

Иначе говоря, автоматизация – это лучший вариант для проведения простых повторяющихся тестов. При этом ИИ-тестирование лучше выявляет и предсказывает типичные проблемы, выявленные в ходе такого анализа.

На данный момент многие крупные предприятия активно интересуются и реализуют методы машинного обучения и искусственного интеллекта в процессе тестирования своего ПО. Однако в этой тенденции есть и неприятный нюанс: большинство из них ограничиваются автоматизацией лишь малой части своих тестовых мероприятий, что значительно снижает эффективность таких инноваций. Те компании, для которых качество и скорость разработки программного обеспечения играет ключевую роль, с каждым днём всё активнее интегрируют в схему своей работы методы автоматического тестирования.

УДК 621

Санцевич С. Н.

РИГГИНГ И СКИННИНГ В КОМПЬЮТЕРНОЙ ГРАФИКЕ

БНТУ, г. Минск

Научный руководитель: ст. преподаватель Ражнова А. В.

В настоящее время технологии развиваются каждый день и компьютерная графика не исключение. В процессе создания трёхмерных мультипликаций, разработки 3D-игр с персонажами, неотъемлемой частью является создание самого персонажа.

В трёхмерной анимации различные персонажи – это зачастую полигональные объекты, состоящие из так называемых полигонов (многоугольников), у каждого из которых есть вершины, называемые вертексами (англ. *vertex*). Чтобы, например, согнуть руку персонажу, нужно передвинуть вертексы руки в определённое место, но сделать это так, чтобы между начальным и конечным положением они переместились по определенной траектории, создавая ощущение враще-

ния вокруг анатомического сустава. Для этого используются кости. Они обычно имитируют повороты реальных костей, положение каждого вертекса зависит от положения определенных костей, причем на один вертекс может действовать сразу несколько из них. В одном персонаже в зависимости от сложности требуемых движений может быть от 20 до 100 и более костей и влиятелей (любые другие элементы, выполняющие роль, аналогичную костям, но находящиеся в другой категории и управляющиеся иначе).

Итак, риггинг – это процесс подготовки персонажа к анимации, включающий создание и размещение внутри трёхмерной модели рига (от англ. *Rig* – оснастка), виртуального «скелета» – набора «костей» или «суставов» (англ. *bones, joints*), установления иерархической зависимости между ними и значений возможных трансформаций для каждой из этих костей. Скелетная анимация, для которой и применяется риггинг (а с ним и скиннинг) удобна прежде всего тем, что позволяет манипулировать большим количеством составных элементов анимируемой фигуры (конечности, глаза, мышцы лица, губы и т.д.) с помощью относительно малого количества управляющих элементов – тех самых костей и их регулируемых характеристик.

С риггингом напрямую связан процесс скиннинга (от англ. *skin* – кожа) – назначения связей между участками поверхности уже самой анимируемой фигуры и костями рига. Иными словами, необходимо, чтобы кость, соответствующая крайней фаланге пальца, влияла только на нужную группу вершин на поверхности анимированную 3D фигуры, и чтобы не получилось, чтобы изменение положения этой фаланги приводило к загибанию всей руки персонажа в рогалик.

У этого процесса немало своих подводных камней. Например, необходимо точное определение, какая группа вершин подвергается воздействию трансформаций отдельных костей, чтобы не зацепить ненароком другие; на какие группы вершин подвергаются воздействиям более чем одной кости и так далее. В сущно-

сти, риггинг и скиннинг – это отдельные профессиональные области компьютерной графики, которые могут изучаться при подготовке разработчиков игр, аниматоров, композёров и др.

УДК 621

Санцевич С. Н.

ТЕХНОЛОГИЯ CSS-IN-JS

БНТУ, г. Минск

Научный руководитель: ст. преподаватель Астанчик Н. И.

Некоторые Web-разработчики считают, что технология CSS-in-JS появилась несколько лет назад. На самом же деле, этой технологии уже более 20-ти лет. CSS-in-JS был разработан в 1996 году, но его закрыли в пользу CSS2.

В то время сфера информационных технологий постоянно расширяла своё влияние на другие сферы деятельности. С течением времени увеличились как спрос на приложения, так и предъявляемые к ним требования, вследствие этого увеличился и объём кода в разрабатываемых приложениях. Для того, чтобы сократить количество программного кода, некоторые программисты вернулись к использованию технологии CSS-in-JS.

CSS-in-JS даёт разработчикам наибольший уровень абстракции, нежели статический CSS, поэтому разработчик практически всегда сможет обойти ошибки или проблемы, которые могут возникать в браузерах.

С помощью встроенных стилей можно получить все программные возможности JavaScript, что дает преимущества в виде предварительного процессора CSS (переменные, примеси и функции), а также помогает решить множество проблем, возникающих в CSS, таких как конфликт пространства имен и применения стилей.

Также одной из особенностей CSS-in-JS являются стили, которые контролируются данными. Существует несколько видов таких стилей: стили, которые базируются на состоянии (состоя-