

решении задач, при подготовке домашних заданий, т. е. осуществлять *самоконтроль*. Поэтому важно обеспечить единый подход и преемственность к формированию общих учебных умений при изучении различных предметов.

## РАЗРАБОТКА ГРАФИЧЕСКОГО РЕДАКТОРА СРЕДСТВАМИ OPENGL

*А. Ю. Ратников (УО «ГГУ им. Ф. Скорины»)*

*Научн. рук. В. О. Родченко,*

*ассистент*

В современном программировании часто возникают задачи, связанные с применением компьютерной графики. Графическая составляющая приложения может быть как основой системы (САД-системы, компьютерные игры), так и играть в ней второстепенную роль (системы обучения, системы управления технологическими процессами).

Библиотека OpenGL (Open Graphics Library – открытая графическая библиотека) представляет собой спецификацию, определяющую независимый от языка программирования кросс-платформенный программный интерфейс для написания приложений, использующих двумерную и трехмерную компьютерную графику. Она включает в себя более двухсот пятидесяти функций для рисования сложных трехмерных сцен из простых примитивов.

Разработанное в среде Delphi средствами OpenGL приложение представляет собой графический редактор, работающий с объектами трех типов: сфера, цилиндр и параллелепипед. Пользователь может произвольно размещать, трансформировать и поворачивать каждый объект. Переносы и повороты доступны и для всей системы объектов в целом.

Перед каждым изменением приложение запоминает свое состояние, что позволяет пользователю в любой момент отменить последнее действие. Присутствует возможность сохранения объектов в файл, с последующей их загрузкой. Также приложение обеспечивает возможность сохранения параметров вида. Это позволяет пользователю быстро восстанавливать нужный угол обзора, расстояние до системы объектов и ее положение в пространстве. Предусмотрен быстрый переход к виду сбоку или сверху путем нажатия на соответствующую ось.

Разработанный графический редактор позволяет создавать модели различной сложности путем их составления из базовых объектов. Также данное приложение демонстрирует использование открытой графической библиотеки при создании трехмерной графики и анимации, иллюстрирует ее возможности при создании интерактивных приложений моделирования, рассчитанных не только на визуализацию модели, но и на взаимодействие человека и компьютера в этом процессе. За счет приемов оптимизации была достигнута высокая производительность, что в совокупности с гибкими настройками обеспечивает комфортную работу пользователя с приложением.

## ШАБЛОННАЯ СИСТЕМА СОЗДАНИЯ ОБУЧАЮЩИХ ЗАДАНИЙ

*Ю. В. Решетько (УО «ГГУ им. Ф. Скорины»)*

*Научн. рук. М. С. Долинский,*

*канд. техн. наук, доцент*

Сайт дистанционного обучения [www.dl.gsu.by](http://www.dl.gsu.by) при работе со школьниками младших классов использует задания и обучающий материал, созданный с помощью анимационной среды Macromedia Flash. Одним из активно развивающихся обучающих курсов является настраиваемый комплекс заданий развивающего обучения. Он был разработан исходя из анализа аналогичных проектов других обучающих систем. Задания, созданные в этом

комплексе, имеют вид анимационных или художественных головоломок, которые побуждают детей активно использовать аппарат мыслительной деятельности. В частности анализировать, сравнивать, синтезировать, обобщать те или иные процессы, которые отражают задания. При этом элементарными операциями для выполнения заданий являются: перенос объектов экрана на свои позиции, поворот этих объектов, изменение цвета объекта, выбор верного ответа или запись ответа в текстовое поле, выделение определенной части экрана.

Данная система создана специально для сайта дистанционного обучения. Она состоит из двух частей: среда создания заданий (в дальнейшем конструктор) и интерпретатор заданий. В конструкторе можно добавлять стандартные элементы, а также создавать собственные. Для каждого элемента задания существует свой образ, который указывает верную позицию при выполнении задания. Элементы настраиваются с помощью специальных панелей конструктора. В итоге конструктор выдает текстовую информацию, которая отражает настройку задания. Данный текст копируется в специальный текстовый файл и отправляется на сайт дистанционного обучения. В момент выбора определенного задания на странице сайта запускается интерпретатор, который считывает загруженный текстовый файл и в соответствии с настройками создает вид задания.

Разработанный программный продукт позволяет в короткое время создавать комплексы развивающих заданий, тестовых заданий. Авторами заданий могут быть не только учителя, но и дети. Таким образом, данная система позволяет развивать творчество учащихся. Для подготовки авторов заданий на сайте дистанционного обучения размещена библиотека видео уроков по созданию разнообразных заданий с использованием данного программного продукта. Глобальным курсом развития системы создания заданий является внедрение в нее базы данных, которая упростит и ускорит процесс создания заданий.

## РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА МНОГОПРОЦЕССОРНОЙ СИСТЕМЫ

*В. В. Романенко (УО «ГГУ им. Ф. Скорины»)*

*Научн. рук. С. Ф. Маслович,  
ассистент*

Имитационная модель (ИМ) вычислительного процесса в МВС состоит из следующих элементов: задачи, поступающие в МВС для обработки и сама МВС. Задачи бывают пакетного и диалогового типов, причем диалог имеет больший приоритет.

Для исследования работы МВС необходимо описание характеристик РН: вектор параметров  $\{X_{РН}\}$  и характеристики структуры этой РН  $\{G_{РН}\}$ . Структура запросов каждого  $i$ -го типа задач (диалоговый и отложенный счет) является уникальной и определяется путем ранее проведенных экспериментов. Управляющими переменными моделирования являются: интенсивность поступления в узел МВС задач диалогового типа и отложенного счета  $\lambda_{\text{диал}}$  и  $\lambda_{\text{отл. счет}}$  соответственно, приоритеты  $\pi^1$  определяющие важность характеристик задач отложенного счета при отправке их на обработку на устройства (CPU, RAM); количество CPU <sub>$i$</sub>  в МВС. Задаваемыми параметрами моделирования являются: скорость CPU <sub>$i$</sub>  ( $v_{\text{CPU}}$ ) задаваемую размером кванта времени, выделяемого  $j$ -ой задаче РН; скорость выполнения запросов на RAM ( $v_{\text{RAM}}$ ) также определяемую размером кванта времени, выделяемого для  $j$ -ой задачи РН.

При реализации ИМ МВС воспользуемся библиотекой Enterprise Library системы AnyLogic. Разрабатываемый класс Transact является расширением стандартного класса Entity. Entity является базовым классом для всех заявок, которые создаются, работают с ресурсами и принимают участие в процессе. Заявка – все, что может являться объектом, для которого задан какой-то процесс. Для генерации транзактов воспользуемся активными