для варианта без мелафена, y=18x+2,0 ($R^2=0,87$) для варианта с мелафеном 10^{-4} % и y=14x+12,66 ($R^2=0,67$) для варианта с мелафеном 10^{-8} %. Результаты сравнения выборок непараметрическим методом парного критерия Вилкоксона дафниевого теста по выживаемости и плодовитости показали наличие достоверного различия результатов опытов с внесением мелафена в концентрации 10^{-4} % от других вариантов эксперимента.

На основе проведенного исследования можно сделать следующие выводы:

- 1. Внесение мелафена в концентрации $1 \cdot 10^{-4} \%$ в донные отложения, загрязненные нефтью, приводит к снижению их остаточного содержания по сравнению с другими вариантами опыта.
- 2. Дисперсионный анализ результатов токсикологического эксперимента в разных вариантах опыта по критерию интенсивности деления с использованием инфузорий и по выживаемости и плодовитости в дафниевом тесте показал наличие достоверных отличий данных в варианте с внесением мелафена в концентрации 1·10-4% от контроля и от варианта с мелафеном в концентрации 1·10-8%.
- 3. На основе проведенного исследования можно рекомендовать использование мелафена в концентрации $1 \cdot 10^{-4}\%$ для стимуляции процессов само-очищения в условиях загрязнения донных отложений нефтепродуктами и уменьшения их токсического воздействия на гидробионтов.

SNAIL ENGINE: КРОССПЛАТФОРМЕННЫЙ МОЛУЛЬ ГРАФИЧЕСКОГО РАСШИРЕНИЯ

Рябинин К.В., Полотняншиков И.С.

Пермский государственный университет, Пермь, e-mail: icosaeder@ya.ru

На сегодняшний день очень многие компьютерные программы используют мультимедийные возможности вычислительных систем — отображение трёхмерной графики и воспроизведение звука. Для работы с графикой используются готовые библиотеки визуализации. Одним из самых популярных стандартов таких библиотек является *OpenGL*. Однако библиотек данного стандарта предоставляют лишь низкоуровневый доступ к графическому оборудованию, поэтому в сложных системах использовать их напрямую неудобно. Необходимо создавать прослойку между основной программой и библиотекой *OpenGL*. Такую прослойку принято называть модулем графического расширения.

В рамках описываемой работы был создан модуль графического расширения, который получил название Snail Engine. Он представляет собой объектно-ориентированную надстройку над библиотекой стандарта OpenGL. Так как трёхмерная сцена представляет собой набор объектов, принадлежащих различным классам, объектно-ориентированный подход наиболее адекватно подходит для организации модуля графического расширения.

Основное назначение модуля — облегчить труд программиста, разрабатывающего крупное мультимедийное приложение. Для этого $Snail\ Engine$ предоставляет возможность управления трёхмерной сценой на уровне геометрических примитивов более высокого уровня, чем OpenGL. Базовыми понятиями в $Snail\ Engine$ являются трёхмерные объекты с их свойствами, такими как материал и положение в пространстве, а также визуальные эффекты; в то время как в OpenGL — точки, многоугольники и матрицы преобразования. Кроме того, $Snail\ Engine$ имеет встроенные функции загрузки моделей и текстур из внешних файлов (поддерживаются различные популярные форматы: для трёхмерных моделей — 3ds и md2, а для двумерных текстур bmp, png и др).

Модуль графического расширения написан на языке C++ с соблюдением требований платформенной независимости и может быть собран под любую платформу, для которой реализован OpenGL. На данный момент модуль тестировался под управлением операционных систем GNU/Linux и Windows.

Чтобы повысить производительность при работе со сложными специальными эффектами (нестандартное освещение, системы частиц и т.п.) в состав Snail Engine включено средство использования микропрограмм – шейдеров – на языке GLSL. Шейдеры позволяют ускорить обсчёт трёхмерной сцены за счёт переноса части вычислений на графический процессор. Кроме того, они делают мультимедийное приложение более структурированным, так как эффекты, описанные шейдерами, находятся за пределами основного программного кода этого приложения. Шейдеры могут быть легко изменены и изменения вступят в силу без перекомпиляции основной программы.

С применением шейдеров в работе реализован ряд нестандартных моделей освещения: модели Орена-Наяра, Кука-Торренса и Миннарта (физически точно учитывают отражение от различных типов поверхностей – камня, металла и земли соответственно), а также модели митации подповерхностного рассеивания (моделирует частичную прозрачность материала) и неточечного источника света (источником является целый отрезок).

Для расширения мультимедийных возможностей модуля в него включены также средства воспроизведения звука, опирающиеся на библиотеку стандарта *OpenAL* и компрессор-декомпрессор *Ogg Vorbis*.

В Snail Engine реализована поддержка отображения трёхмерной сцены в стереорежиме при помощи шлема виртуальной реальности eMagin Z800. Таким образом, любое приложение, имеющее в основе данный модуль, будет совместимо с этим шлемом.

На базе *Śnail Engine* был создан виртуальный интерактивный исторический музей, «экспонаты» для которого были получены на трёхмерном сканере *Roland LPX-600*. Это приложение является показательным примером использования модуля для решения реальной прикладной задачи.

В дальнейшем планируется использовать Snail Engine для визуализации задач имитационного моделирования и искусственного интеллекта (когнитивных игр).

ПОДСИСТЕМА УПРАВЛЕНИЯ ОПЕРАТИВНОЙ ПАМЯТЬЮ

Рябцев Р.А

Воронежский институт высоких технологий, Воронеж, e-mail: kitaevakseniyavivt@yandex.ru

Оперативная память (ОП) – это важнейший ресурс любой вычислительной системы, поскольку без нее невозможно выполнение ни одной программы.

Самая большая трудность управления памятью заключается в том, что память не бесконечна и потому приходится постоянно учитывать возможность исчерпания свободной памяти. Практический опыт показывает, что управление оперативной памятью операционной системой недостаточно эффективно. Существует несколько основных проблем при использовании ОП. Главная проблема – фрагментация памяти – ситуация, когда неиспользуемое пространство достаточно велико, но настолько раздроблено, что найти свободный участок для размещения крупного объекта не представляется возможным.

Другой проблемой является то, что многие программы и сама ОС загружают в память множество библиотек. И не обязательно, что эти библиотеки всегда будут использоваться, занимая свободное место в оперативной памяти.

Еще одна проблема – утечка памяти. Приложениям выделяется определенное количество памяти. Но