Литература

1 Рихтер, Д. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C# / Д. Рихтер. – Спб. : Питер, 2016. – 896 с.

2 Петцольд, Ч. Программирование для Microsoft Windows 8 / Ч. Петцольд. – Спб. : Питер, 2013. – 6-е изд. – 1008 с.

УДК 514 (004), 422.639(410)

А. Д. Голосова

СТРУКТУРА ХРАНЕНИЯ ИНФОРМАЦИИ О ТЕСТАХ ДЛЯ ОПРЕДЕЛЕНИЯ СКОРОСТИ ЗРИТЕЛЬНО-МОТОРНОЙ РЕАКЦИИ

В статье обсуждаются вопрос определения скорости зрительно-моторной реакции. В частности, структура хранения информации о тестах для определения реакции и результатов тестирования. Предложенная система позволяет более гибко проводить тестирование испытуемых и удобно обрабатывать результаты проведенных испытаний. Результаты работы могут быть использованы повсеместно для тестирования студентов, преподавателей и т. д. А также рассмотренные структуры хранения данных могут быть интерпретируемы для решения иных прикладных задач.

Способность реагировать на изменение светового потока является наиболее фундаментальным свойством зрительной системы, лежащим в основе всех остальных сторон ее деятельности.

Сложная зрительно-моторная реакция используется для оценки уровня сенсомоторных качеств. В рамках тестирования сложной зрительно-моторной реакции обследуемому последовательно предъявляются различающиеся по цвету световые раздражители, которые испытуемый должен максимально быстро «гасить» нажатием на соответствующую клавишу.

Чтобы оценить скорость сложной зрительно-моторной реакции, была написана программа, реализующая процесс тестирования, настройку тестов и приложение, генерирующее отчеты о проведенных тестах.

Настройки тестов передаются между средой настройки и средой тестирования в виде файла настроек, settings.ini.

Рассмотрим в таблице 1 структуру хранения информации о тестах.

Таблица 1 – Структура хранения информации

Имя параметра	Тип параметра	Назначение параметра
Name	AnsiString	Индивидуальное название для каждого теста
Field_Color	long	Цвет фона поля, на котором будут появляться точки
Point_Size	int	Начальный размер каждой точки, мм
Speed	int	Скорость роста точек, м/с
Interval_Min	int	Минимальный интервал между появлением точек, мс
Interval_Max	int	Максимальный интервал между появлением точек, мс
N	int	Количество точек
Point_Color	vector<int>	Цвет каждой точки
Point_Key	vector<AnsiString>	Клавиша для «погашения» точек

В программе информация, полученная из файла настроек, хранится с помощью структур данных. В разработанных приложениях используется структура Test:

```
structTest
{
AnsiString Name;
long Field_Color;
float Point_Size;
float Speed;
int Interval_Min;
int Interval_Max;
int N;
vector <int> Point_Color;
vector <AnsiString> Point_Key;
};
```

После объявляется вектор из элементов типа Test.

```
vector <Test> Tests;
```

При создании нового теста создается сперва новый элемент структуры Test, который потом заносится в вектор Tests. Таким образом, есть возможность создавать и хранить большое количество тестов.

Рассмотрим процесс удаления информации о тесте. Когда пользователь удаляет тест, необходимо удалить выбранный элемент из вектора Tests. для этого «сдвигаются» все последующие элементы вектора на одну позицию, как бы «закрывая» удаляемый элемент. А последний, лишний элемент удаляется из вектора методом pop_back.

После завершения работы приложения вектор Test удаляется, предварительно сохранив информацию в файл. Информация о настройках тестов хранится в Ini-файлах, которые имеют простую структуру.

Весь файл разбит на блоки: секции. В нашем случае это номера тестов. Название каждой секции имеет вид «Test_*i*», где *i* – номер теста.

Каждая секция хранит параметры теста. В Ini-файлах каждое значение хранится с указанием ключа – названия значения, например, Key = value.

Информация по каждому тесту условно разбита на две части: информация о точках (их цвет, клавиша, соответствующая каждой точке, начальный размер, скорость роста) и о поле (цвет, максимальный и минимальный интервалы между появлением точек, количество точек для данного теста и др.). Пример сохраненной информации в файле настроек:

```
[Test_3]
Name = Тест 3
Field_Color = 8454143
Point_Size = 13997
Speed = 10000
Interval_Max = 1000
Interval_Min = 304
N = 2
Point_Color1 = 16744576
Point_Color2 = 8388863
Key1 = Q
Key2 = W
```

Чтобы не допустить потерю или некорректное хранение информации, файл каждый раз полностью перезаписывается. В процессе записи текущей структуры в файл названия секций упорядочиваются, что позволяет в дальнейшем избежать ошибок при добавлении новых тестов.

Разработанная среда может быть установлена на компьютерах в школах, университетах, иных предприятиях для оценки восприятия у сотрудников, что поможет скорректировать рабочий график в случае сильной загрузке сотрудников при работе, на что будет указывать более низкая скорость зрительно-моторной реакции.

Литература

1 Лафоре, Р. Объектно-ориентированное программирование в С++ / Р. Лафоре. – СПб. : Питер, 2003. – 678 с.

2 Эккель, Б. Философия С++. Часть 1. Введение встандартныйС++ / Брюс Эккель, ЧакЭллисон. – СПб.: Вильямс, 2012. – 577 с.

3 Стефан, Р. С++ Для чайников / Стефан Р. – СПб. : Вильямс, 2011. – 336 с.

УДК 004.4

Д. Н. Голубев

ТЕСТИРОВАНИЕ ЗНАНИЙ СТУДЕНТОВ – НЕОТЪЕМЛЕМАЯ ЧАСТЬ УЧЕБНОГО ПРОЦЕССА

Статья посвящена описанию приложения для тестирования студентов 1 курса специальностей «Программное обеспечение информационных технологий» и «Информатика и технологии программирования» по языку программирования Assembler. Приложение написано в среде Builder 6.0 на языке С++, база данных Microsoft Access, которая подключается к приложению с помощью технологии ADO. Приложение позволяет оценить знания студентов как по отдельным темам курса, так и по всему курсу целиком.

В настоящее время компьютерные технологии стали неотъемлемой частью учебного процесса, значительно повышающие его эффективность. Проведение компьютерного контроля знаний студентов является основой получения объективной независимой оценки уровня учебных достижений (знаний, умений и практических навыков) студентов.

Разработано приложение, позволяющее оценить уровень знаний студентов по языку программирования Assembler. Приложение написано в среде Builder 6.0 на языке С++ [1–2], база данных Microsoft Access, которая подключается к приложению с помощью технологии ADO.

Разработанное приложение позволяет: оперативно вносить изменения в программное обеспечение; проверять знания с занесением информации в базу данных; доступ преподавателя к итогам работы одного студента или всей группы. Приложение предназначено для проверки знаний студентов как по отдельным темам курса, так и по всему курсу целиком.

В приложении реализована возможность различного уровня доступа к данным: уровень администратора и пользователя. Администратор может просматривать правильные ответы, редактировать базу данных, а так же просматривать номеров вопросов. Для пользователей, работающих с приложением, предусмотрена система изменяющихся подсказок.