

предоставляет возможность замены имитируемых элементов реальными объектами с помощью Simulation Interface Toolkit, что позволяет существенно сокращать время и затраты на проектирование и исследование различных автоматических систем [6].

Современные технологии бурно развиваются, что приводит к расширению методов и форм исследования и проектирования систем. Большинство систем автоматического управления необходимо моделировать перед внедрением перед эксплуатацией, и LabVIEW обеспечивает инженеров, студентов мощным и удобным средством программирования, которое широко используется не только для моделирования, но и для автоматизации и управления различными технологическими процессами в промышленности и в научных исследованиях.

Литература

1. Тревис Дж. Labview для всех. – М.: ДМК Пресс, 2005. -544 с.
2. Васильев В. Г. Моделирование систем автоматического управления в программной среде LabVIEW. – Тверь, 2007. – 25 с.
3. TechTeach [Электронный ресурс] – Режим доступа – <http://techteach.no/> свободный Загл. с экрана
4. Introduction to Control Design and Simulation using LabVIEW, By: Erik Luther, Rice University, Houston, Texas
5. Жуков К.Г. Модельное проектирование встраиваемых систем в LabVIEW. – М.: ДМК Пресс, 2011. – 688 с.
6. Hocdelam Group [Электронный ресурс] – Режим доступа – <http://hocdelam.org/> свободный – Загл. с экран

ПРОГРАММНЫЙ КАРКАС ДЛЯ СОЗДАНИЯ СПРАЙТОВОЙ АНИМАЦИИ НА HTML5

Лизин А.С, Коровин А.С.

Томский политехнический университет
634050, Россия, г. Томск, пр-т Ленина, 30
E-mail: sogimu@nxt.ru

Нашу жизнь теперь невозможно представить без веб-технологий, с помощью них можно общаться, развлекаться, зарабатывать. Веб-технологии дают огромные возможности пользователям, к тому же эта отрасль постоянно растет и совершенствуется. С помощью веб-технологий можно создавать высокопроизводительные, красивые и легкие приложения, которые выполняются в веб браузере, поэтому их можно считать кроссплатформенными, так как веб браузер позволяет одинаково отобразить приложение как на мобильных устройствах (планшетах, телефонах), так и на настольных компьютерах.

С появлением технологии HTML5 2d-context для создания растрового двумерного изображения в браузере, в связи с ее новизной и еще недописанной спецификацией, перед разработчиками появилась задача упрощения разработки с использованием данной технологии. Одним из вариантов упростить процесс разработки, есть разработка специализированного программного каркаса, который применяется для сокращения времени разработки и минимизации ошибок программирования.

Программный каркас является программным обеспечением, объединяющим разные компоненты большого программного проекта. Обычно для разработки и применения каркаса используются техники объектно-ориентированного программирования, так как именно парадигмы объектно-ориентированного программирования позволяют создать эффективную архитектуру каркаса, которая основана на наследовании.

В итоге, проанализировав все технологии разработки программного каркаса был создан программный каркас для создания спрайтовой анимации, который поделен на модули, так как с помощью деления на модули можно выбирать загружаемые компоненты, что значительно позволяет уменьшить время загрузки и старта приложения. Каркас разделен на два основных модуля: ArmGraph и ArmContext.

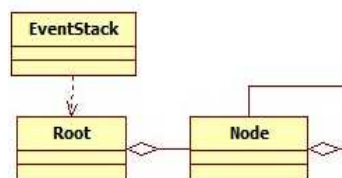


Рис. 1. Диаграмма классов ArmGraph

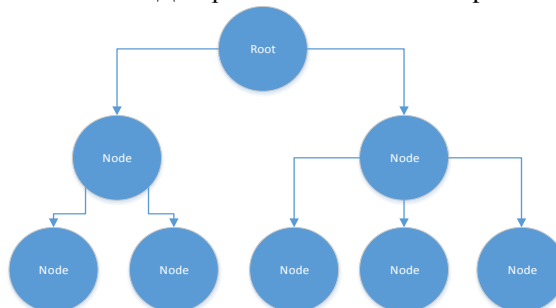


Рис. 2. Пример графа сцены

Модуль ArmGraph – это граф сцены, который является структурой, упорядочивающая объекты сцены в древовидную зависимость, что позволяет удобно организовать навигацию по объектам.

Корневым объектом сцены является объект Root, узлами сцены являются объекты классов производных от класса ArmObject. Наличие Root позволяет ассоциировать всю сцену с одним объектом.

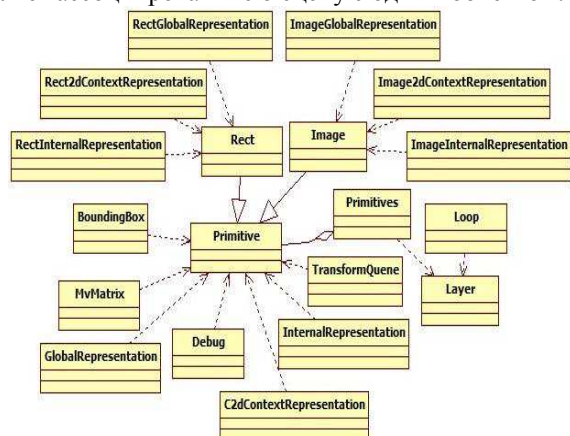


Рис. 3. Диаграмма классов ArmContext

Модуль ArmContext отвечает за графический слой, а именно содержит классы обертки для графических примитивов, которые описывают внешний вид и свойства примитивов, содержат методы для проведения аффинных преобразований, а также методы с помощью которых можно получать информацию об основных точках примитива на экране. Также модуль вводит класс содержащий объект 2d-context и методы для работы с ним, класс, отвечающий за загрузку мультимедиа зависимостей.

Программный каркас для создания спрайтовой анимации позволяет:

1. Предоставлять пользователю возможности взаимодействия с объектами. Поэтому был реализован механизм событий, который позволяет каркасу реагировать на все действия пользователя.
2. Не реализовывать графический цикл, т.е. с разработчика убрана потребность в создании цикла перерисовки изображения.
3. Предоставлять классы описывающих графические примитивы, что дает возможность работать с примитивом как с объектом, а не изображением на экране. Так же данные классы позволяют получить исчерпывающую информацию о местоположении примитива на экране.
4. С помощью узлов сцены изменять свойства примитивов т.е. их положение, угол поворота, цвет и т.д. на протяжении некоторого времени.
5. Легко изменить способ отрисовки (2d-context [1], WebGL [2] и т.д.) через написание соответствующего класса.
6. Снять с разработчика необходимость реализации некоторых оптимизаций, например очистка только необходимой части экрана или округление экранных координат.

Сейчас в интернете можно найти множество каркасов, которые работают с HTML5 тегом <can-

vas>. Три самых популярных каркаса на данный момент – это JCanvasScript [3], KineticJS [4] и LibCanvas [5]. У всех трех отсутствует загрузка мультимедиа-зависимостей объектов, зато присутствует поддержка создания готовых графических циклов, поддержка слоев и все примитивы имеют объектно-ориентированное представление. Все библиотеки абсолютно бесплатны и свободно распространяются по лицензии MIT или GPL:

1. JCanvasScript – Библиотека довольно легковесная, отсутствует модульность и вызовы цепочкой. С точки зрения поддержки, отсутствует форум, на котором можно задать вопрос связанный с JCanvasScript, зато присутствует обширная документация.

2. LibCanvas – Библиотека довольно много весит, что довольно сильно влияет на загрузку страницы. Также присутствует зависимость от внешней библиотеки – AtomJS, что увеличивает дополнительный вес подгружаемых библиотек. Зато есть модульность и вызовы цепочкой.

3. KineticJS – Библиотека, которая весит довольно немного. Присутствуют модульные тесты, форум, и документация, к тому же обладает большим количеством туториалов. Отсутствуют вызовы цепочкой.

В итоге, в ходе анализа существующих программных каркасов для создания спрайтовой анимации на HTML5 и анализа технологии создания программного каркаса был создан программный каркас, который позволяет упростить процесс разработки графического веб приложения, а также позволяет реализовать все возможности тега <canvas>. Очевидно, что использование технологии тега <canvas> технологии HTML5 – это шаг в будущее, так как эта технология, позволяет эффективно создавать растровые двумерные изображения и отображать их на разных платформах.

Литература

1. W3C. 2d-context specification [Электронный ресурс]. – Режим доступа: <http://www.w3.org/TR/2010/WD-2dcontext-20101019/>, свободный. Дата обращения: 12.10.2013.
2. Context. WebGL – a new dimension for browser exploitation [Электронный ресурс]. – Режим доступа: <http://www.contextis.com/research/blog/webgl-new-dimension-browser-exploitation/>, свободный. Дата обращения: 12.10.2013.
3. Официальный сайт каркаса JCanvasScript [Электронный ресурс]. – Режим доступа: <http://jscript.com/>, свободный. Дата обращения: 12.10.2013.
4. Официальный сайт каркаса KineticJS [Электронный ресурс]. – Режим доступа: <http://kineticjs.com/>, свободный. Дата обращения: 12.10.2013.
5. Павел Пономаренко. LibCanvas [Электронный ресурс]. – Режим доступа: <https://github.com/theshock/libcanvas>, свободный. Дата обращения: 12.10.2013.