

## **ИЗ ОПЫТА РАБОТЫ**

**М.С. Долинский**, к.т.н., доцент кафедры математических проблем управления Гомельского государственного университета им. Ф. Скорины,

**Ю.В. Решетько**, аспирант Гомельского государственного университета им. Ф. Скорины,

**М.А. Долинская**, магистрант Гомельского государственного университета им. Ф. Скорины,

**Н.С. Лебедько**, учитель информатики Шерстинской БШ

## **Конструктор интерактивных флеш-заданий как открытая система для создания электронных учебных пособий**

### **Введение**

В серии публикаций [1-11] описана авторская система развивающего обучения программированию, ориентированная на подготовку школьников к олимпиадам по информатике от школьной до международной. Важной составляющей этой системы являются задания, которые выполнены на базе флеш-технологий. В работах [9-11] были описаны возможности специально разработанного «Конструктора» флеш-заданий для развивающего обучения. Использование заданий, созданных таким образом, для

фронтального обучения младших школьников СШ № 27 г. Гомеля и факультативного обучения на базе сайта <http://dl.gsu.by> (далее DL) школьников разных возрастов г. Гомеля и Гомельской области, показало высокую эффективность такого подхода. Более того, создание и использование заданий в «Конструкторе», с одной стороны, стимулировало его развитие, а с другой наглядно показало, что его возможности стали значительно шире, чем задумывалось изначально. В результате непрерывной апробации и развития «Конструктор» вырос в полноценную открытую среду для создания современных интерактивных электронных средств обучения. Данная статья посвящена описанию «Конструктора», как открытой системы для создания интерактивных электронных учебных пособий. Важно отметить, что «Конструктор» позволяет быстро делать интерактивные флеш-задания для произвольных предметных областей. Одновременно «Конструктор» позволяет накапливать наработки в некоторой предметной области, обеспечивая их многократное повторное использование.

### **Приемы работы обучаемого**

Для удобства дальнейшего обсуждения напомним приемы работы обучаемого, которые поддерживает «Конструктор». На рис. 1-10 представлены задания, иллюстрирующие возможности общения пользователя со средой обучения. Заметим, что сами эти задания объединены в пакет заданий, который называется «Приемы работы» и этот пакет используется нами для знакомства новичков со всеми возможными способами выполнения предлагаемых заданий. Содержательно данные задания чрезвычайно просты. Это сделано намеренно, чтобы трудности с пониманием задания не мешали освоению приемов работы. Обращаем внимание на то, что в левом верхнем углу фрейма задания находится иконка, подсказывающая, каким способом нужно выполнять это задание.

На рис. 1 показано задание, для выполнения которого необходимо перетащить с помощью мышки рисунки на нужные позиции. На рис. 2 фигурки можно не только перетаскивать, но и поворачивать с помощью клавиш «Стрелка влево» и «Стрелка вправо».

Рис. 3-5 иллюстрируют три вида заданий, которые выполняются кликами мыши по нужной области. На рис. 3 отметка появляется после каждого правильного клика. На рис. 4 кликнуть нужно на картинке, и соответствующая область отметится и на картинке поиска, и в таблице задания. Заметим, что клик по рисунку в таблице задания не приводит к выполнению задания. На рис. 5 требуется пос-



Рис. 1. Перенос рисунка в прямоугольную область



Рис. 2. Перенос (разрешены повороты) рисунка в прямоугольную область

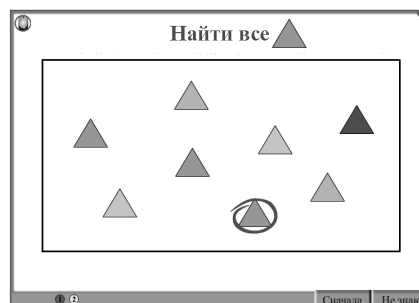


Рис. 3. Указать (выделить) области рисунка



Рис. 4. Связанные области выделения

ледовательно кликнуть на всех правильных областях, только после этого появятся отметки на всех них.

На рис. 6 показано задание, в котором для получения правильного результата нужно кликами мыши «перелистывать» возможные варианты ответов в поисках правильного. Задание считается выполненным, если во всех перечисляемых полях выбраны правильные ответы.

Задание на рис. 7 выполняется раскрашиванием некоторых областей с помощью стандартного «обмакивания курсора-кисточки» в нужную краску.

Рис. 8 и рис. 9 иллюстрируют способы выполнения задания с помощью соединения нескольких точек (последовательными кликами мыши в этих точках).

Задание на рис. 10 иллюстрирует ввод с клавиатуры.



Рис. 5. Скрытые связанные области выделения



Рис. 6. Перечисляемые поля



Рис. 7. Раскрашивание



Рис. 8. Точки соединения

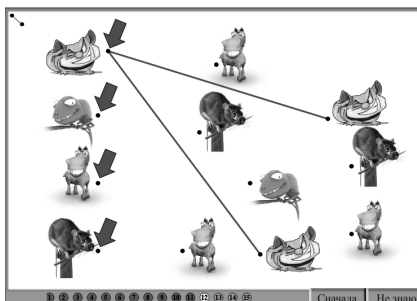


Рис. 9. Множественные точки соединения



Рис. 10. Ввод с клавиатуры

## Принципы конструирования одиночных заданий

Одиночное задание конструируется из произвольных рисунков (готовых или создаваемых непосредственно в конструкторе), надписей, и специальных активных элементов конструктора, таких как: таблицы, линии, перечисляемые поля, поля ввода, таймер, жизни. Результатом конструирования задания является текстовый файл в формате XML, который полностью описывает задание, все его компоненты и их настройки, сопровождаемый файлами использованных внешних рисунков (в форматах jpeg, png). При выполнении этого задания, запускается специальная программа «Плейер», которая считывает соответствующий XML-файл и использованные рисунки и предъявляет задание ученику. В каждом задании автоматически появляется кнопка «Сначала», позволяющая ученику, запутавшемуся при выполнении сложного задания, вернуться в исходное состояние. Этой кнопкой может пользоваться также и учитель – если он (или другой ученик) помогли кому-то сделать задание – для принудительного повторного выполнения задания. Кроме того, в целях усложнения выполнения задания, автор задания может включить в задание специальную кнопку «Проверить». При наличии этой кнопки правильность выполнения задания проверяется только в тот момент, ког-

да ученик нажмет эту кнопку. Если кнопка «Проверить» отсутствует, правильность выполнения задания проверяется автоматически непрерывно.

### Принципы конструирования пакетов заданий

Задания могут объединяться конструктором в пакеты. Пакет заданий может быть контролирующим или обучающим.

В случае контролирующего пакета задания имеют линейную структуру, выдаются последовательно. Переход от одного задания к другому происходит в одном из двух случаев: задание выполнено правильно или истекло время, отведенное автором на выполнение этого задания, в этом случае оно считается невыполненным (выполненным неправильно). Цель контролирующего пакета – оценить уровень подготовленности ученика в какой-то теме. Результат – количество правильно выполненных заданий из определенного количества предложенных заданий.

Обучающий пакет заданий имеет древовидную структуру (см. рис. 11). Задания следующего уровня вложенности являются обучающими к текущему заданию. Если ученик правильно выполнил задание, то все обучающие

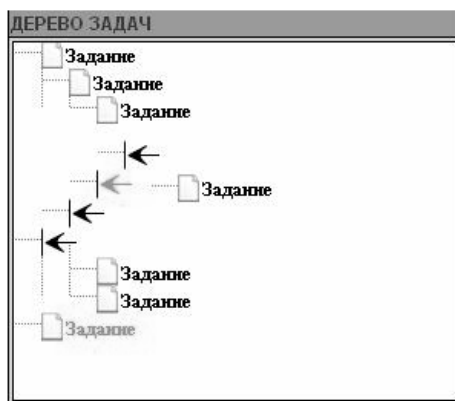


Рис. 11. Древовидная структура заданий

задания пропускаются, иначе ему выдается первое подводящее задание, в случае правильности его выполнения – второе подводящее. После правильного выполнения последнего подводящего задания ученику предъявляется исходное неправильно выполненное задание. Кроме того, ученику даются две кнопки ручной навигации по заданиям: «Не знаю» и «Я понял». В каждом подводящем задании автор может поместить свои подводящие и т.д.

### **Интеграционные механизмы «Конструктора»**

Важным качеством «Конструктора» является его открытость к развитию, в том числе и сторонними разработчиками. Такая открытость проявляется на многих уровнях взаимодействия конструктора с авторами заданий, учениками и внешней компьютерной средой.

*Уровень заданий.* Для конструирования заданий можно использовать произвольные рисунки, созданные в сторонних графических редакторах. Разнообразие приемов выполнения заданий обеспечивает поддержку практически всех распространенных заданий, как в компьютерном мире, так и в соответствующей литературе. Имеется также возможность загружать ранее созданные задания для быстрого создания на их основе аналогичных новых заданий. Поддерживаются базы заданий; внешних рисунков; рисунков, созданных в конструкторе.

*XML-представление.* Поскольку «по построению» задание – это XML-файл (возможно, сопровождаемый файлами использованных рисунков), то обеспечивается использование сторонних средств для генерации XML-файлов с заданиями и тем самым, автоматического создания заданий в некоторой предметной области.

*Модули.* Альтернативным механизмом стороннего развития «Конструктора» являются модули. Имеется строго определенный интерфейс взаимодействия основного задания с вызываемым им swf-файлом (модулем). Это позволяет модульное и независимое развитие «Кон-

структора» без перегрузки его ядра узкоспециальными применениями.

На рис. 12-14 показаны примеры заданий, созданных на базе трех модулей.

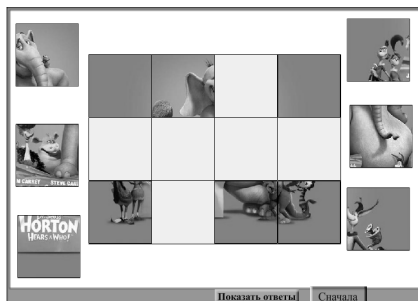


Рис. 12. Пример использования модуля «Пазл»

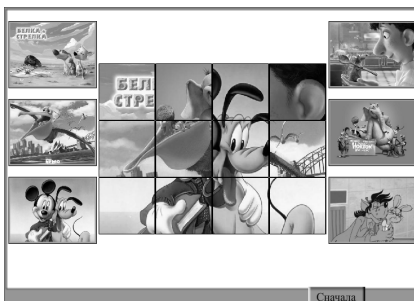


Рис. 13. Пример использования модуля «Кубики»

Для выполнения задания, представленного на рис. 12, необходимо внести в соответствующие области фрагменты рисунка. Ученики могут нажать кнопку «Показать ответы», чтобы увидеть полностью картинку, которую необходимо собрать.

При создании данного задания с помощью модуля «Пазл», автор должен загрузить рисунок; указать, на сколько строк и столбцов нужно разрезать рисунок и разместить получившиеся фрагменты рисунка по своему усмотрению. Например, можно поставить некоторые фрагменты на свои места, а можно не ставить ни одного, можно растащить фрагменты рисунков по границе поля, а можно сложить все фрагменты в стопочку и т.д.

Для создания этого задания автор загружает до 6 рисунков и указывает количество строк и столбцов, на которые их нужно разрезать. При выполнении задания ученик, кликая по квадратику, перебирает все картинки в этой позиции в поисках нужной. Автор может указать, вариант выполнения задания: подсказки размещать перед глазами или на отдельной странице, а также подсказки «исчезают»



после сбора соответствующей картинки, подсказки «появляются» после сбора соответствующей картинки, подсказки статичны.

На рис. 14 представлен пример использования модуля «Аналогия». Поле задания разделено на левую и правую половины. В левой половине представлен пример выполнения задания. В правой нужно выполнить задание по аналогии, перетаскивая картинки с верхней части в правильные позиции нижней части. Кроме того, в некоторых полях (x, y, z) нужно выбрать правильный ответ получившегося примера.

5	2					1	4				
1	3					3	2				
5	+	2	=	7		+		=	x		
7	-	1	=	6	x	-		=	y		
6	+	3	=	9	y	+		=	z		

Рис. 14. Пример использования модуля «Аналогия»

*Интерфейс с вызывающей средой.* Задания, выполненные в «Конструкторе», могут выполняться автономно. В то же время, обеспечен интерфейс взаимодействия с вызывающей средой. В частности, с использованием этого интерфейса, осуществлена интеграция заданий (и пакетов заданий), разработанных в «Конструкторе», с инструментальной системой дистанционного обучения «Дистанционное обучение в Беларуси» (<http://dl.gsu.by>). Это позволяет обеспечить непрерывный удаленный доступ к заданиям посредством Internet-технологий, а также обеспечить накопление статистики по всем обучаемым, с ее предъявлением ученикам, учителям, родителям и широкой общественности. Кроме того, задания могут быть интегрированы с теорией по предмету, создаваемой с помощью произвольных средств современных интернет-технологий.

## Заключение

В данной работе «Конструктор» интерактивных флеш-заданий, представлен как открытая мощная среда для раз-

работки интерактивных электронных учебных пособий. Приведены аргументы открытости «Конструктора». Использование флеш-технологий делает процесс выполнения заданий простым и привлекательным по форме, оставляя изначальную содержательность заданий. Наличие автоматической проверки правильности выполнения заданий и интеграция с системой дистанционного обучения DL.GSU.BY обеспечивают возможность эффективного фронтального использования созданных заданий в учебном процессе.

### Литература

1. Долинский, М.С. Об опыте подготовки школьников Гомельской области к республиканским и международным олимпиадам по информатике / М.С. Долинский // Информатизация образования. – 2009. – № 1(54). – С. 29-40.

2. Долинский, М.С. Система интернет-курсов дифференцированного обучения программированию школьников и студентов / М.С. Долинский, М.А. Кугейко // Информатизация образования. – 2010. – № 1(58). – С. 58-68.

3. Долинский, М.С. Как учить думать школьников и студентов? / М.С. Долинский, М.А. Кугейко // Информатизация образования. – 2010. – № 2(59). – С. 62-72.

4. Долинский, М.С. Технология развивающего дифференцированного обучения программированию младших школьников «с чистого листа» / М.С. Долинский, М.А. Кугейко // Информатизация образования. – 2010. – № 3(60). – С. 12-20.

5. Долинский, М.С. Интернет-курс «Базовое программирование» как средство подготовки к областным олимпиадам по информатике / М.С. Долинский, М.А. Кугейко // Информатизация образования. – 2010. – № 4(61). – С. 3-15.

6. Долинский, М.С. Развитие мышления младших школьников на основе флеш-заданий на рисование, раскраску и конструирование в системе DL.GSU.BY / М.С. Долинский, Ю.В. Решетько, М.А. Кугейко // Информатизация образования. – 2011. – № 1(62). – С. 24-35.

7. Долинский, М.С. Какими должны быть задачи на олимпиадах по информатике / М.С. Долинский, М.А. Кугейко // Информатизация образования. – 2011. – № 1(62). – С. 68-76.

8. Долинский, М.С. Флеш-шаблоны для создания заданий развивающего обучения / М.С. Долинский, Ю.В. Решетько, М.А. Долинская // Информатизация образования. – 2011. – № 2(63). – С. 14-28.

9. Долинский, М.С. Конструирование интерактивных флеш-заданий на развитие мышления / М.С. Долинский, Ю.В. Решетько, М.А. Долинская // Информатизация образования. – 2011. – № 3(64). – С. 21-33.

10. Долинский, М.С. Конструирование интерактивных флеш-заданий на развитие мышления на базе произвольных картинок / М.С. Долинский, Ю.В. Решетько, М.А. Долинская // Информатизация образования. – 2011. – № 4(65). – С. 3-14.

11. Долинский, М.С. Конструирование интерактивных флеш-заданий на базе собственных танов / М.С. Долинский, Ю.В. Решетько, Н.С. Лебедько // Информатизация образования. – 2012. – № 1(66). – С. 24-34.

*Статья поступила 24.04.2012*

