракталов. Программа отображает множество Мандельброта, и множества Жюлиа, позволяет приближать, удалять и перемещаться по изображениям, а также отображает некоторые геометрические фракталы. В ходе работы программы пользователь может сохранять полученные изображения в формате BMP с задаваемым разрешением.

3. Варенцов Михаил (8^B). Логика совмещенного использования структур и массивов в игровом программировании.

Основной целью доклада является решение проблем, связанных с использованием структур и массивов при игровом программировании. Будут рассмотрены следующие темы: логика создания наиболее удобных структур для описания игровых объектов, выделение общей части в полученных структурах, внедрение массивов из структур для увеличения количества игровых объектов, минимальная оптимизации программы. Использование этих методов значительно расширяет возможности игрового программирования, открывает новые возможности, помогает сделать код лучше и избежать многих «глюков» и ошибок. В качестве примера приведена простая игра, написанная в среде Borland C++.

4. Власюк Владимир, Кеворков Артем (8^B). Перенос программного игрового проекта из среды MS-DOS в MS Windows.

Цель работы – создание графического приложения (компьютерной игры) с использованием нестандартной библиотеки TXLib.h, в которой совмещены способы записи программного кода среда Borland C++ с функциями Visual C++.

В проекте использовано современные технологии компилятора g++ (Dev-CPP), а так же функции стандартной библиотеки компьютерной среды Windows.

Для многих начинающих разработчиков насущной проблемой является перенос проекта с одной платформы на другую (портирование). Одна из наших главных задач – помочь осуществить переход всем желающим и показать разницу между функциями Borland C++ и Dev – C++ (g++). Дать советы начинающим (в Dev-C++) и помочь тем, кто программирует в MS-DOS.

Один из «плюсов» этого перехода, это цветовая гамма и хорошее разрешение, а за счет этого, и использование заранее подготовленных ресурсов, таких как картинки (формата .bmp) и музыкальные файлы (формата .wav), ну и, конечно же, большая популярность.

5. Воинов Андрей (10^A). Генерация водной 3D поверхности на OpenGL и трехмерное программирование под MS-DOS.

Целью было создание анимированной поверхности воды на базе OpenGL и разработка программы под MS-DOS, способную визуализировать функции от трех параметров X, Y и Z.

Была создана анимированная поверхность воды с волнами, задающуюся с помощью тригонометрических функций. Для MS-DOSa была написана библиотека, в чем-то похожая на OpenGL (естественно, значительно менее мощная, но для экспериментов под DOS вполне подходящая).

В дальнейшем я надеюсь сделать редактор воды, устроить рендеринг, возможно написать 3D игру. Также в грядущем будут решаться проблемы реализации пены в «бочках», усложнение структуры воды.

6. Волынский Андрей (9^A). Разработка движка для создания RPG-игр.

Для максимальной имитации жизни были поставлены следующие задачи: возможность ходить по карте с указанием места назначения (алгоритм прохождения лабиринта), диалог (выбор вариантов ответов из имеющихся), бой.

Хотелось сделать максимально удобный интерфейс при создании игр; удобное управление и большие возможности в самой игре.

Для создания игр пришлось разработать несколько частей: редактор карт, редактор диалогов, редактор персонажей и инвентаря.

И сам движок, который создает игровой процесс, на основе созданных карт, диалогов и т.д. созданных редакторами.

На данный момент уже существует редактор карт. Он помогает создать матрицу свойств клеток на основе картинки карты. У клетки есть свойства, является ли она препятствием, дверью, переходом на другую локацию, если да, то координаты перехода и карту на которую ведет этот переход и описание клетки.

Сейчас уже ведется разработка редактора диалогов.

7. Гречаник Сергей (11^Г). Дифференцирование функций на этапе компиляции при помощи шаблонов C++.

Нередко в программах, занимающихся математическими или физическими расчётами, требуется нахождение производных некоторых функций. Если функция известна до этапа выполнения, то можно либо найти производную вручную, либо использовать специальные программы.

Предлагаемый способ позволяет автоматизировать поиск производной при помощи только компилятора C++. Способ основан на шаблонах выражений (Expression Templates) и шаблонном метапрограммировании – методе, позволяющем выполнять программы на этапе компиляции. Дифференциатор просматривает синтаксическое дерево, соответствующее выражению, а затем строит дерево, соответствующее производной.

Таким же способом возможно создание не только дифференциатора, но и других алгоритмов, работающих с функциями на этапе компиляции, например оптимизатора.

8. Костин Григорий, Швец Павел (10^Г). Реализация ролевой игры на базе скриптового движка.

Целью нашего проекта является реализация скриптового движка и создание ролевой игры на его базе.

Наша основная задача – реализация возможности изменения пользователем игрового мира без изменения кода программы. Эта возможность даёт пользователю свободу, позволяющую изменять всё, начиная рельефом карт и заканчивая созданием собственных юнитов и предметов, так же она упрощает создание сюжета и дополнений к игре.

Ещё одной задачей является создание проработанных игровых правил. Были придуманы специальные формулы для расчёта игровых ситуаций.

Другой нашей задачей было изучение операционной системы Windows и переход под неё. Частью этой задачи было изучение и использование GDI.

9. Кузнецов Иван (8^В). Алгоритм поворота объекта на C++ и другие геометрические преобразования тела.

Во многих современных игровых программах, и даже в самых простейших, написанных нами на C++, часто необходимо создание хорошей физической модели тела. Представьте себе столкновение двух героев игры. Они могут отскочить друг от друга, а потом начать вращаться вокруг своей оси (если они столкнутся краями, например). Или более распространённый случай: тело движется по кругу и при этом остаётся под одним и тем же углом к центру описываемой им окружности, а, значит, совершает поворот, относительно своей оси. Именно о повороте я хотел бы рассказать.

10. Кузьмин Алексей (10^Г) Реализация системы частиц для визуализации графических эффектов.

Система частиц – одна из важнейших из используемых в трёхмерной графике приспособлений, для создания визуальных эффектов типа огня, брызг, дыма. Она должна иметь большое количество настраиваемых параметров, чтобы обеспечивать свою многофункциональность.

В ходе выполнения пришлось выполнить следующий ряд задач:

Разобрать алгоритм моделирования системы частиц и ознакомиться с примерами (Рассмотреть основные классы, сделать их удобными в использовании и настройке).

Продумать и реализовать графическую независимость (На данный момент существует две основные графические библиотеки: DirectX и OpenGL – и огромное количество менее популярных. В мои задачи так же входила лёгкая переносимость системы с одной такой графической библиотеки на другую. Для этого в каждом классе содержится функция `render`, переписав которую в наследном классе можно обеспечить переход с одной графической системы в другую).

Рассмотреть частные случаи системы (поворачивающаяся система частиц, текстурированная система частиц).

В процессе работы над программой была изучена графическая библиотека DirectX.

11. Левкович-Маслюк Федор, Гречаник Сергей (11^Г). Моделирование газогидратов метана методом молекулярной динамики.

Целью работы было создание программной среды для молекулярного моделирования, позволяющей исследовать ряд вопросов, связанных с механизмом фазового перехода в газовых гидратах метана. Эти газогидраты важны для энергетики как возможный источник метана.

Кристаллы газогидратов состоят из додекаэдрических кластеров, образованных молекулами воды. Полости некоторых кластеров содержат молекулы метана. Стабильность кластеров поддерживается водородными связями. В условиях фазового перехода молекулы метана покидают полости кластеров. Согласно одной из теорий, в процессе перехода разрывается минимально возможное число связей. По изменению симметрии расположения молекул (возникновению т. н. алмазоподобной структуры пустого льда) можно установить, какие именно связи разрываются.

Для анализа этих предположений нами был разработан алгоритм, находящий оси вращательной симметрии произвольной системы точек в 3D. Этот алгоритм использовался в процессе расчета для отслеживания изменения симметрии молекулярной конфигурации.

Моделирование проводилось методами молекулярной динамики. Межмолекулярное взаимодействие описывалось двумя потенциалами: Леннарда-Джонса и электростатическим. Формирование водородных связей учитывалось заданием частичных зарядов на атомах кислорода и водорода (центрах взаимодействия). Архитектура системы позволяет легко добавлять новые потенциалы, численные алгоритмы.

Для управления моделированием была создана интегрированная среда, обеспечивающая визуализацию. Среда поддерживает настройку с использованием скриптового языка программирования, который имеет и самостоятельную ценность. Это императивный язык, основанный на принципах объектности и обобщённости. Он является экспериментальной разработкой с расчётом на выход за рамки проекта.

С помощью созданной системы был проведен расчет для ряда начальных конфигураций. В дальнейшем планируется усовершенствовать физическую модель, расширить возможности созданного языка.

12.Остроумов Олег (10^A). Редактор HTTP-запросов.

Цель – создание программы, имитирующей посетителя web-сайта, позволяющей редактировать HTTP-запрос и анализировать HTTP-ответ.

Общение броузера с web-сервером происходит по протоколу HTTP, описанному в RFC 2616. Запрос состоит из нескольких частей. В первой строке указан метод (броузер использует методы GET и POST) и адрес запрашиваемой страницы. Далее следуют заголовки, каждый заголовок представлен строкой вида Имя-заголовка: значение-заголовка. Далее может следовать тело запроса (если метод POST), в котором содержатся параметры.

Броузер позволяет контролировать адрес запрашиваемой страницы, а метод и заголовки HTTP-запроса броузер устанавливает самостоятельно. К тому же броузер не позволяет полностью контролировать поля форм и значения cookie.

Программа написана на языке Java (SDK 1.4.2), имеет графический интерфейс на основе библиотеки Swing. Интерфейс позволяет установить метод (GET или POST), адрес запрашиваемой страницы, набор и значения заголовков, параметров и cookie. Программа позволяет полностью проанализировать HTTP-ответ, включая значения заголовков. Реализованы в качестве дополнительной возможности автоматическая установка приходящих с сервера cookie, а также небольшой броузер для легкого заполнения форм и перехода по гиперссылкам. Возможна отправка нескольких запросов через заданный интервал времени.

Программа может использоваться для проверки серверного приложения на обработку некорректных запросов, в качестве вспомогательного средства для написания автоматизированных клиентов, а также для имитации посещения рекламных ссылок на web-сайтах. В дальнейшем будет улучшаться графиче-

ский интерфейс, создаваться планировщик запросов для большего удобства автоматизации.

13. Переславцев Алексей (10^B). Разработка генератора синтаксических LR1 анализаторов и алгоритмов восстановления после синтаксических ошибок LR1 разбора.

Целью работы является создание программы генерации синтаксического LR1 анализатора по заданной контекстно-свободной грамматике.

Разработка методов восстановления после ошибок является одной из самых сложных и интересных задач работы. Для восстановления были использованы следующие методы: подстановка предположительно не хватающих символов, удаление предположительно лишних символов, виртуальная свертка правил.

Другой задачей являлась разработка удобного интерфейса синтаксического модуля для взаимодействия с семантическим модулем. Этот интерфейс состоит из API библиотеки синтаксического анализатора и сгенерированного API по конкретной грамматике.

Ключевой задачей является разработка языка задания грамматики. За основу была взята классическая форма описания правил, но были сделаны некоторые особенности, связанные с возможностью более удобной для пользователя генерацией API по грамматике и поддержки методов восстановления после ошибок. Также были разработаны средства отладки логических ошибок при задании грамматики.

В настоящее время рабочая версия генератора используется для разработки языка формального описания условий математических и физических задач, а также входного языка формального описания функциональных интерфейсов робототехнических систем (Robot Description Language, RDL).

14. Рудой Георгий (9^B). Применение концепции MVC (Model-View-Controller) для портирования DOS -приложений под ОС Windows.

При переходе из-под «плоской» DOS-системы программирования в событийные системы, ярким представителем которой является Windows, неизбежно возникают сложности, связанные с тем, что весь код из-за отсутствия проработанной архитектуры "перетекает" в WndProc. Из-за этого программа становится непонятной, и ее сопровождение превращается в ад.

Концепция MVC, о которой пойдет речь в данном докладе, призвана устранить эти проблемы. Она разбивает программу на три (или более) основные составные части, максимально независимые друг от друга. После того, как произведен рефакторинг программы с ее использованием, она становится быстрой, понятной и легко сопровождаемой. Данная концепция также позволяет легко портировать и изменять существующую программу, сменив лишь ее составные части.

Данная концепция применима не только при программировании в событийных системах, но и в системах, основанных на вызовах callback-функций или даже «плоских» моделях.

В качестве примера была выбрана программа-тестировщик сортировок, так как сама по себе она представляет довольно легкую задачу и позволяет полностью сконцентрироваться на вопросах архитектуры.

15. Федоров Владимир (11^Г). Использование компонентной архитектуры для создания модульных приложений.

Цель работы – построение программного комплекса, построенного по компонентно-объектной модели.

В ходе создания и эксплуатации каких-либо больших программных комплексов, включающих в себя множество различных по смыслу и назначению частей (программных модулей), встает вопрос об обновлении отдельных частей или подключении новых без обновления остальных частей программы.

Для решения подобной задачи в современных операционных системах реализованы системы загрузки и работы динамически-подгружаемых или совместно-используемых библиотек (например, dll-файлы в системе Windows или so-файлы в Unix-подобных системах). Эти библиотеки способны экспортировать различные функции в главную программу.

Компонентная архитектура позволяет достичь независимости нескольких различных модулей программы и ее расширяемости, то есть возможности добавления дополнительных модулей без перекомпиляции всей программы.

Универсальность вышеописанных действий за счет особой структуры модулей, интерфейсов и внутрипрограммных систем менеджмента и поддержки компонентной структуры.

В разрабатываемом программном комплексе реализован ряд программных решений для обеспечения вышеописанных принципов. Создано описание общего универсального базового интерфейса IUnknown. Реализованы основополагающие структуры управления загруженными интерфейсами программы. Реализованы системы управления загрузки и использования модулей программы.

В докладе рассмотрены детали и механизмы реализации вышеописанных принципов и примера комплекса создаваемого по подобным принципам, а также более глобальной системы СОМ, универсализирующей данную архитектуру до масштабов целой рабочей станции или локальной сети.

16. Хохлова Марина (10^Г). Реализация программы построения сечений фигур.

Задача данного проекта состоит в том, чтобы построить сечение прямой для некоторой составной фигуры, расположенной на плоскости. Данная фигура должна состоять из участков разных цветов, которая может задаваться пользователем с помощью формулы, или являться набором кружков разных диамет-

ров и цветов, которые могут ставиться на плоскость пользователем с помощью мышки. Пользователь может пересечь данную фигуру прямой, поставив 2 точки, которые соединятся линией. Сечение будет представлять собой набор отрезков прямой различной длины и цветов.

Также одной из основных задач данного проекта была разработка описанного приложения под операционную систему Windows. Были реализованы такие аспекты программирования под Windows, как реализация межресурсного взаимодействия и сохранения графики в рабочем окне.

Возможные приложения данной задачи связаны с визуализацией характерного расположения атомов и молекул при компьютерном моделировании молекулярных систем, а также с изучением вида различных сечений живых организмов. Возможны приложения при изучении курса геометрии.

17. Чикунев Егор, Устинов Владислав (10^Г). Редактор карт для трёхмерной игры «Kayak Extreme».

Программа предназначена для редактирования карт игры «Kayak Extreme» с помощью трехмерного интерфейса. Пользователь выбирает один из участков карты и может изменять высоты точек на этом участке, задавать уровень воды, устанавливать метки начала и конца трассы.

При создании любого трехмерного мира встает проблема его внешнего вида. Каким-то образом необходимо хранить информацию об окружающей среде этого мира, об его поверхности. Все эти данные должны храниться в каком-то виде и обрабатываться программой. Необходима детализация ландшафтных объектов, с оптимальным количеством полигонов, отсечением невидимых объектов и разбиением игрового пространства на видимые зоны. Затем, когда мир создан, нужна программа, связывающая с ним пользователя. Для её создания достаточно простого использования Win32, однако чем она сложнее, тем более очевидным будет необходимость библиотеки надстройки над Win32.

Для выполнения поставленной задачи была создана библиотека, связывающая трехмерную графику и Win32 наиболее удобным способом. С помощью неё достигается независимость от используемой графической библиотеки – как бы ни была реализована графика, она будет также связана с тем же пользовательским представлением. Таким образом, при смене графической библиотеки, управление Win32 останется неизменным. Библиотека использует широкий набор разных представлений пользовательских оболочек, и это проявляется в их разнообразии на экране. Для графического представления гор была использована карта высот. О том, какой метод реализации трёхмерной графики лучше невозможно однозначно судить и это послужило основной причиной создания библиотеки, описанной выше. Для связи редактора с игрой был разработан собственный формат карт. Для выбранной мной реализации трёхмер-

ной графики была написана ещё одна библиотека, основанная на DirectX. Эти две библиотеки и составляют программу – графика и её связь с пользователем.

В докладе будут разобраны эти и другие вопросы, такие как: способы задания и генерации ландшафта, поверхностей, детализированная проработка, методы освещения, оптимизация, сглаживание, смешение текстур. Помимо этого будут разобраны некоторые неочевидные особенности библиотеки надстройки, а также представлена программа, иллюстрирующая возможности этой библиотеки – редактор карт. Будут также рассмотрены особенности сохранения результатов его работы.

18. Шарко Федор, Богданов Евгений (10^Г). Физическое моделирование.

Продемонстрировать такое сложное явление, как возникновение необратимости в механической системе.

Рассчитать во времени эволюцию системы нескольких тел (молекул), движущихся в замкнутом прямоугольном пространстве.

Реализовать возможность запуска всех частиц в обратном направлении, и рассчитать степень обратимости эволюции системы.

