

## ♦ Глава 2.3. Разработка адаптивного сайта по начертательной геометрии

Курс «Начертательная геометрия» требует от студента напряженной и продолжительной самостоятельной работы, состоящей в чтении лекционных материалов, выполнении индивидуальных практических заданий, изучении учебников и решении множества задач. Последнее в большой степени реализуется созданным редактором для отрисовки условия и построений при решении позиционных и метрических задач [32].

Изучение теоретического материала в настоящее время нуждается в поддержке специального учебного web-ресурса – студенты активно пользуются всемирной паутиной, применяя для этого разнообразные технические средства, а не только персональный компьютер. Существующий Интернет-ресурс для поддержки изучения курса «Начертательная геометрия» не является адаптивным – он не просматривается на мобильных устройствах: телефонах, смартфонах, планшетах и пр. Опыт разработки различных web-ресурсов подсказывает, что создать единственную версию сайта, работающую одинаково хорошо на всех устройствах, скорее всего, не удастся. Поэтому следует ограничить набор устройств для отображения, на которых будет адаптироваться разрабатываемый ресурс.

**Выбор устройств отображения сайта.** Функционал существующего Интернет-ресурса не ограничивается возможностью просмотра статических web-страниц. Важнейшей его особенностью является наличие Flash-редактора для построения условий и пошагового решения задач. Типовые примеры, представляющие собой неинтерактивные Flash-ролики, демонстрирующие важнейшие методы решения, также составляют важную компоненту сайта: наряду с редактором этот раздел чаще всего посещается студентами. Неинтерактивные *Flash-ролики включают компоненты, которые не предназначены для масштабирования* – их наглядность обеспечивается продуманностью размеров и соотношений элементов изображения. Специфическими требованиями к ресурсам по инженерной графике и начертательной геометрии являются абсолютная точность расположения изображений, подписей их элементов, соотношение символов и пр., что весьма непросто сочетается с требованием адаптивности. Несмотря на программно реализованную масштабируемость редактора, полагаем, что минимальный размер экрана мобильного устройства, для которого следует адаптировать сайт, составляет 4 дюйма. Такой размер экрана

♦ Развитие образовательной среды средствами информационных технологий

имеют, главным образом, смартфоны и некоторые мобильные телефоны. Дальнейшее уменьшение окна редактора может привести к плохой читаемости изображения вследствие слишком мелкого кегля текста и недопустимого «наползания» подписей на построения, что делает просмотр решения бессмысленным. Кроме того, по данным некоторых web-ресурсов, посвященных анализу покупательского спроса на мобильные устройства [1], четырехдюймовый размер экрана является одним из самых актуальных (см. табл. 10 и рис. 50).

Таблица 10

**Размеры экрана, разрешения мобильных устройств и популярность у пользователей [1]**

Разрешение \ Размер	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Всего
Small	8.2%						8.2%
Normal	0.1%	14.2%		33.0%	20.1%	10.6%	78.0%
Large	0.9%	4.5%	1.7%	0.6%	0.7%		8.4%
Xlarge	0.1%	4.7%		0.4%	0.2%		5.4%
Всего	9.3%	23.4%	1.7%	34.0%	21.0%	10.6%	



Рис. 50. Диаграммы популярности мобильных устройств у пользователей по а) размеру экрана, по б) разрешению экрана [1]

Следующими по распространенности являются размеры экрана в 7 и 10 дюймов. Такой размер экрана имеют планшеты, которые очень популярны в студенческой среде. Разрешение экрана самых распространенных у пользователей мобильных устройств (см. рис. 51) составляет 200-300dpi [7]. Это обстоятельство потребует повторной разработки практически всего набора растровых изображений для сайта.

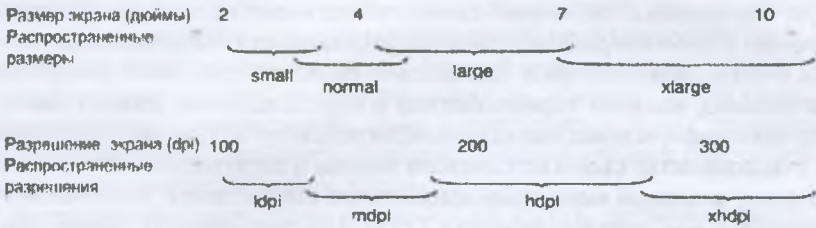


Рис. 51. Наиболее распространенные размеры экрана и их разрешения [7]

Таким образом, для адаптации нами выбран смартфон с размером экрана 4 дюйма, планшеты с размером экрана 7 и 10 дюймов, работающие под операционной системой Google Android. Среда разработки Flash Builder, созданная на базе Eclipse, содержит возможность конвертации ресурса для платформ Apple iOS, и BlackBerry Tablet OS. Не вызывает сомнений, однако, что все элементы технологически неоднородного Интернет-ресурса, безусловно, потребуют собственного набора атрибутов и, возможно, иного программного решения под каждую операционную систему, разрешение и размер экрана.

Необходимость разработки нового дизайна для мобильной версии сайта не нуждается в пояснениях. Следует, однако, заметить, что появление адаптивной версии требует изменения дизайна полной версии сайта (сайта для персонального компьютера), чтобы обеспечить преемственность, узнаваемость. Интернет-ресурса при работе с любой из его версий – на любой платформе и любом устройстве.

Рассмотрим информационную архитектуру разработанного для персонального компьютера web-сайта и проанализируем ее на предмет применимости для создаваемого адаптивного ресурса.

**Информационная архитектура сайта.** Информационная архитектура разработанного сайта заслуживает обсуждения в части систем *организации, предметизации и навигации*. Ресурс строится на основании *точной схемы организации*. Точная схема организации применима, когда информация разделена на корректно определенные и непересекающиеся части [121]. В web-пространстве широко распространены точная алфавитная, точная хронологическая или точная схема организации по географическим признакам. Недостатком точной схемы организации Интернет-ресурсов является то обстоятельство, что пользователь должен знать правильное наименование искомой информации. Для образовательных ресурсов, посвященных изучению фундаментальных

наук, построенных по точной схеме, это не является проблемой ввиду хорошей структурированности и разработанности терминологии точных и естественнонаучных дисциплин. Пользователи таких ресурсов, как правило, владеют терминологией и определенно не рассчитывают получить информацию, наугад блуждая по сайту. К тому же такие сайты в большинстве своем не слишком велики и загружены контентом, к которому в полной мере применимы такие определения, как «неоднозначность» или «гетерогенность» [121]. Неоднозначность минимизируется благодаря развитой системе классификации в предметной области. Некоторые темы, например «Определение натуральной величины отрезка прямой», могут быть отнесены как к разделу «Позиционные задачи», так и к разделу «Проекции точек и прямых». Однако, согласно традиционному порядку проведения курса, тема, а, следовательно, и страница сайта размещена в последнем разделе. Такая неоднозначность устраняется однократным проходом по разделам меню и проблемы не составляет. Гетерогенность представляемой информации ограничивается усилиями разработчиков сайта, хотя и не везде удается избежать размещения одних и тех же (или подобных) текстов, графики и видео в разных форматах, поскольку такова сущность мультимедийного ресурса. Создавать и сопровождать точные схемы организации достаточно просто ввиду того, что категоризация элементов ресурса для специалиста-разработчика не составляет никакого труда: информация размещается один раз, формировать и, следовательно, разрешать проблему перекрестных ссылок не требуется.

*Предметизация* и именование разделов сайта не вызвала практически никаких трудностей – образовательный ресурс по начертательной геометрии посещается почти исключительно заинтересованными пользователями: студентами и преподавателями. Разработанность терминологии предопределяет известность предмета и области *поиска* и практически исключает какую бы то ни было неоднозначность даже для студента, начинающего изучать курс «Начертательной геометрии».

Некоторые проблемы верстки возникали из-за длинных названий (с количеством символов свыше 70). В таком случае название пункта меню приходилось сокращать, а если это было невозможно – изменять формат элемента и верстку страницы.

Системы *поиска*, *тезаурус* и *управляемый словарь* для данного ресурса, на наш взгляд, не актуальны. *Поиск* не планируется организовывать, исходя из тех соображений, что преподавателю и знающему студенту он не нужен, а начинающему изучать курс полезно глубже

ознакомиться с его структурой, чему способствует многократный просмотр меню, разделов, анимационных роликов, иллюстраций и прочих элементов сайта в поисках нужной информации. Тезаурусы и управляемые словари требуются, главным образом, когда списки эквивалентных терминов или списки предпочтительных терминов должны облегчить создание и ведение большого объема слабоструктурированного гетерогенного контента.

Интегрированность ActionScript 3.0 во FLEX, который является расширением функциональности Flash-плейера и включает инструменты для быстрого создания Rich Internet Applications, позволила создать сайт [33, С. 143–183], содержащий следующий функционал:

- теоретические материалы;
- типовые примеры;
- задачи;
- справку.

Перечисленные задачи реализованы в главном меню на главной странице и составляют *глобальную навигацию* по сайту (рис. 52). «Теоретические материалы» позволяют просматривать html-страницы по следующим разделам курса: «Проекция точек и прямых», «Позиционные задачи», «Метрические задачи», что отражено в подменю этого пункта – меню глобальной навигации. Возможно расширение этого раздела наиболее важными темами курса.

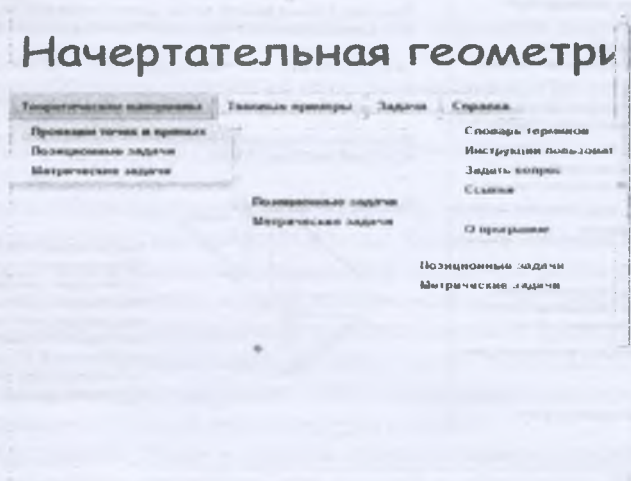


Рис. 52. Глобальная навигация по сайту «Начертательная геометрия»

◆ Развитие образовательной среды средствами информационных технологий

Локальная навигация раздела «Проекция точек и прямых» представлена левосторонним меню и включает:

- построение проекций точки на плоскости проекций H и V;
- точку в системе трех плоскостей проекций H, V и W;
- проекции отрезка прямой линии общего положения. Следы прямой;
- частные положения отрезка прямой линии относительно H, V, W;
- точку и прямую;
- определение натуральной величины отрезка прямой и углов ее наклона к H и V;
- относительное положение двух прямых;
- проекции прямого угла.

При выборе пункта меню загружается страница с html-тегами и специальными тегами-инструкциями, читаемая ActionScript 3.0 (рис. 53).

## Начертательная

Проекция точек и прямых

Построение проекций точки на плоскости проекций H и V

Точка в системе трех плоскостей проекций H, V и W

Проекция отрезка прямой линии общего положения. Следы прямой

Частные положения отрезка прямой линии относительно H, V и W

Точка и прямая

Определение натуральной величины отрезка прямой и углов ее наклона к H и V

Относительное положение двух прямых

Проекция прямого угла

Теоретические материалы | Типовые примеры | За

Проекция отрезка прямой AB, не параллельной плоскости проекции, искажения зависят от угла наклона прямой к плоскости проекции

$$A'B' = AB \cos \alpha_1, \quad A''B'' = AB \cos \alpha_2, \quad A'''B''' = AB \cos \alpha_3,$$

где  $\alpha_1, \alpha_2, \alpha_3$  – углы наклона прямой к плоскостям проекции

Возможны случаи, когда прямая равно наклонена к трем плоскостям проекции этой прямой к осям проекций равны  $45^\circ$  (см. рис. 4)



При выполнении машинного черчения к соответствующим размерам (безосновным чертежам) чертёж локально на рисунке показан на рисунке с помощью вспомогательных линий связи В'В'' строится третьей проекции прямой AB'

Рис. 4

Рис. 53. Локальная навигация раздела «Проекция точек и прямых» и элемент контекстной навигации – гиперссылка на странице

Раздел «Позиционные задачи» содержит следующие пункты локальной навигации, организованные в виде левостороннего меню:

- принадлежность точки прямой. Конкурирующие точки;
- задание плоскости. Прямые частного и общего положения в плоскости. Следы плоскости:
  - относительное положение прямой и плоскости;
  - относительное положение двух плоскостей;
  - поверхности. Задание на чертеже – определитель, очерк, каркас;
  - линейчатые и кривые поверхности. Точка и линия, принадлежащие поверхности;
  - относительное положение прямой и поверхности, плоскости и поверхности;
  - относительное положение двух поверхностей. Общие и частные случаи пересечения;
  - применение вспомогательных секущих плоскостей и сфер в качестве посредников.

Левостороннее меню раздела «Метрические задачи» содержит следующие пункты локальной навигации:

- преобразование чертежа способом замены плоскостей проекций;
- преобразование чертежа способом плоскопараллельного перемещения;
- преобразование чертежа способом вращения вокруг проецирующей оси;
- преобразование чертежа способом вращения вокруг горизонтали или фронтали;
- преобразование чертежа способом вращения вокруг следов плоскости.

Раздел «Типовые примеры» содержит неинтерактивные ролики, иллюстрирующие важнейшие методы решения позиционных и метрических задач. Большинство из перечисленных роликов находится в состоянии разработки или доработки вследствие изменения структуры flа-файла для придания функции пошаговой прокрутки ролика.

Подменю «Позиционные задачи» меню глобальной навигации «Типовые примеры» включает следующие неинтерактивные ролики, организованные в виде левостороннего меню:

- построение точки и прямой, принадлежащей плоскости;
- относительное положение прямой и плоскости. Перпендикулярность;

#### ◆ Развитие образовательной среды средствами информационных технологий

- относительное положение прямой и плоскости. Пересечение;
- относительное положение прямой и плоскости. Параллельность;
- относительное положение прямой и плоскости. Принадлежность;
- относительное положение двух плоскостей. Перпендикулярность;
- относительное положение двух плоскостей. Пересечение;
- относительное положение двух плоскостей. Параллельность;
- поверхности. Задание на чертеже. Построение точек поверхности;
- точка и линия, принадлежащие кривым поверхностям;
- пересечение плоскости с поверхностью. Построение касательной плоскости;
- пересечение прямой общего положения с поверхностью общего положения;
- пересечение двух поверхностей вращения. Теорема Монжа;
- пересечение соосных поверхностей;
- пересечение поверхностей. Метод эксцентрических сфер;
- пересечение поверхностей. Метод концентрических сфер;
- пересечение поверхностей. Метод секущих плоскостей.

Подменю «Метрические задачи» раздела «Типовые примеры» содержит следующие swf-ролики, названия которых также составляют левостороннее меню:

- определение натуральной величины отрезка прямой и углов ее наклона к  $H$  и  $V$ ;
- определение расстояний от точки до прямой;
- определение расстояний от точки до плоскости;
- определение расстояний от точки до поверхности;
- определение расстояний между двумя параллельными прямыми;
- определение расстояний между двумя скрещивающимися прямыми;
- определение натуральной величины угла наклона плоскости к плоскостям проекций  $H$  и  $V$ ;
- определение натуральной величины двугранного угла;
- определение угла наклона прямой к плоскости проекций;
- определение натуральной величины угла между пересекающимися прямыми;
- площади плоских фигур.



В частности, рис. 54 иллюстрирует результат работы Flash-ролика о построении линий пересечения поверхностей с помощью метода секущих плоскостей.

Неинтерактивные ролики для выбранных размеров экрана в 4, 7 и 10 дюймов предстоит переделать, начиная с этапа сценария, отчасти пожертвовав наглядностью и подробностью отображения решаемой задачи. Низкое разрешение устройств воспроизведения диктует необходимость создавать векторные объекты точно по пиксельной сетке, не использовать графические объекты повышенной сложности, а также лишние промежуточные кадры и эффекты прозрачности [5].

## Начертательная геометрия

Теоретические материалы

Типовые примеры

Задачи

Справка

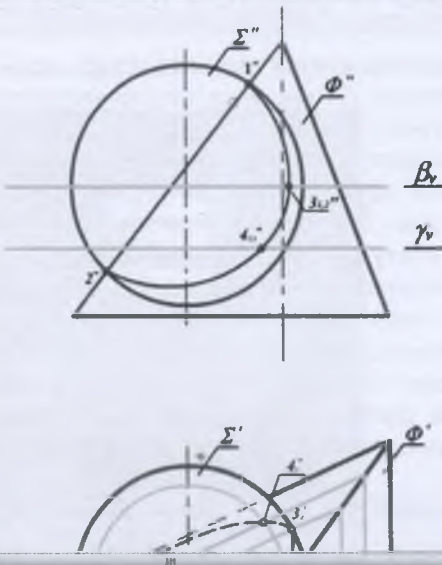


Рис. 54. Раздел «Типовые примеры»  
с загруженным неинтерактивным роликом

#### ♦ Развитие образовательной среды средствами информационных технологий

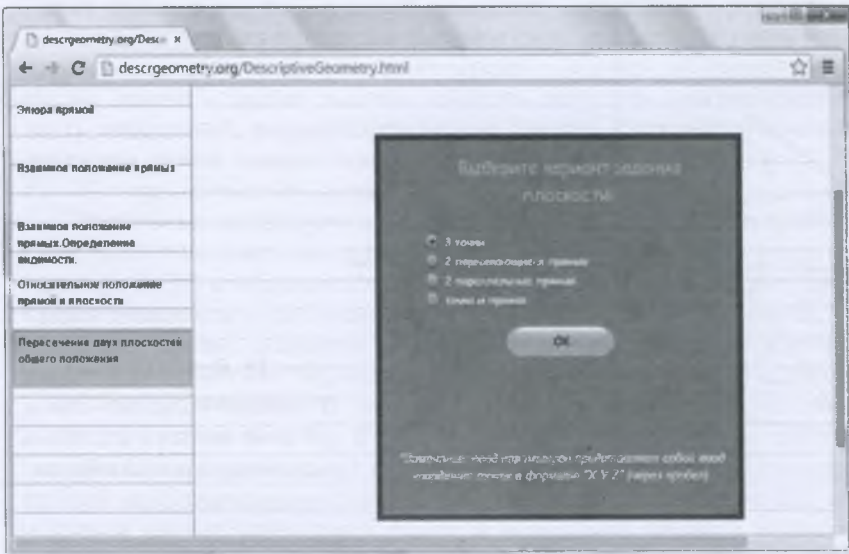
Раздел «Задачи» включает интерактивные ролики, которые позволяют [32]:

- задать условие в окне диалога (рис. 55);
- визуализировать построение условия (рис. 56);
- визуализировать пошаговое решение сформированной задачи (рис. 57);
- осуществить откат на произвольное количество шагов алгоритма решения;
- обеспечить диалог с пользователем и дополнительные построения в окне редактора (рис. 58, 59, 60).

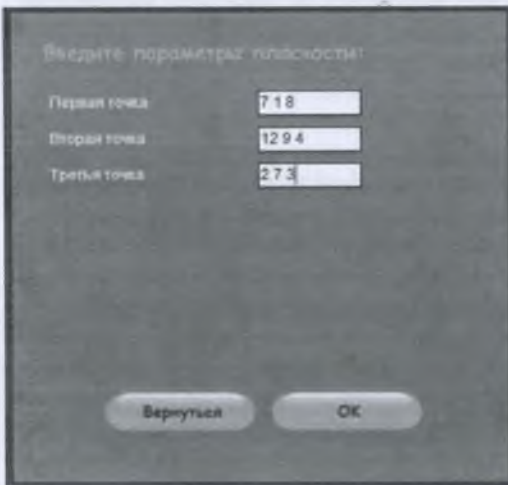
На рис. 55 представлены окна диалога для ввода условия задачи о пересечении двух плоскостей общего положения. Вначале выбирается один из четырех способов задания плоскостей на чертеже. В зависимости от выбора в этом диалоговом окне (рис. 55 а) программа открывает для пользователя окно диалога (рис. 55 б) для ввода координат точек плоскости. В программе реализована возможность изменить выбор, сделанный на предыдущем этапе ввода данных по клавише «Вернуться».

По завершении ввода открывается окно редактора задач, и пользователь, нажимая на стрелку «Вперед» на панели управления справа (рис. 57), пошагово строит условие.

Рисунок 56 иллюстрирует последовательность отрисовки условия задачи о пересечении двух плоскостей общего положения по данным, введенным на рис. 55. Возможность построить все условие (как и все решение) сразу программно реализуется несравненно проще. Однако принудительная реализация всех шагов построения практически избавляет нашу обучающую систему от недостатка, присущего множеству компьютерных обучающе-контролирующих приложений, когда студент изучает не столько конкретную задачу, метод и предмет в целом, сколько приобретает навыки работы с интерфейсом обучающей программы – «нажать нужные кнопки, чтобы получить результат». •

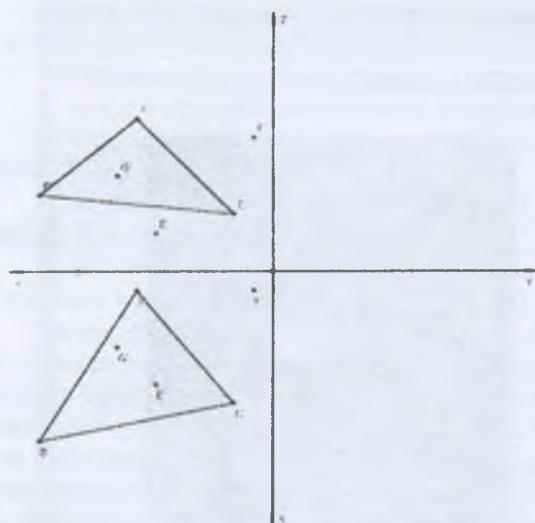


*а) выбор способа задания плоскости*



*б) ввод координат точек первой плоскости*

*Рис. 55. Диалоговые окна задания условия задачи о пересечении двух плоскостей общего положения*



*Рис. 56. Визуализация пошагового построения условия задачи о пересечении плоскостей общего положения*

Стрелка «Назад» (рис. 57) отменяет предыдущий шаг построения, и обучающийся при желании может повторить последовательность построения, что повышает вероятность усвоения материала. Фактически, если с системой работает думающий студент стационара или заочной формы обучения, редактор задач выполняет только техническую работу – отрисовку условия и решения, чем существенно экономит его время, освобождая от рутинных операций. Очень помогает описываемый инструмент и среднему студенту, позволяя воспроизвести каждый шаг решения в любом направлении с выбранной скоростью и максимальной степенью подробности. Никак не помогает редактор построения задач только тому студенту, который не понимает, что происходит на экране. Впрочем, особенности эксплуатации сайта разными категориями учащихся – это тема для отдельного обсуждения.

Один из шагов решения задачи о пересечении двух плоскостей общего положения по введенным ранее данным (рис. 55) демонстрирует рис. 57. Проекция последней из построенных точек выделяется красным цветом. Окно редактора задач, кроме описанных панелей и меню сайта, имеет область построений и панель управления. В области построений выполняется пошаговая программная отрисовка условия и решения задачи, а также реализуются графические диало-

говые процедуры: пользователь указывает элементы чертежа и выполняет построения в ответ на запросы системы. Панель управления, кроме описанных выше кнопок «вперед» и «назад», включает кнопку смены задачи «Другая задача», область запросов программы и область сообщений, выделенную белым цветом. Результат решения – проекции линий пересечения плоскостей общего положения  $PP'P'$  и  $PP''P''$  – представлены на рис. 58.

Диалоговые процедуры в процессе решения этой задачи требуются не только на этапе ввода данных. На панели управления редактора задач (рис. 58) отображается диалог выбора варианта построения секущей плоскости – по имеющимся линиям или по горизонтали. Область построений становится доступной для выбора существующих линий или отрисовки горизонтали указанием двух точек с помощью левой клавиши мыши. В данном случае для получения результата выбраны прямые  $G''E''$  и  $E''F''$ . Указанная пользователем линия (и нарисованная горизонталь) выделяется красным цветом. Процедура выбора завершается по клавише ОК. Если выбор пользователя корректен и удачен (отрисовка решения в видимой области редактора возможна), построения пошагово продолжают до получения результата. Однако пользователь может допускать ошибки, выбирать неудачные варианты решения и нерационально задавать объекты в области построений.

Обработка ошибок диалоговых процедур – важная часть любой обучающей системы. На рис. 59 подсвеченная красным цветом прямая  $B''C''$  отражает некорректный выбор пользователя – в области сообщений панели управления отображается реакция системы: «Выделено меньше двух линий». В области сообщений выводятся не только сведения об ошибках, но и числовые данные, указания о необходимости повторного выбора или отрисовки – все необходимые реакции на корректные и некорректные действия пользователя. Рисунок 60 представляет результат неудачного использования опции «по имеющимся линиям» в диалоговой процедуре выбора варианта построения секущей плоскости – указания прямых  $A''B''$  и  $B''C''$ . В результате линия пересечения получена далеко вне области видимости, доступной редактору, о чем и сигнализирует область сообщений: «Некоторые точки пересечения лежат вне отображаемой области». В данном случае, все четыре точки, по которым строятся проекции линий пересечения, несмотря на свойство масштабируемости, присущее всем объектам области построений, оказались вне отображаемой области.

♦ Развитие образовательной среды средствами информационных технологий

Пользовательский интерфейс позволяет перемещаться вперед и назад по шагам решения и перейти к другой задаче и практически не вызывает трудностей при работе. Исключение составляют, пожалуй, только задачи, включающие диалог и дополнительные построения. Перемещение на предыдущий шаг решения представляет собой только просмотр выполненной диалоговой процедуры (как и формальной процедуры), а не возможность выполнить ее заново, то есть изменить ход решения. Для этого придется вернуться на этап ввода исходных данных и выполнить диалоговую процедуру по-новому. Эта и другие особенности интерфейса пользователя отражены в окне «Инструкции пользователю» меню глобальной навигации «Справка».

Контекстная навигация на любом сайте развивается по мере его наполнения. На рис. 53 элементом контекстной навигации страницы является гиперссылка «линия связи», по клику на которую открывается определение этого понятия из «Словаря терминов», который находится в меню глобальной навигации «Справка» (см. рис. 52). Словарь терминов (гlossарий) является важной составляющей контекстной навигации любого образовательного ресурса. На рис. 61 представлено окно гlossария с определением понятия «линия связи».

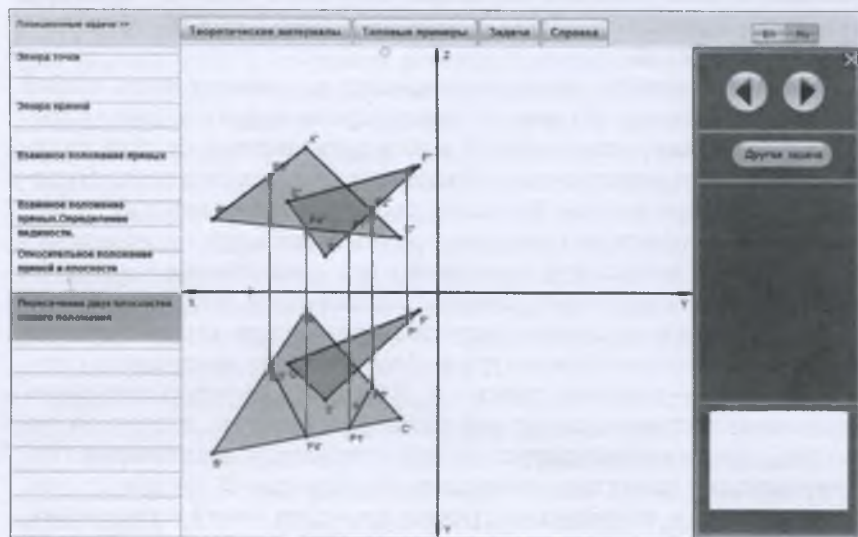


Рис. 57. Пошаговое решение задачи о пересечении двух плоскостей. Промежуточный результат

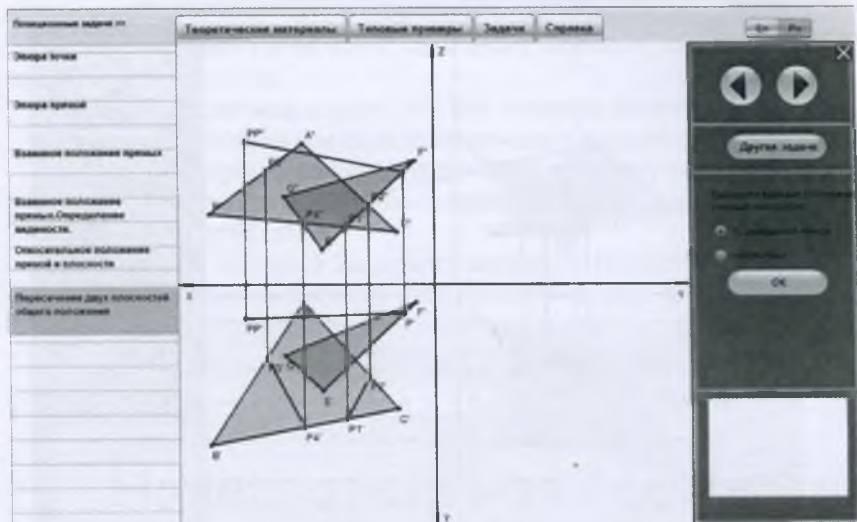


Рис. 58. Результат решения задачи о пересечении двух плоскостей в окне редактора на сайте

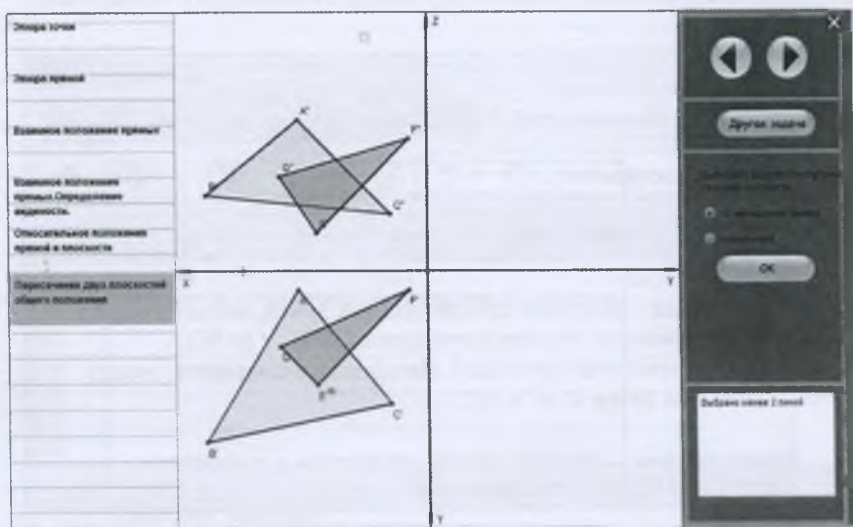


Рис. 59. Указание вспомогательной линии на экране и сообщение об ошибочном выборе пользователя

◆ Развитие образовательной среды средствами информационных технологий

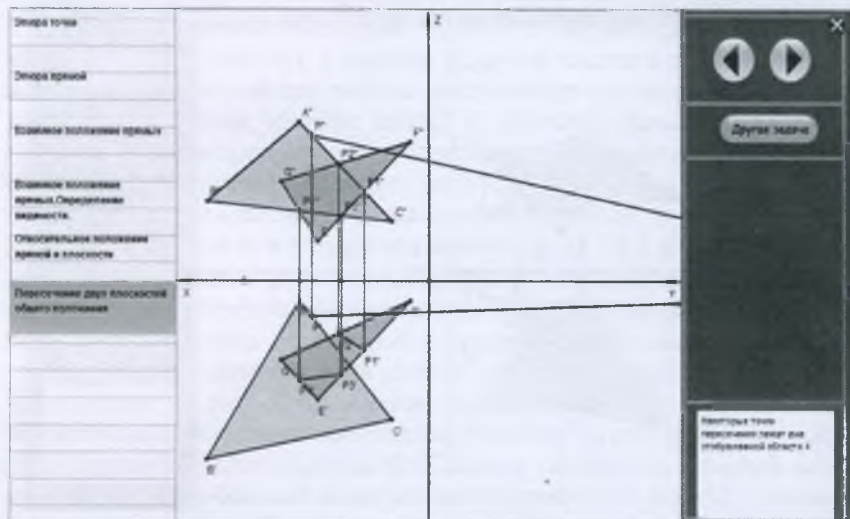


Рис. 60. Решение задачи о пересечении двух плоскостей при неудачном выборе вспомогательных линий и окно диалога редактора с сообщением

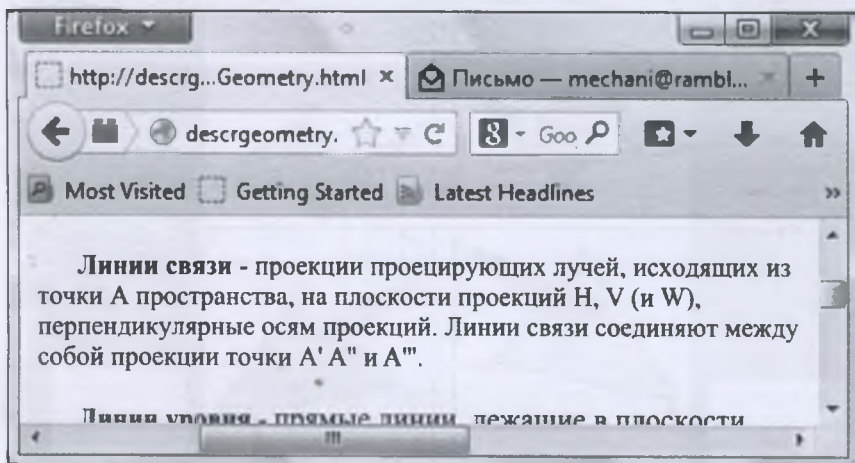


Рис. 61. Окно глоссария, открывающееся по клику на элемент контекстной навигации страницы – гиперссылку «линия связи»



Меню «Справка» глобальной навигации, кроме «Словаря терминов», включает пункты:

- «Инструкции пользователю», где описаны возможности сайта и особенности его пользовательского интерфейса;
- «Задать вопрос», который открывает страницу с формой обратной связи (рис. 62), позволяющей отправить вопрос на почту разработчикам;
- «Ссылки», который содержит перечень Интернет-ресурсов, посвященных начертательной геометрии (рис. 63);

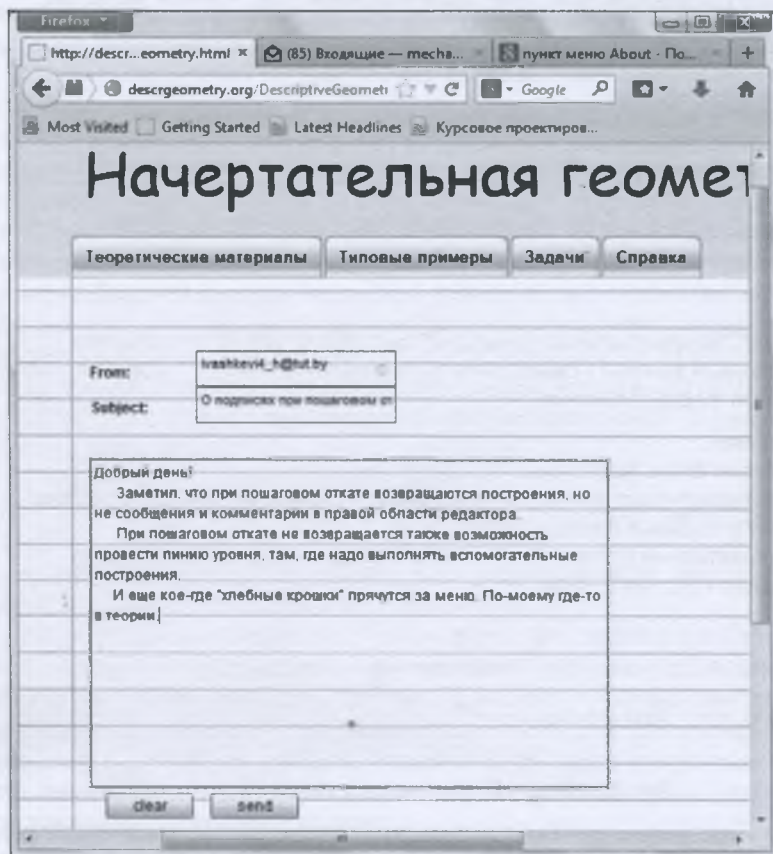


Рис. 62. Форма обратной связи, позволяющая задать вопрос разработчику ресурса

◆ Развитие образовательной среды средствами информационных технологий

- «О программе», который сообщает о возможностях программы, разработчиках, версии программы (рис. 64).

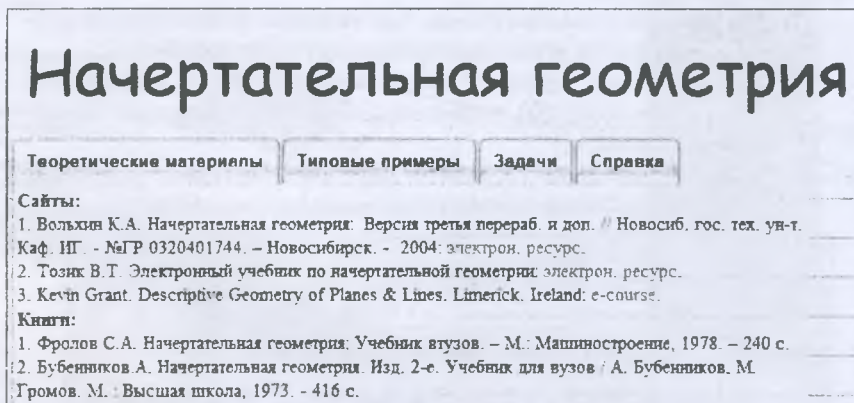


Рис. 63. Окно «Ссылки», содержащее список сайтов и литературы по начертательной геометрии

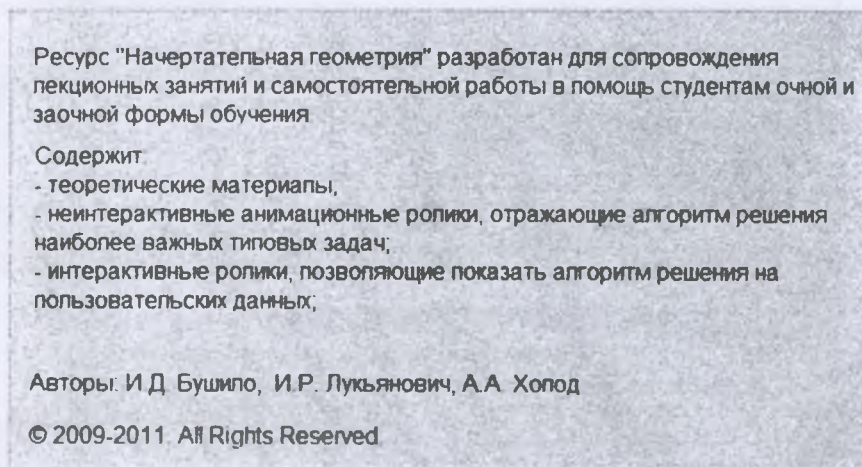


Рис. 64. Окно «О программе» меню «Справка»

По мере развития Интернет-ресурса скорее всего будет реализовываться вспомогательные системы навигации в виде карты сайта и указателя.

Информационный дизайн полной версии сайта и интерфейс пользователя, на наш взгляд, не содержат избыточных элементов – все вышеописанное представляет полезный функционал сайта и требует перенесения на мобильную платформу.

**Создание адаптивного дизайна.** Важнейшей задачей, реализуемой в настоящее время авторами ресурса, является достижение адаптивности сайта, контент которого содержит большое количество Flash-анимации.

В настоящее время эффективность web-сайта зависит от качественного выполнения его важнейших составляющих: *дизайна, навигации, контента, кроссбраузерности и кроссплатформенности.*

Минималистичный *дизайн* и простая навигация создаваемого интернет-ресурса, казалось бы, не представляют проблемы для переносимости. Однако даже применение фоновых рисунков, прозрачностей, градиентов оказалось нежелательным вследствие низкого разрешения и небольших размеров экранов мобильных устройств. Создавать несколько комплектов графики и «подгонять» путем итерационного приближения к наиболее рациональному решению для выбранной в качестве базовой группы устройств только ради сохранения некоторых внешних признаков нашего ресурса мы сочли неразумным. Было принято решение разработать дизайн для мобильных устройств и в соответствии с этим изменить дизайн полной версии сайта. Вообще существует несколько подходов к созданию адаптивной верстки Интернет-ресурсов:

- сначала мобильные (*mobile first*);
- верстка полной версии с адаптацией под мобильные устройства;
- «итерационное сближение» уже разработанных сайтов с бесконечным внесением правок и улучшений.

Первые два подхода вполне реализуемы для вновь разрабатываемых сайтов. Третий подход неизбежен для ресурсов, подлежащих рефакторингу. Несмотря на то, что нами для работы был выбран первый из подходов, в полной мере избежать сложностей, сопутствующих третьему подходу, не удалось. Рисунки 65–67 дают представление о том, как выглядят страницы сайта на эмуляторе смартфона Nexus S со следующими характеристиками:

- платформа – Android;
- размер экрана – 4’’;
- разрешение экрана – 480x800px;

♦ Развитие образовательной среды средствами информационных технологий

– плотность экрана (пикселей на дюйм, профиль для стандартов Android) – hdpі (см. таблицу 10 и рис. 51).

Как правило, для мобильного приложения дизайнер на выходе предоставляет MDPI- и HDPI-наборы графики. Нам для работы понадобятся LDPI-, MDPI- и HDPI-наборы.

Если сайт базируется на общепринятых технологиях (HTML, CSS, JavaScript), то для создания макета и графики сайта принята следующая последовательность действий [46]. Адаптивный сайт распознает ширину браузера и подстраивается, тогда как мобильные сайты должны распознать устройство. Скрипт определения устройства постоянно обновляется. Создается три вида макетов страниц для экранов с шириной 1024x (компьютер, полная версия сайта), 768x (iPad в вертикальной ориентации) и 320x (iPhone в вертикальной ориентации).

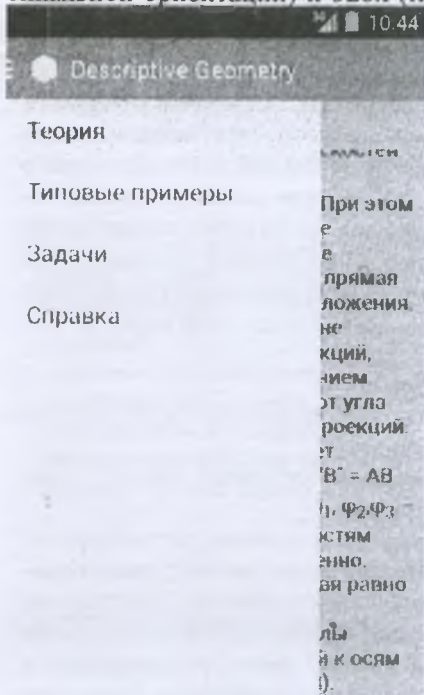


Рис. 65. Главная страница сайта на мобильном устройстве Nexus S (4»; 480x800)

Некоторые приложения имеют и горизонтальную и вертикальную ориентации. Устройство iPhone 4 имеет новый дисплей «retina», который удваивает количество пикселей, вплоть до 2048x1536.

Макеты и графика выполняются изначально для Retina (640x960), а затем сжимаются до обычного экрана (320x480). Это означает, что необходимо также создать 2 набора иконок для макетов.

Ориентация на размеры экрана в 4”, 7” и 10”, принятая нами с самого начала, ввиду специфики содержимого Интернет-ресурса, а также использование Flex, ActionScript, AIR и Flash (как будет показано ниже), избавляет от необходимости выполнять одну из версий макета, графики и иконок. Полная версия сайта при неизменности информационной архитектуры подлежит переделке для достижения узнаваемости ресурс-

са на всех платформах воспроизведения.

Для мобильного приложения, как и полной версии сайта, создается *глобальная* и *локальная навигация*. *Глобальная навигация* представлена на рис. 65. Ее отличие от глобальной навигации полной версии сайта состоит в том, что целиком она будет видна только на главной странице. Предметизация претерпела изменение только в названии одного пункта главного меню – «Теоретические материалы» заменены на «Теория» для мобильной версии ресурса. Названия некоторых лекций в разделе «Теория» также подверглись изменению и сокращению, однако при наведении на сокращенное название пункта во всплывающих подсказках (хинтах) можно прочесть его полное название.



Локальная навигация реализуется следующим образом – название пункта глобального меню помещается в заголовке, а соответствующее ему меню локальной навигации – слева (в столбик). Результат перемещения, соответствующий «хлебным крошкам»: Теория> Проекция точек и прямых> Проекция отрезка прямой линии общего положения. Следы прямой – представлен на рис. 66. Хлебные крошки принято размещать даже на мобильных версиях сайта, поскольку они весьма полезны, однако в текущей версии эта возможность пока не реализована.

Проекция отрезка прямой  $AB$ , не параллельной плоскости проекций, проецируется на нее с искажением. Величина искажения зависит от угла наклона прямой к плоскости проекций. Эта зависимость соответствует выражению:  $A'B' = AB \cos \varphi_1$ ,  $A''B'' = AB \cos \varphi_2$ ,  $A'''B''' = AB \cos \varphi_3$ , где  $\varphi_1, \varphi_2, \varphi_3$  – углы наклона прямой к плоскостям проекций  $H, V$  и  $W$ , соответственно. Возможны случаи, когда прямая равно наклонена к трем плоскостям проекции. Тогда на чертеже углы наклона проекций этой прямой к осям проекций равны  $45^\circ$  (см. рис. 4).

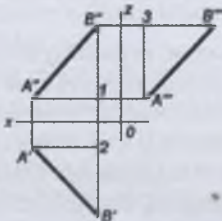


Рис. 4

Рис. 66. Лекция «Проекция отрезка прямой линии общего положения. Следы прямой» из раздела «Проекция точек и прямых» меню глобальной навигации «Теория»

Планируется поместить их под заголовком пункта глобального меню: проблеме представляет недостаток места на экране.

*Контекстная навигация* для мобильной версии сайта существует, главным образом, для того, чтобы визуально выделить понятие, определение которого есть в глоссарии, а также URI ресурса. И хотя устройства указания для мобильных и планшетов – редкость, а попасть пальцем по подчеркнутой строчке не представляется возможным, тем не менее на элементах контекстной навигации оставлены гиперссылки. Это дает возможность практически не переделывать верстку хотя бы в этой части и учесть интересы меньшинства, пользующегося устройствами указания для мобильных устройств. Речь, разумеется, не идет о графических планшетах, являющихся устройствами ввода информации и активно применяемых для работы с САД-системами.

Размеры кнопок при разработке интерфейса пользователя принято создавать такими, чтобы в них было просто попасть пальцем. Нужно добиться, чтобы движения были естественны для человека и всегда была видна реакция и изменения. Размер шрифта должен быть достаточно велик. Несмотря на планируемую оптимизацию под Android, приведем также данные, касающиеся других платформ, поскольку в части дизайна возможно учесть интересы их пользователей. Согласно рекомендациям [2; 3; 6] кнопки должны быть 44x44 пикселя для iPhone, 36x36 для Windows Phone, Nokia же рекомендует делать кнопку не менее 1 см. Это информация из официальных руководств по разработке, в которых даже единицы измерения разнятся. Существуют и отличающиеся рекомендации разработчиков-практиков: кнопка хорошо нажимается, если ее размеры 45-75 dp в зависимости от задачи [141]. Кроме того, следует учитывать, что реальный размер кнопки и другого элемента навигации зависит от плотности пикселей конкретного устройства [46].

После нажатия по кнопке и перехода на новую страницу пользователь должен понимать, где он, как попал сюда, как вернуться обратно. В айфонах и айпадах используется очень правильный, кинематографичный прием – переход справа налево и обратно, что оставляет визуальную связь в структуре окон интерфейса. Следование этому приему позволяет обойтись без лишних напоминаний, экономит «драгоценное» место на экране приложения.

Самой радикальной переделки потребуют неинтерактивные ролики, расположенные в разделе «Типовые примеры». При их создании практически не применялось программирование, использовалось столько промежуточных кадров, эффектов, заливок, прозрачностей, сколько позволяла фантазия графического дизайнера.

Flash-анимация была абсолютным лидером по производительности разработки. Размер swf-файлов достаточно сложной анимации был наименьшим по сравнению с аналогичными файлами других приложений и пр. Забота о минимизации размеров и нагрузки на процессор представлялась излишеством и, как правило, считалась явным признаком принадлежности к старшему поколению разработчиков.

При создании содержимого Flash для мобильных устройств следует придерживаться некоторых основных правил. Следует разделить *правила для анимации и правила для векторных и растровых изображений*.

Приведем наиболее важные правила для успешного создания и последующего воспроизведения *анимации*, которые следуют из собственного опыта и рекомендаций других разработчиков [5]:

- необходимо, чтобы размер рабочей области документа совпал с разрешением целевого устройства во избежание потери производительности при плавном масштабировании;

- следует исключить графические объекты повышенной сложности, а также лишние промежуточные кадры и эффекты прозрачности;

- надо стремиться ограничить количество одновременных промежуточных кадров или расположить их таким образом, чтобы каждый следующий начинался после окончания предыдущего;

- эффекты прозрачности (альфа-эффекты) следует применять к символам экономно, поскольку они требуют больших вычислительных ресурсов. Особенно следует избегать применения промежуточных кадров, содержащих символы с прозрачностью альфа-канала менее 100 %;

- не следует использовать визуальные эффекты, требующие больших вычислительных ресурсов, например, крупные маски, большое количество движения, альфа-наложение, большое количество градиентов и сложные векторы;

- наиболее эффективные результаты могут получиться в результате экспериментов с сочетаниями промежуточных кадров, анимации ключевых кадров и движения на основе ActionScript;

- для визуализации векторных овалов и окружностей требуется намного больше памяти, чем для визуализации четырехугольников. Кроме того, использование круглых и овальных обводок приводит к значительному увеличению загрузки процессора;

#### ◆ Развитие образовательной среды средствами информационных технологий

– при выводе области анимации программа Flash определяет прямоугольную ограничительную рамку вокруг этой области. Чтобы оптимизировать вывод, нужно сделать эту рамку как можно меньше. Следует избегать наложения промежуточных кадров, поскольку Flash воспринимает области слияния как один прямоугольник, что приводит к созданию более крупной общей области. Оптимизировать анимацию можно с помощью функции «Показать область перерисовки» приложения Flash;

– не следует скрывать видеоролики на экране с помощью параметра `_alpha = 0` или `_visible = false`. Если просто сделать видеоролик невидимым или задать для него нулевую альфа-прозрачность, то ролик будет по-прежнему обрабатываться при вычислениях линейной визуализации, что может привести к снижению производительности;

– не нужно пытаться скрыть видеоролик путем наложения на него других графических объектов. Он все равно будет обрабатываться при вычислениях проигрывателя. Вместо этого следует перемещать видеоролики за пределы рабочей области или удалять их путем вызова функции `removeMovieClip`.

*Векторные и растровые изображения* для создаваемого ресурса не имеют такого большого значения, как анимация. Использование векторных изображений в файле, как правило, позволяет уменьшить объем используемой памяти и размер файла. Векторные изображения сохраняют гладкие контуры при масштабировании. Растровые изображения при масштабировании могут становиться угловатыми, и их качество может заметно снижаться.

Приведем некоторые правила создания и сжатия этих изображений для мобильных устройств [5]:

– не использовать контуры в векторных фигурах. Контуры имеют внутренний и внешний край (заливки имеют только один), поэтому для их визуализации требуется вдвое больше работы;

– визуализация углов проще, чем кривых. По возможности следует использовать плоские края, особенно в сочетании с очень маленькими векторными фигурами;

– небольшие сложные изображения (например, значки) следует, как правило, делать растровыми, а более крупные и простые — векторными;

– следует импортировать растровые изображения необходимого размера;



– текст является чрезвычайно сложной векторной фигурой. При использовании текста лучше не применять к нему анимацию и не помещать его поверх анимации. Текст надо растеризовать. Для многострочного динамического текста и текста ввода символ конца текстовой строки не кэшируется. По возможности следует использовать статические текстовые поля;

– нужно как можно меньше использовать прозрачность в PNG-файлах. При перерисовке приложению Flash приходится выполнять вычисления даже для прозрачных частей растрового изображения. Например, при использовании PNG-файла с прозрачным изображением, который представляет элемент переднего плана, не следует экспортировать прозрачный PNG-файл полноэкранного размера. Вместо этого лучше экспортировать файл реального размера элемента переднего плана;

– следует обязательно группировать растровые слои отдельно от векторных слоев. В приложении Flash для растрового и векторного содержимого используются разные средства визуализации, переключение между которыми занимает определенное время.

При использовании растровых изображений нужно установить параметры сжатия изображения (отдельно для каждого изображения или глобально для всех растровых изображений), что позволяет уменьшить размер SWF-файла. В зависимости от характера изображения следует применять JPEG- или PNG-сжатие. JPEG-сжатие лучше использовать для изображений со сложными вариациями цвета или тона, а PNG-сжатие – для изображений с простыми фигурами и небольшим количеством цветов.

На рис. 67 представлен редактор задач в мобильной версии – наиболее важная часть сайта. Очевидно, панель управления, которая в настоящий момент является частью редактора, занимает несообразно много места, которое следовало бы использовать для увеличения изображения. Размеры панели управления следует привязать к размерам экрана устройства воспроизведения.

Редактор задач, который представлялся отдельным завершенным блоком и предназначался для переноса на мобильную версию сайта без изменений, таким образом, требует существенной доработки. Панель управления должна быть вынесена из приложения редактора задач и стать одним из элементов оболочки. Редактор должно переделать так, чтобы он стал полностью событийно управляемым приложением.

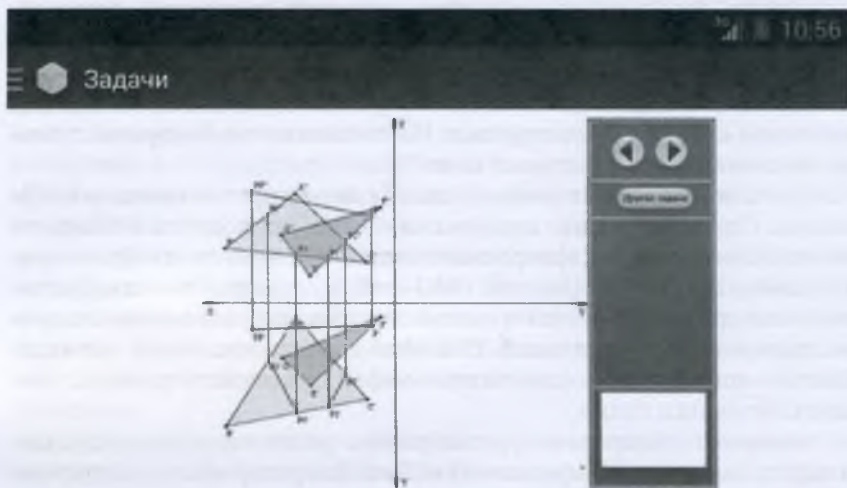


Рис. 67. Пункт меню глобальной навигации «Задачи» и окно редактора задач. Изменение интерфейса потребует глубокой переработки

Контент полной версии учебного сайта не должен радикально отличаться от его мобильной версии, в особенности при одинаковой информационной архитектуре. Не подлежит сомнению, однако, что мобильные приложения требуют особенно внимательного отношения к количеству текста. В этой связи нельзя обойти вниманием третий закон Usability Стива Круга [66]: «Избавьтесь от половины слов на каждой странице, затем уберите еще половину из того, что осталось». Специфика публикуемого контента не позволяет следовать этому закону в полной мере. Однако в нашем случае (как и в любом другом) вполне применимо семнадцатое правило из «Основы стиля» Э. Б. Уайта, послужившее основой для третьего закона Usability: «Опустите лишние слова. Хорошее письмо всегда кратко. Предложение не должно содержать ненужных слов, а абзац не должен содержать ненужных предложений по той же причине, по которой на холсте не должно быть ненужных линий, а в машине – ненужных деталей».

Для учебного сайта по «Начертательной геометрии» эти рассуждения материализовались в виде двух версий контента для полной и мобильной версий.

В отличие от полной версии сайта, контент мобильной версии должен будет содержать единственную формулировку правила,

единственный пример и рисунок для его иллюстрации. От некоторых примеров и рисунков, указанных преподавателем начертательной геометрии, придется отказаться. Некоторые рисунки «Теоретических материалов» представляют собой неинтерактивные Flash-ролики, включенные для наглядности в текст лекции. Единственная их «интерактивность» состояла в том, что анимация запускалась по нажатию на кнопку. Все эти ролики будут пока заменены растровыми рисунками ввиду того, что некоторые из них, выполненные без учета приведенных выше правил, минимизирующих размер и расход вычислительных ресурсов, могут приводить к потере производительности при работе с файлом. Применение других технологий создания анимации для встраивания в html-страницы разрабатываемого ресурса, как будет показано ниже, не является сколько-нибудь приемлемой альтернативой.

*Кроссбраузерность* – свойство сайта отображаться и идентично работать во всех популярных браузерах – в полной мере реализовать проблематично. Достаточно обеспечить отсутствие развалов верстки и устойчивую работу основного функционала. Простое перечисление только самых популярных браузеров для персональных компьютеров и мобильных устройств позволяет тому, кто добивался одинаковости отображения сайта хотя бы для трех браузеров, представить трудоемкость этой задачи:

- Opera, Internet Explorer, Google Chrome, Mozilla Firefox, Safari, NetSurf и др. – браузеры для Windows, Mac OS, Linux;

- Safari, Google Chrome, Atomic, Skyfire, Mercury, Opera Mini, Perfect Browser, Dolphin Browser HD, Firefox, Opera, Miren Browser, Maxthon, Web Viewer, и пр. – браузеры для iOS, Android, BlackBerry.

*Кроссплатформенность* – способность сайта или другого программного обеспечения функционировать в нескольких различных операционных системах или на разных аппаратных платформах.

Требование кроссплатформенности будет выполнено наилучшим образом, если проект реализовать в блочном типе верстки с применением адаптивного дизайна.

Адаптивный web-дизайн (Responsive web design) – дизайн, который проектируется и разрабатывается таким образом, что сайт становится совместимым с любым устройством, которое может быть использовано для работы в Интернете.

Целью адаптивного web-дизайна является универсальность web-сайта для различных устройств. Для того чтобы web-сайт был

удобно просматриваемым с устройств различных разрешений и форматов, по технологии адаптивного web-дизайна не нужно создавать отдельные версии web-сайта для разных видов технических средств. Один сайт может работать на смартфоне, планшете, ноутбуке и телевизоре с выходом в Интернет, то есть практически на всем спектре устройств.

Как уже упоминалось, полная версия сайта будет оптимизироваться для работы на PC под ОС Windows с браузерами Opera, Internet Explorer, Google Chrome, Mozilla Firefox, а мобильная – для работы с мобильными телефонами и планшетами под Android.

**Идеологическая и технологическая составляющая проекта.** Итерационный характер любого проектирования не позволяет описать этот процесс поэтапно и последовательно. Даже информационная архитектура, определяющая основной функционал сайта, может изменяться с учетом проектных решений, принятых на последующих этапах проектирования и реализации – выбор инструментальных средств разработки, верстка и программирование. Описывая информационную архитектуру и дизайн полной и мобильной версии сайта, мы опирались на принятые решения об инструментарии разработки, выбор которого не представлялся очевидным и заслуживает пристального внимания.

При выборе способов достижения *кроссбраузерности* и *кроссплатформенности* сайта следует выделить идеологическую и технологическую компоненты разработки.

Идеологическая компонента состоит в том, чтобы рассмотреть и проанализировать современные инструментальные средства создания адаптивных сайтов (с учетом тенденций развития) на предмет их применимости для наиболее эффективного решения задачи создания адаптивного Интернет-ресурса на базе существующего, наиболее важным содержимым которого является Flash-анимация.

Технологическая компонента разработки состоит в решении вопросов, возникающих в процессе программной реализации поставленной задачи выбранными инструментальными средствами.

Задачи проекта по созданию адаптивного сайта для курса «Начертательная геометрия» состоят в том, чтобы:

- создать мобильную версию сайта;
- обновить полную версию существующего ресурса с целью:
  - заменить некоторые статические изображения в теоретических материалах на анимированные;

➤ исследовать возможности альтернативных методов создания анимации, чтобы избежать необходимости загрузки плагина проигрывания Flash-анимации, хотя бы для теоретических материалов;

➤ избавиться от ограничений, возникающих в процессе работы со стилями при создании html-страниц вследствие недостатков инструментария ActionScript 3.0 и Flex;

В полной мере все задачи разработки и рефакторинга сайта позволяют решить, на наш взгляд, две группы инструментальных средств:

- HTML5, CSS3, JavaScript;
- Flash, ActionScript, Flex Framework, Adobe AIR.

Рассмотрим каждую из них применительно ко всему комплексу поставленных задач.

**HTML5, CSS3 и JavaScript** являются *традиционным* инструментами создания web-приложений самого разного уровня сложности.

HTML 5 – новая открытая платформа, предназначенная для создания веб-приложений, использующих аудио, видео, графику, анимацию и др. В него добавлено 28 элементов с новой семантикой, ряд элементов управления формами, средств создания интерактивных приложений. Особенное значение HTML 5 приобретает в связи с ориентированностью некоторых его инструментов на мобильные платформы. HTML 5 преследует три основные цели [77]:

- установить особенности современных браузеров, в которых возможна интероперабельность;
- определить обработку ошибок;
- доработать язык для упрощения создания web-приложений.

CSS3 – это новый стандарт оформления HTML документов, значительно расширяющий возможности предыдущего стандарта. Он, в частности, позволяет:

- создавать элементы со сглаженными углами;
- формировать линейные и сферические градиенты;
- более гибко оформлять фоновую картинку элементов;
- добавлять тени к элементам и их контенту;
- загружать нестандартные шрифты;
- создавать анимацию и различные эффекты переходов;
- задавать цвета несколькими новыми способами и пр.

Важнейшим свойством CSS3 является модульность (дробление спецификации), позволяющая одним фрагментам спецификации

двигаться быстрее, чем другим, и подталкивает производителей браузеров к тому, чтобы они внедряли хорошо проработанные фрагменты до того как спецификация CSS3 будет считаться целиком законченной [125].

JavaScript – JavaScript – это интерпретируемый язык программирования с объектно-ориентированными возможностями. Кроме того, JavaScript имеет ряд свойств, присущих функциональным языкам:

- функции могут быть переданы как параметр, возвращены из функции, присвоены переменной;
- автоматическое приведение типов, замыкания, что придает языку дополнительную гибкость.

Однако механизм объектно-ориентированного наследования JavaScript скорее похож на механизм прототипов в таких малоизвестных языках, как Self, и сильно отличается от механизма наследования в C++ и Java. JavaScript может использоваться в качестве языка разработки для мобильных устройств Palm с Palm webOS в качестве операционной системы.

Наиболее популярными библиотеками JavaScript являются Dojo Toolkit, Extjs, jQuery, MooTools, Prototype.

Библиотеки JavaScript представляют собой набор многократно используемых объектов и функций. Они разработаны для ухода от деталей реализации достижения приемлемой степени кроссбраузерности при разработке web-приложений.

Особого внимания заслуживает библиотека jQuery, которая идеологически отделяет поведение от структуры HTML, то есть отделяет функциональность веб-страницы («уровень поведения») от структуры, содержания и представления Web-страницы. Библиотека jQuery содержит функциональность, полезную для максимально широкого круга задач, позволят упростить JavaScript и использовать повторяющиеся фрагменты кода так, чтобы не беспокоиться о кроссбраузерных вопросах.

Для того, чтобы собрать для ресурса именно ту JavaScript-функциональность, которая на нем была бы востребована, реализована архитектура компактного универсального ядра библиотеки и плагинов.

jQuery имеет следующие основные возможности:

- представляет движок кроссбраузерных CSS-селекторов;

- реализует переход по дереву DOM, включая поддержку XPath;
- обрабатывает события;
- воспроизводит анимацию и визуальные эффекты;
- добавляет AJAX-на страницы;
- подключает JavaScript-плагины.

jQuery Mobile – мобильная версия jQuery – реализует внедрение клиентского JavaScript в наиболее часто используемые браузеры на мобильных платформах. Перечислим ее ключевые особенности [34]:

- синтаксис построен на JQuery;
- совместимость со всеми основными мобильными платформами – iOS, Android, Blackberry, Palm WebOS, Nokia/Symbian, Windows Mobile, bada, MeeGo и с базовой поддержкой всех устройств, которые понимают HTML;
- небольшой размер файла (12кб, включающий функциональность для всех мобильных) и минимальная зависимость от изображений для увеличения скорости;
- HTML5 разметка, управляемая конфигурацией страниц, и минимальные требования для быстрого развития сценариев;
- прогрессивный подход, усиление основного содержания и функциональность всех мобильных, электронных книжек, Desktop Platform, будет легко использоваться на новых мобильных платформах;
- автоматическая инициализация с помощью HTML5 date-role атрибутов в разметке HTML, автоматическая инициализация всех JQuery Mobile виджетов, найденных на странице;
- включение стандартов доступности активных Интернет-приложений, таких как WAI-ARIA, чтобы обеспечить удобное чтение с экрана (например VoiceOver в IOS) и других вспомогательных технологий;
- простое API, рационализирующее процесс поддержки Touch, мыши и других видов устройств ввода;
- новые модули, расширяющие возможности сенсорного управления;
- мощные темы визуализации приложения делают весьма простым построение интерфейсов.

На стадии alfa релиза jQuery Mobile поддерживает платформы [34]:

#### ◆ Развитие образовательной среды средствами информационных технологий

- Apple iOS: iPhone, iPod Touch, iPad (все версии);
- Android: все устройства (все версии);
- Blackberry Torch (version 6);
- Palm WebOS Pre, Pixi;
- Nokia N900 (в разработке).

Описав главные инструменты создания Интернет-ресурсов, перечислим основные направления достижения адаптивности сайта:

- применение гибкого макета на основе сетки;
- использование адаптивных изображений;
- работа с медиа-запросами.

Техника создания адаптивного сайта с применением традиционного инструментария разработки состоит:

- в грамотном использовании атрибутов `max-width`, `min-width`, `auto` (например, для `<div>`, `<img>`), относительных значений для `margin`, `padding`, шрифта;
- «очистке» атрибута `float` от предыдущего элемента и использования конструкции `overflow:hidden`,
- переносе текстовых конструкций с помощью атрибутов `word-wrap: break-word`;
- использовании медиа-запросов для создания большого количества трансформаций, создавая свое «поведение» разметки для каждого из установленных разрешений экрана;
- мета-тегов, которые влияют на режим отображения Интернет-страниц, CSS-спрайтов, главным преимуществом использования которых является однократная загрузка сервером сразу всех элементов в одном файле, что позволяет существенно сократить количество HTTP-запросов к серверу и повышает производительность;
- создании адаптивных изображений;
- применении JavaScript-библиотеки jQuery и ее надстройки JQuery UI, которые помогают легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими и др.

Адаптивные изображения – одна из проблем адаптивного дизайна. В настоящее время нет общего подхода к адаптации изображений. Есть несколько способов решения проблемы:

- JavaScript;
- изображения большего разрешения.



При помощи скрипта можно считывать разрешения экрана пользователя и подставлять нужные изображения. Второй способ предполагает, что используемое изображение выполнено в разрешении заведомо большем, чем это требуется для большинства применений, а нужный размер достигается применением средств CSS. Ни один из этих способов не решает проблему полностью даже при использовании традиционных средств разработки: скриптовое решение может не работать, если у пользователя будут отключены скрипты в браузере. Подстановка картинки очень большого разрешения не только увеличивается время загрузки страницы, но и непригодна для мобильных устройств. В настоящее время разрабатываются новые методы решения данной проблемы, для этого даже собрана специальная группа специалистов, которая работает над новыми стандартами [80].

HTML5, CSS3 и JavaScript с его развитыми библиотеками условно позволяют создать единый адаптивный сайт для визуализации текстовой и графической информации на самом широком спектре платформ, особенно если учесть наличие и постоянное совершенствование средств работы с мобильными устройствами. Однако для представления анимации и других форм мультимедийной информации их универсальность не столь очевидна.

Поддержка возможностей стандарта CSS3 [125], который реализует множество функций анимации, достаточных для большинства неинтерактивных роликов, требует тщательного тестирования на разных типах устройств и в разных браузерах.

Нами была исследована возможность применения CSS3-анимации для несложных роликов, моделирующих переход к эпюру от пространственной картины расположения плоскостей проекций точек и отрезков прямых (см. рис. 68, рис. 69, рис. 70).

На разработку анимации в стандарте CSS3 требуется примерно в четыре раза больше времени, чем это занимает отрисовка во Flash, и, полагаем, в любой другой системе графического дизайна или CAD-системе, обладающей инструментами объектной привязки.

Возможно, для создания баннеров, простых объектов и движений CSS3 вполне применим. Однако точные геометрические построения, которые требуются для большинства анимационных роликов создаваемого сайта, невозможны без привязки к характерным точкам объекта. В CSS3 для этого требуются усилия, непомерные даже с точки зрения программиста – работа с этим инструментом для описанных задач недопустимо мало производительна.

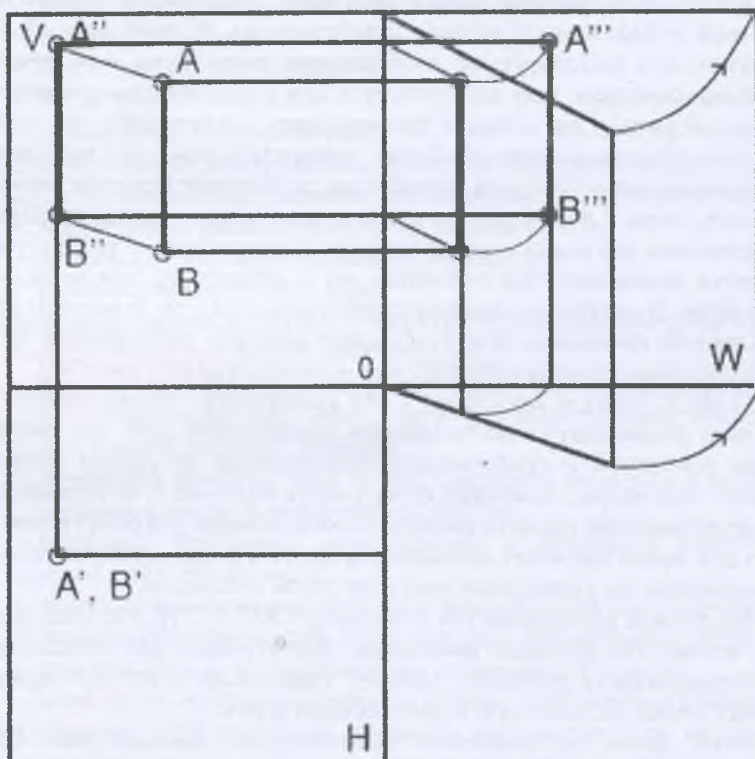


Рис. 68. Промежуточный кадр моделирования перехода к эюру

Безусловным преимуществом CSS3-анимации является отсутствие необходимости подгрузки плагина для проигрывания Flash-анимации. Однако трудно себе представить современный PC, на котором браузер не обладал бы таким инструментом, а, кроме того, без плагина на PC не будет работать редактор задач, и функционал сайта минимизируется. Для мобильных платформ CSS3-анимация также не нуждается в установке никаких дополнительных программ и потому для изображений иной структуры весьма перспективна. Редактор задач на мобильной платформе будет недоступен ввиду того, что соответствующих плагинов для их браузеров не существует.

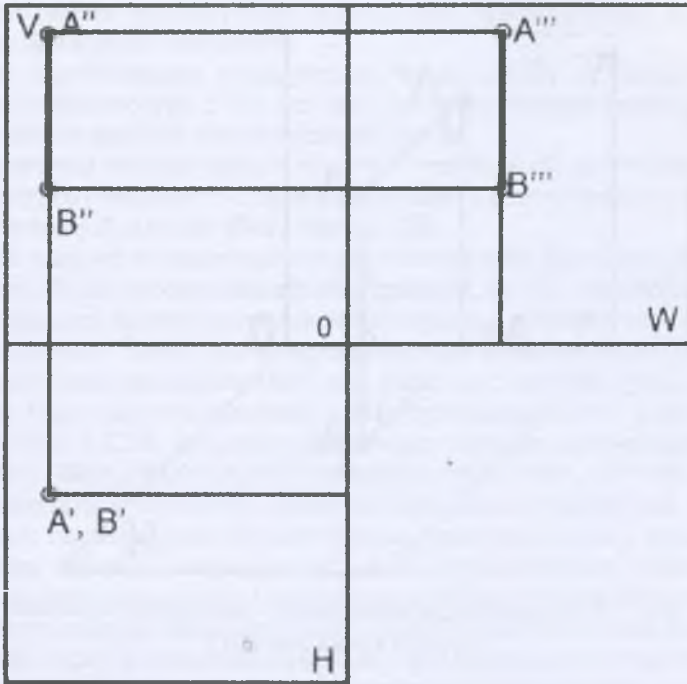


Рис. 69. Последний кадр анимации – переход к энтюру завершен

Анимационный ролик построения проекций точки  $A$  в системе  $H$  и  $V$ , созданный с помощью CSS3, результат работы которого представлен на рис. 70, имеет размер около 11 Kb. Чтобы его реализовать, потребовалось около 300 строк кода (874 символа). Возможно, что этот код можно несколько усовершенствовать и сократить, но ActionScript-программа для такой анимации более чем на порядок короче.

Чтобы добиться кроссбраузерности этого CSS3-ролика, потребуются дополнительные усилия и строки кода – корректно он отработал только в Google Chrome, Mozilla Firefox. В Opera, Internet Explorer возникло множество ошибок.

Важнейшим аргументом за применение Flash-анимации является необходимость минимизировать размер файла для мобильной платформы, что наилучшим образом допускает программирование анимации описываемых объектов на ActionScript 3.0.

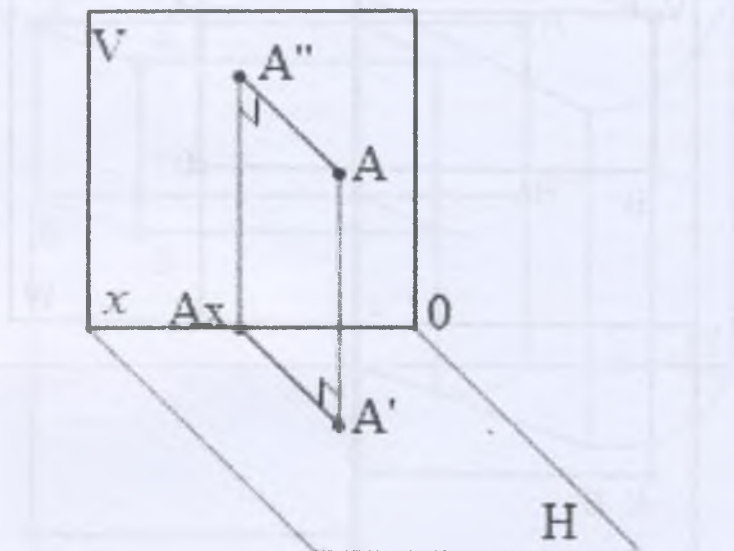


Рис. 70. Результат работы ролика – построены проекции точки  $A$  в системе  $H$  и  $V$

Заслуживают внимания исследования заявленной возможности экспортировать Flash-анимацию в HTML5 и JavaScript при помощи расширения Toolkit for Create JS и других инструментов. Результаты этих исследований описаны в [100]:

- Flash IDE даже полное отсутствие ActionScript-кода не позволило безошибочно преобразовать объекты и анимацию относительно несложного баннера в формат HTML/CSS/JavaScript;

- Adobe Wallaby Technology (инструмент Adobe Labs, аналогичный по назначению, выполненный в виде отдельного приложения и использующий jQuery вместо Create JS Toolkit) тоже не справился с задачей;

- Swiffy от Google по сравнению с двумя выше упомянутыми выглядел чуть лучше – ошибок при экспортировании было меньше;

- ActionScript 2.0, который по своей архитектуре ближе к JavaScript, чем его преемник ActionScript 3.0, часто при экспортировании дает безошибочный результат, но размер файлов данных, получаемый на выходе, на порядок больше исходного – вместо 60 Кб

получено свыше 600 Кб HTML и JavaScript, что запрещено на большинстве баннерных площадок;

– из опробованных конвертеров только Swiffy от Google поддерживает ActionScript 3.0, а без него попытки конвертировать игры и приложения на Flash теряют всякий смысл.

Рассмотрим вторую группу языков и технологий, рассмотренную на предмет применимости для решения задач нашего проекта – **Flash, ActionScript 3.0, Apache Flex, Adobe AIR**.

Особо следует остановиться на ограниченности возможностей ActionScript 3.0 для отображения html-страницы, на том, что послужило основанием для поиска альтернативных средств разработки.

Приложение Flash Player предоставляет расширенный усовершенствованный интерфейс API для работы с текстом [85]. Однако Flash Player не обеспечивает полнофункциональную поддержку языка HTML и CSS. ActionScript 3.0 поддерживает следующие теги: <A>, <B>, <BR>, <FONT>, <I>, <IMAGE>, <LI>, <P>, <SPAN>, <U>. Поддерживаемые атрибуты также весьма немногочисленны. Класс StyleSheet, позволяющий форматировать текстовые поля с помощью каскадных таблиц, базируется на очень ограниченном подмножестве – спецификации CSS1. Собственно, таблицы стилей приложения Flash Player даже не каскадируются. Поддерживаются только базовые селекторы типов и селекторы классов – все остальные множества селекторов не поддерживаются. Это ставит разработчика html-страниц в nepозволительно жесткие рамки – такого функционала явно недостаточно для эффективной работы со стилями. Наличие формул со множеством верхних и нижних индексов и других особенностей текстов на сайте требует более развитых инструментов форматирования.

В связи с этим возникала мысль о переводе страниц из html в другой формат. Идеальным решением для создания переносимых теоретических материалов был бы pdf-формат, который в полной мере обеспечивает точность соотношения всех элементов обозначения, однако анимированные иллюстрации в тексте будут потеряны.

Рассматривая альтернативные варианты создания анимации, следует отметить, что характер изображений разрабатываемого ресурса позволяет применить gif-анимацию, однако для сложных изображений и анимации с большим количеством шагов размер картинки может быть весьма значительным, что ограничивает ее применение.

О достоинствах Flash для создания веб-приложений и мультимедийных презентаций нами уже упоминалось в нескольких рабо-

тах [32; 33]. Следует отметить только, что говорить об Adobe Flash как об отмирающей под давлением HTML 5 и открытых стандартов было бы неправильно. Добиваться приемлемой производительности Flash Player в мобильных браузерах – задача бессмысленная ввиду того, что они не рассчитаны на работу с полноценными RIA-приложениями [100].

Flash использует язык программирования ActionScript, основанный на ECMAScript.. ActionScript – объектно-ориентированный язык программирования, который добавляет интерактивность, обработку данных в содержимое Flash-приложений. ActionScript исполняется виртуальной машиной (ActionScript Virtual Machine), которая является составной частью Flash Player. ActionScript компилируется в байт-код, а он включается в SWF-файл. На ActionScript 3.0 написан редактор задач нашего сайта, язык используется и при разработке некоторых роликов.

Полная версия сайта реализована в виде Flex-оболочки, описанной в работе [33, С. 143–183]. Apache Flex – это большой набор классов, который расширяет базовые возможности Flash, дает возможность описывать интерфейс приложения на XML, ускоряя и упрощая процесс разработки Rich Internet Applications (RIA-приложений). Логика Flex-приложения пишется на ActionScript 3.0. Результатом компиляции Flex-приложения является swf-файл, исполняемый обычно Flash Player в браузере или на платформе AIR уже в качестве самостоятельного приложения. Flex-framework включает возможности локализации, стилизации приложения, разработки модульного приложения, встроенные валидаторы. Flex легко адаптирует известные в Java архитектурные решения, позволяет использовать паттерны проектирования и пр.

Несмотря на то что Adobe не поддерживает Flash Player для неуклонно растущего количества мобильных браузеров, разработка Flash-анимации для мобильных платформ успешно развивается благодаря AIR [76].

Платформа AIR появилась как альтернатива Flash Player в браузерах. Она является кросс-платформенной средой, которая позволяет RIA-приложениям исполняться вне браузера, расширяя, таким образом, их возможности за счет доступа к ресурсам операционных систем [100]. AIR позволяет разработчику, не изменяя код, сделать из Flash-сайта AIR-приложение [4]. Сайт преобразуется в кроссплатформенное приложение, которое можно скачать

и установить на различных устройствах: от PC под управлением Microsoft Windows или MacOS до Apple iPhone, коммуникатора с ОС BlackBerry или Android, то есть везде, где поддерживается технология Adobe AIR. Для работы этому приложению будет не нужен браузер – оно будет работать точно так же, как любое другое в среде ОС, и запускаться по клику по ярлыку на рабочем столе. В этом состоит основное отличие Adobe AIR от Adobe Flash Player для браузера. Следует отметить, что, в отличие от сайта, приложение предварительно нужно скачать, установить и запустить. Так же, как это происходит с любыми другими приложениями для iOS, Android, BlackBerry, PC и Mac [100]. Разместить его можно, например, в Apple App Store, Android Market, BlackBerry App World, в Intel AppUp, в GooglePlay или на собственном ресурсе, в зависимости от назначения.

Возможность не переписывать Flex-оболочку для создания мобильной версии сайта является серьезнейшим аргументом за использование AIR при разработке.

Adobe AIR обеспечивает полнофункциональную поддержку языка HTML и CSS, аналогичную той, которая присутствует в браузерах, так что проблема отображения теоретических материалов сайта, по крайней мере для мобильной его версии, полностью снимается. От полного отказа от браузерного варианта воспроизведения, на наш взгляд, следует воздержаться, несмотря на весь небезосновательный оптимизм относительно AIR-платформы [100]. Любой благоразумный пользователь не рискнет устанавливать себе на машину незнакомое приложение, не изучив основательно все его возможности в «нейтральной зоне» – в online под управление браузера.

AIR-приложение располагается на сайте в области глобальной навигации и представляет собой apk-файл.

Apk-файл является подобием архива, где расположены ресурсы и скомпилированные файлы для работы под OS Android – все файлы сайта, от swf-роликов до файлов, формирующих структуру ресурса.

Таким образом, адаптивность существующего ресурса [33, С. 143–183] будет достигнута путем применения достаточно однородных инструментальных средств – Flash, ActionScript 3.0, Apache Flex, Adobe AIR. Потребуется значительная переработка графического содержимого теоретической части, неинтерактивных роли-

#### ◆ Развитие образовательной среды средствами информационных технологий

ков, самого редактора задач, однако применимость существующей Flex-оболочки значительно сокращает трудозатраты по сравнению с другими инструментами разработки адаптивных сайтов.

Тенденции развития избранных инструментальных средств развития сайта компаниями-разработчиками, растущее количество подобных приложений на специальных площадках в сети свидетельствуют о справедливости принятого нами проектного решения. Интернет-ресурс для поддержки курса начертательной геометрии будет существовать в виде отдельного приложения в мобильной версии и в прежней, отчасти переработанной, полной браузерной версии.