

minor parties – 11,76 %; 2004 – ALP – 37,69 %, The Coalition – 46,7 %, minor parties – 10,39 %; 1993 – ALP – 44,92 %, The Coalition – 44,27% , minor parties – 5,61 %; 1983 – ALP – 49,48 %, The Coalition – 43,57 %, minor parties – 5,03 %. Between major parties two-party-preferred vote (TPP) is held. The TPP is an indicator of how much swing is required to change the result, so as the swing is very small and it can be proved by the following data: 2010 – ALP – 50,12 %, the Coalition – 49,88 %, swing – 0,24 % 2004 – ALP – 47,26 %, the Coalition – 52,74 %, swing – 5, 48 %; 1993 – ALP – 51,44 %, the Coalition – 48,56 % , swing – 2,88 % ; 1983 – ALP – 53,23 %, the Coalition – 46,77 %, swing – 6,46 %.

All the facts are in favour of the belief that in spite of a great variety of parties in the country, the power is concentrated at hands of one or two main parties, which are formed through the centuries and satisfy the demands of the vast majority of people, living in this country.

#### **References**

1. The Encyclopedia Americana: international addition complete in thirty volumes; Volume 2. / Danbury, Connecticut, 2003. P. 730–733.
2. The World Book Encyclopedia: Volume 1. – World Book International (World Book Inc.) / Sydney, Australia, 2001. P. 813–824.
3. Australian Politics. Political Parties [Internet Resource] / D & B Mann Webmasters. - Mode of access: <<http://www.aussieeducator.org.au/reference/government/australianpolitics.html>>.
4. Political Parties in Australia [Internet Resource] /Educational Services Australia. Mode of access: <[http://www.parliament.curriculum.edu.au/pol\\_parties.htm](http://www.parliament.curriculum.edu.au/pol_parties.htm)>.

## **РАЗРАБОТКА ИГРОВОГО ДВИЖКА С ИСПОЛЬЗОВАНИЕМ СРЕДСТВ HTML, CSS И JAVASCRIPT**

**А. С. Федоров**

### **ВВЕДЕНИЕ**

Цель данной работы – изучить и продемонстрировать возможности данных технологий в применении к сфере компьютерных игр на примере собственной разработки. Почему были выбраны именно эти средства? Сейчас вышеозначенные технологии получили большое развитие, и уже обладают функциональностью достаточной для реализации разнообразных проектов. Мое веб-приложение как раз показывает, что можно реализовать, основываясь лишь на стандартной функциональности.

## СТРУКТУРА

То, что видит пользователь в браузере – HTML страница. Каждый элемент приложения представлен отдельным тегом DIV, с примененными к нему свойствами CSS. На начальном этапе страница содержит лишь DIV-элементы контейнеры, внутрь которых затем вкладываются динамически генерируемые объекты игры. Атрибут ID у тега DIV позволяет однозначно идентифицировать объекты игры.

В основном скрипте Logic.js проверяется нынешнее значение переменной состояния приложения: Меню, Игра, Выигрыш или Поражение. В зависимости от него, в html файл записываются DIV элементы с необходимыми ID, по которым, в дальнейшем и определяется их дальнейшее поведение.

Также там содержатся функции, использующиеся в игровом цикле. Игровой цикл в данном случае представляет собой функцию, вызывающуюся через короткие промежутки времени, здесь – каждые 30 миллисекунд. Через эту функцию производятся все действия, которые необходимо выполнить за этот кадр. Таким образом, мы получаем приложением, выдающее около 33-х кадров в секунду.

Также этот скрипт отвечает за отслеживание перемещения мыши и вычисления относительного перемещения курсора.

С точки зрения кода, использовался объектно-ориентированный подход к программированию. В этом плане каждый элемент является объектом, содержащим методы для самостоятельной обработки своих состояний. На данный момент представлены такие объекты как Муха – Fly.js, Полоски – Strip.js, Фон – Jam.js. А также вспомогательные – OverlapTester.js и Indicator.js.

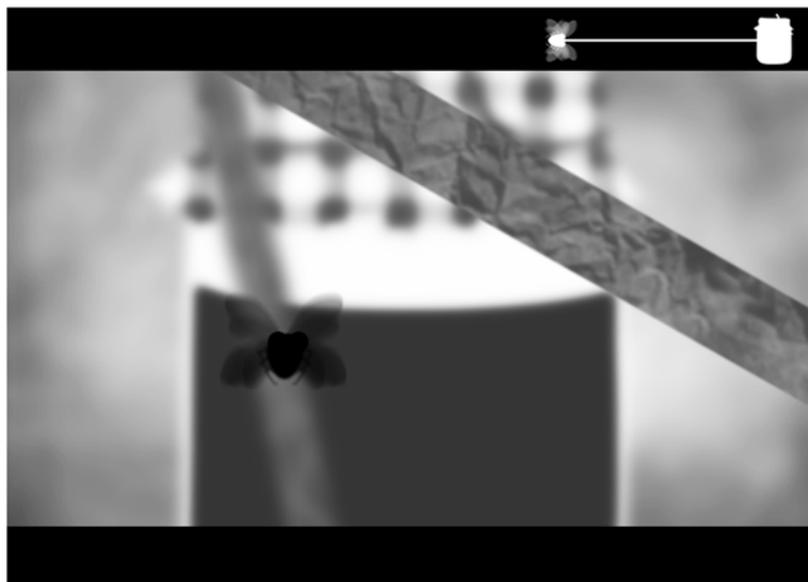


Рис. 1. Скриншот игрового процесса

## ОПРЕДЕЛЕНИЕ СТОЛКНОВЕНИЙ

Для обозначения физических границ каждого объекта используются так называемые Bounding Box, то есть, по сути, каждый объект просто ограничен прямоугольником с нужным наклоном, для упрощения подсчета столкновений, которое сводится к определению пересечения прямоугольников.

Для этого был введен класс Vector2.js для представления точек и класс Rectangle.js для представления самих прямоугольников. Класс OverlapTester.js содержит методы для определения вхождения точки в прямоугольник, определения пересечения прямых и пересечения прямоугольников.

Хоть игра и представляет собой псевдо трехмерную, определение столкновений просчитывается в двумерном пространстве. Каждый элемент имеет свой z-index, определяющий координату объекта в глубину. Положение игрока-Мухи было принято за точку отсчета и имеет нулевую глубину, координату  $Z = 0$ . Двигается не игрок, а окружение вокруг него. Поэтому, исходя из специфики игры, в расчете столкновений можно брать лишь объекты, также имеющие z-index 0. То есть в каждый момент мы имеем просто плоскость в координате  $Z=0$ , и прямоугольники объектов, лежащие на ней в данный момент.

## CSS

Эффект трехмерности был легко достигнут благодаря новым свойствам, введенным в CSS3.

Свойство Perspective позволяет задать значение перспективы в рамках одного элемента-контейнера, translate3D и rotate3D позволяют управлять положением объекта в трехмерном пространстве.

Свойство Blur используется для размытия изображения. Что применялось для создания эффекта Focus Blur. В зависимости от удаления от координаты  $Z=0$ , предполагаемой точки фокусировки камеры, изменялось значение размытия объектов.

Свойство Opacity отвечает за прозрачность объекта. Оно использовалось для устранения резкого появления элементов, а также для их плавного выцветания, чтобы предотвратить их наезд на камеру и закрытие обзора.

На CSS основано создание эффекта взмаха крыльев у мухи. Это реализовывалось последовательным переключением свойства видимости элементов с изображениями крыльев в разных состояниях.

Для реализации окраски индикатора использовались дублирующие объекты с изменяющейся прозрачностью.

Также CSS используется для быстрого создания анимации наведения курсора на кнопки. В CSS3 теперь большое количество свойств можно анимировать, достаточно указать время анимации и имя анимируемого свойства, в данном случае – это размер шрифта и его стиль.

## **КРОССБРАУЗЕРНОСТЬ**

CSS3 на данный момент не является согласованным и общепринятым стандартом. Различные браузеры по-разному интерпретируют новые свойства, а некоторые свойства для работы в конкретном браузере требуют конкретного префикса в записи. Также различается общая производительность приложения.

Данная конкретная игра корректно работает пока только в Chrome.

## **ЗАКЛЮЧЕНИЕ**

В заключении мы видим, что обладая лишь стандартным набором инструментов, с помощью html, css и javascript можно создать продукт достаточного для рынка уровня.

Эти технологии сейчас лишь набирают популярность. Они вытесняют технологию flash, более того, ведется работа по их активной интеграции с мобильными устройствами. Уже сейчас подобные приложения достаточно легко переносятся на такие платформы как iOS, Android, WindowsPhone.

Являясь скриптовым языком высокого уровня, javascript легко изучаем. Он не типизирован, достаточно гибок. HTML и CSS также легко поддаются освоению. Это инструменты, позволяющие больше сосредоточиться на самой логике игры, на игровом процессе, не задумываясь о низкоуровневом взаимодействии с аппаратной частью.

Отсюда, конечно, вытекает главный недостаток этих методов – невысокая производительность. В этом плане стоит отметить нововведения HTML5 – Canvas и встроенная библиотека WebGL, позволяющие оптимизировать работу с графикой.

В итоге мы имеем отличный кроссплатформенный инструмент, который уже на данный момент позволяет достаточно легко реализовывать разнообразные задачи. При этом, активно развивающийся и имеющий большие перспективы, на который возлагаются серьезные надежды по объединению многих платформ.

## **Литература**

1. JavaScript для всех / Алексей Георгиевич Федоров – Москва; КомпьютерПресс, 1998.

2. Большая книга CSS / Дэвид Макфарланд – СПб; Питер, 2011.
3. HTML5. Для профессионалов / Хуан Диего Гоше – СПб; Питер, 2013.

## **ОСОБЕННОСТИ ПЕРЕВОДА АНГЛОЯЗЫЧНЫХ СЛОГАНОВ И РЕКЛАМНЫХ ТЕКСТОВ НА РУССКИЙ ЯЗЫК**

**Л. В. Фляменг**

Реклама вездесуща, многолика, разнообразна. Она тесно связана со средствами массовой информации, которые выступают в качестве носителя конкретных рекламных текстов. Вместе с тем не следует забывать, что реклама – не только массовая, но и во многом принудительная коммуникация [1, с. 72].

В настоящее время в нашей стране возрос интерес к рекламе в связи с развивающимися международными отношениями в сфере торговли и туризма. По этой причине предоставляется актуальным не только углубленное изучение рекламных текстов, написанных на одном из наиболее распространенных международных языков – английском, но и изучения проблем, возникающих при переводе англоязычных рекламных слоганов и текстов, и поиск путей их разрешения. Кроме того, подобные исследования важны и для развития рекламной науки, и для совершенствования рекламного образования.

Перевод рекламного текста, в отличие от перевода художественной литературы, в которой переводчик обязан передать художественно-эстетические достоинства оригинала, несколько отличается по форме, языковым средствам, а также ярко выраженной коммуникативной направленностью. В процессе перевода таких текстов, переводчику приходится решать как чисто языковые проблемы, обусловленные различиями в семантической структуре и особенностями использования двух языков в процессе коммуникации, так и проблемы социолингвистической адаптации текста.

Однозначно ответить какой стратегии перевода лучше придерживаться при переводе рекламных текстов тяжело, но нужно отметить, что буквальный или дословный перевод применяется крайне редко, поскольку в рекламе важна не форма текста, а образность.

Представив, что слоган для рекламы шоколадного батончика Bounty «A taste of paradise» был бы переведен дословно – «Вкус рая» или «Райский вкус», заметим, что эти варианты не таким яркие как «Bounty – райское наслаждение». Тут переводчики удачно подобрали эмоциональное и выразительное слово «наслаждение», вместо общеупотребительного «вкус». Такие слова-усилители – неотъемлемая часть успешной рекламы.