

Федеральное государственное бюджетное образовательное
учреждение высшего образования

**ВСЕРОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ КИНЕМАТОГРАФИИ
ИМЕНИ С.А. ГЕРАСИМОВА**

Кафедра киноведения

На правах рукописи

Гершанов Григорий Владимирович

Создание цифрового антропоморфного персонажа в медиа индустрии

Направление подготовки 50.06.01 -Искусствоведение
Направленность 17.00.03 Кино-, теле- и другие экранные искусства

**Научный доклад
об основных результатах научно-квалификационной работы
(диссертации)**

Научный руководитель

Торопыгина Марина Юрьевна,
кандидат искусствоведения,
доцент

Рецензент

Кочеляева Нина Александровна,
кандидат исторических наук

Москва, 2019

Содержание

Введение.....	3
Глава 1. Проблематика эстетики и технологии в работе художника 3D-персонажей.....	4
1.1. Принципы этапов создания трехмерной модели.....	4
1.2. Трансформация классических наук прикладного искусства в цифровой вид.....	6
1.3. Понятие карты разности и их применение в игровой и кинопродукции.....	8
Глава 2. Мифы современного CGI-сообщества.....	13
2.1. Миф об идеальном инструменте.....	13
2.2. Узкоспециализированные решения программного обеспечения.....	17
2.3. Ассоциативная подмена профессий программным обеспечением.....	19
Глава 3. Революционные решения развития эстетики и технологии цифрового программного обеспечения	22
3.1. «Маленькая революция» систем видеомонтажа.....	22
3.2. «Маленькая революция» принципов создания цифровых персонажей.....	25
Глава 4. Миф об универсальной CGI-профессии.....	29
4.1. Пайплайн и его прохождение на киноигровом проекте.....	29
4.2. Основные отличия продакшн-референсов для 2D- и 3D-персонажей.....	30
Заключение.....	34
Список использованной литературы.....	35
Фильмография.....	36

Введение

Работа посвящена одной из самых малоизученных, но наиболее актуальных на сегодняшний день областей, относящихся к самым разным сферам современной жизни: речь идет о компьютерной графике, имеющей свое технологическое воплощение в виде антропоморфного образа. В работе проанализированы эстетика развития художественно-прикладных наук и их трансформация в качестве поэтапного создания современного цифрового образа.

К концу XX века компьютерные технологии стали неотъемлемой частью кинопроизводства, хотя основной процесс создания персонажей кардинально не менялся (как и были сорок лет назад сплайны, нурбсы, безье и полигоны, так они и сохранились). Значительные изменения претерпела в основном визуальная составляющая. Раньше все компьютерные персонажи воспринимались исключительно как смоделированно-нереальные. Это происходило во многом за счет несовершенства технологий создания (моделирования) и визуализации. В фильмах 1980–1990-х годов любое применение компьютерной графики сразу становилось заметным. Если раньше компьютерный персонаж появлялся в кинематографе, то по причине еще неразвитых технологий он воспринимался как нереальный. Если такого персонажа (героя) выбрасывают из окна, то данная сцена вызывает у зрителя другие эмоции, нежели если бы на его месте был реальный актер. Сопереживание цифровому персонажу не происходило из-за его условности. Прошло время, и технологии шагнули далеко вперед, а к началу XXI века CGI стали неотъемлемой частью медиаккультуры.

Глава 1. Проблематика эстетики и технологии в работе художника 3D-персонажей

1.1. Принципы этапов создания трехмерной модели

Преимущество наличия современных программных продуктов для создания цифровых персонажей позволяет художнику максимально сосредоточиться на создании придуманного им образа и минимально отвлекаться на технические ограничения, связанные с рабочим процессом. На самом деле, проблемы технической оптимизации никуда не исчезли, как хотелось бы считать производителям данного программного обеспечения, просто порядок преодоления этих проблем меняется поэтапно. Если раньше создание цифрового героя предполагало собой весьма грубое моделирование образа с учетом сохранения низкополигональной – анимабельной сетки, – то на данный момент, современные программные продукты позволяют художнику максимально сосредоточиться на художественном образе.

Чтобы компьютерная модель могла подвергаться анимации, строение сетки этой модели должно быть сделано с учетом определенного условия – это плотности самой сетки. Как правило, большинство 3D-редакторов для дальнейшей анимации и визуализации модели, в том числе и игровые движки, не могут работать с высокоплотной сеткой. Также в список требований входит и норма самой структуры сетки. Классическая сетка представляет собой полигональную меру исчисления геометрии.

При создании модели пользователь может интерактивно переключаться между тремя уровнями редактирования сетки. Мерой всех вещей в создании компьютерной модели является полигон (Polygon). По словообразованию можно понять, что «поли-» переводят как «множество». Только множество чего? Полигон образует собой компьютерный квадрат, из которого непосредственно и состоит 3D-модель – это самый общий уровень

взаимодействия с моделью. Спускаясь на уровень ниже, можно обнаружить, что этот квадрат состоит из четырех ребер (Edge), которые, собственно, его и образуют. Когда пользователь взаимодействует с ребром, то он получает более точный контроль над квадратом. Спускаясь еще на один уровень ниже, становится ясно, что ребро состоит из двух точек (Vertex). При взаимодействии с точками пользователь может получить уже контроль над структурой ребра.

Существуют еще классы компонентов геометрии 3D-модели, такие как треугольники (Tris) и Nгоны (Ngon). Данный вид геометрии является нежелательным при классическом моделировании.

Давайте разберемся – почему? Начнем с треугольников: на что стоит обратить внимание – это то, что квадратный полигон можно представить как 2 треугольника, если провести ребро по диагонали между его точками. По такому принципу работает операция Triangulate, которая служит для того, чтобы описать полигональную поверхность в треугольниках таким образом, чтобы оставить структуру, но уменьшить фактический размер модели в 2 раза. Этим алгоритмам очень часто подвержены игровые движки, которые чтобы уменьшить нагрузку на железо компьютера, преобразуют модель. Проблема как-раз и кроется в том, что если алгоритм автотриангуляции модели столкнется с треугольником пользователя, он создаст не 2 треугольника, а в лучшем случае 4 или вовсе геометрию с пустыми точками, что будет не лучшим образом влиять на структуры карт и текстуры, т. к. могут образоваться замыленности. Также триангулированная геометрия плохо подвержена деформации, что может вызвать артефакты, вплоть до разрыва сетки. Nгоны – это как раз и есть полигон с лишними точками, который свиду может даже иметь форму квадрата, но по структуре состоять из компонентов, число которых будет больше четырех. Такая геометрия также является неправильной.

1.2. Трансформация классических наук прикладного искусства в цифровой вид

Теперь данная проблема существует не на этапе создания модели, а на этапе оптимизации. Этот процесс никак не связан с созданием образа, а заниматься им, по-хорошему, должны совсем другие специалисты из других отделов.

Парадокс заключается в том, что для удобства работы специалистов из следующих отделов, в том числе анимация и визуализации, сверхдетализированная модель не подходит.

Аниматоры не в состоянии будут сделать риггинг (Rigging)¹(настройка виртуального скелета), т. к. виртуальные кости (Joints) не смогут взаимодействовать с хаотичной высокополигональной структурой геометрии трехмерной модели. В данном случае на этапе оптимизации мы фактически сводим геометрию нашей модели к анахроничной версии, тем самым облегчая взаимодействие с ней.

Перед нами 2 версии одного и того же образа: высокополигональная, отвечающая всем современным требованиям по уровню графики и детализации модель, и примитивная, низко полигональная модель, больше напоминающая обрубку², но при этом прекрасно оптимизированная под дальнейшую работу с ней. Сама эта формулировка подталкивает к следующему выводу: как сохранить высоко детализированный образ и при этом не потерять удобную для работы кастомизированную геометрическую модель?

На самом деле первые попытки ответить на этот вопрос были сделаны сразу после революционной короткометражки «Голова Юты» 1974 года³. Спустя 4 года, Джеймсом Блинном (который, кстати, и создал технологию

¹ После того как базовая модель готова, чтобы аниматоры могли начать непосредственно процесс анимирования, за дело берутся специалисты из отдела риггинга, которые собственно и создают аниматорам все рычаги управления моделью.

² Художественная модель тела или головы, разбитая на плоскости.

³ «Faces and Body parts – Utah Head» 1974 By Fred Parke: цифровой короткометражный фильм, представленный исследователями компьютерных технологий университета Юты.

Blinn, знакомую всем современным пользователям 3D-пакетов) была представлена первая прототипная технология BumpMapping³.

Прежде чем анализировать дальше, стоит сделать важную оговорку, сказав, что вся суть данной технологии заключается в создании видимости высокой детализации на низко полигональной модели, которая такую детализацию содержать не может.

Конкретно BumpMapping создавал эффект видимости рельефной поверхности (шероховатости, выпуклости, неровности, вмятины) на однородной геометрии. Происходило это за счет того, что сама технология не вносила изменений в геометрию, а была всего лишь настраиваемой текстурой, которая проецировалась на объект.

Аналогию данного математического процесса ретопологии⁴ объектов, то есть замены изначально непригодной «эскизированной⁵ сетки» на структурированно-анимационную, можно провести с техникой создания гипсовых объектов в условиях конвейерно-поточкового производства. Представим, что необработанная сетка модели, которая имеется на этапе эскизирования трехмерного объекта в трехмерном моделировании, – это аналогия физического материала, используемого только для создания чернового художественного образа в скульптуре.

Происходит это следующим образом: набирается общая масса необработанного и заранее не пригодного для «эталонной модели» материала, но отвечающего всем требованиям эластичной и удобной работы скульптора. Из этого материала художник создает нужную ему форму и силуэт, который затем детально прорабатывается. Готовая модель

³Bump Mapping — простой способ создания эффекта рельефной поверхности с детализацией большей, чем позволяет полигональная поверхность. Эффект главным образом достигается за счет освещения поверхности источником света и черно-белой (одноканальной) карты высот, путем виртуального смещения пикселя. Как правило, Bump Mapping позволяет создать не очень сложные бугристые поверхности, плоские выступы или впадины, на этом его использование заканчивается.

⁴ Перестроение трехмерной сетки объекта.

⁵ Высоко полигональная сетка, предназначенная для создания образа, но не соответствующая никаким техническим нормам – не анимабельная.

помещается в цилиндрическую емкость и заливается силиконом или другим аналогичным материалом (загустителем), затем эту емкость встряхивают, чтобы избавиться от попадания воздуха внутрь загустителя, иначе говоря, чтобы не было пузырей. Далее «материал» на модели сохнет, после чего силикон аккуратно разрезается на две части и эскизированный образец вынимают. Все, он больше не нужен. Затем уже готовую форму заливают гипсом или другим материалом. Теперь художник может при необходимости внести мелкие коррективы уже в гипсовую модель.

Что процесс ретопологии в 3D-моделировании, что отливка иного материала по эталонной форме в скульптуре – действие исключительно инженерного характера, задачей которого, в обоих случаях, является оптимизация материала нужного объекта к требованию той или иной индустрии.

Помимо замены материала (сетки) в обоих случаях, еще немаловажным фактом является текстура эталонной модели, которая при соприкосновении с силиконовой оболочкой создает некий слепок нужной формы, способный помимо общего очертания модели передать ее мелкие детали.

Сам слепок, то есть отпечаток скульптуры, можно представить в виде карты. Иными словами, если взять трехмерный силиконовый слепок и разложить его на двумерное изображение, то перед нами будет плоская поверхность с вдавленными и выдавленными участками. Именно она и отвечает за мелкие детали на силуэте «нового» образца». В трехмерной графике тоже представлена данная модель переноса мелких деталей на силуэт. Ее, как и в этом случае, можно представить как двумерную карту. За этот процесс отвечают так называемые карты разности.

1.3. Понятие карты разности и их применение в игровой и кинопродукции

Математических алгоритмов этих карт существует немало. Это может быть и алгоритм BumpMapping (*см. выше*). Вообще, алгоритм BumpMapping в современных программных продуктах представляет собой просто узел, через который можно подключать карту, алгоритм которых заключается в эмуляции глубины модели – Normal.

Displace подключается через другой узел, который влияет на более сложные уровни редактирования. Разница заключается только в том, что в настройке узла Bump можно через настройки Шейдера управлять силой микро рельефа. В настоящее время для органической модели отсутствует эталонный образец карты. Неважно, с какой из карт «разностей» предпочитает работать специалист, сам алгоритм поведения карты неизменен в независимости от выбора, т. к. самой разницей в любом случае обозначают разницу между самой низкой и самой высокой точкой выдавливания на этой двухмерной карте. Именно поэтому иное название у этой карты – карта высот.

Если рассматривать способы применения различных математических иллюзий, то индустрия разделилась на два лагеря: компьютерным играм отошла иллюзия HiPoly геометрии через NormalMapping, аниматики – кинематографические фильмы и мультипликация, они обманывают своего зрителя при помощи DisplaceMapping.

DisplaceMapping – более сложная технология иллюзии рельефа там, где его нет. В отличие от того же Normal, где путем цветовой информации происходит проецирование деталей, алгоритм DisplaceMapping физически деформирует каждый полигон. Учитывая, что основным рендером любой компьютерной игры является сам игровой движок (Engine), то загрузка карты Displace, в отличие от Normal, очень сильно влияет на производительность системы, тем более если на объекте предполагается деформация и анимация.

Для того чтобы спроецировать карту с деталями художественной модели на низкополигональную, нам нужно ответить еще на один вопрос: по какому же принципу строится наложение карты на 3D-модель?

Для этого давайте вспомним сам принцип построения карты. Ведь карта, снятая с трехмерного объекта, должна быть представлена в двухмерном виде.

Эти отрезки двухмерного представления 3D-модели называются UV. Иначе говоря, чтобы сделать двухмерную карту для 3D-модели, сначала необходимо разбить трехмерную модель на участки, которые впоследствии необходимо развернуть в плоскостном измерении, в них запекается информация о картах разности. Иначе говоря, UV служат именно некой прослойкой между низкополигональной сеткой инженерной модели и высокополигональным слепком эталонно-художественного образца. Именно UV создает форму для запекания карт разности.

Карты разностей, в отличие от текстурных карт, используют двухмерное наложение цвета на модель. Иными словами, вся информация о текстуре модели сводится в итоге к каналу цвета. Текстуры никак не влияют на геометрию самой модели. Вопрос в том, а в каких каналах представлены карты разностей? Как ни странно, но по техническому представлению любая карта «разностей» ничем не отличается от той же самой текстурной карты. (За исключением Displace, но об этом чуть позже). Вся ее информация, которая направлена на прямое воздействие, на геометрию, также представляет собой канал цвета, иначе говоря, простую картинку. Например, регулируя яркость и контрастность и того же Displacement, цветовое представление которой ограничивается только градациями черного и белого, можно повышать или понижать уровни вдавливания карты на модели.

Стоит сказать, что сейчас появился еще один формат эмуляции, так называемой VectorDisplacement (VDM), который помимо использования осей X и Y задействует сдвиг информации по оси Z, тем самым обеспечивая более качественную информацию на объектах со сложным рельефом.

NormalMap, помимо всего прочего, ограничен не только сферой действия, точнее ее отсутствия, в отличие от Displacement, но и своей цветовой моделью.

Формально любая карта представляет собой оформление тайла UV с закодированной цветовой информацией. Как было сказано до этого, фактически, с точки зрения изначального языка, карты ничем не отличаются от текстурных карт. Так ли это? Возникает другой вопрос, чем же является цвет в пространстве математических алгоритмов?

Стоит отметить, что цветовая информация на языке математики представляет собой набор растровых пикселей с информацией о горизонтальном (ось X) и вертикальном расположении (ось Y). Каждый же пиксель цвета в классическом цветовом пространстве глубины имеет вариативность только одного цветового оттенка, именуемым математическим префиксом (скажем, красный цвет обозначается: #f71010). Каждый канал, к которому возможно подключить какую-либо информацию о цвете, будет воспринимать ее совершенно по-разному.

Для того чтобы цвет воспринимался только как передача цветового значения, используется канал Color⁶. Если же брать за основу такое понятие, как «маска» (Masking), то, во-первых, ее цветовой диапазон монохромен, то есть уже спектр считывания цвета в этом канале ограничен от белого до черного с использованием оттенков серого между ними. Во-вторых, она воспринимает эти цвета не как классический Color-канал. Канал Masking работает следующим образом: белый цвет для него обозначает то, что информацию можно отобразить, черный – что ее надо скрыть. Несложно догадаться, что серый цвет и оттенки отвечают за прозрачность (так как это баланс между показанной информацией и скрытой). И это алгоритмы действий самых базовых операторов. На разных этапах производства создания персонажей отношения подключаемых модулей к цвету через различные каналы могут выдавать самые различные результаты. Причем

⁶ В некоторых 3D-редакторах обозначается как Diffuse.

помимо расширения цветового спектра от монохромного до общего полного цветового радиуса.

Скажем, в момент риггинга персонажа (настройки виртуального скелета) цветовой диапазон немонохромного типа влияет на распределение физического веса геометрии по отношению к виртуальным костям (Joint). Физический алгоритм распределения отношений костей к геометрии называется «весами»(weights). Оператор weights имеет помимо математической также и визуальную цветовую форму представления, которая работает через полноформатный, а не монохромный цветовой радиус, выраженный через градиент перепада от самого слабого значения действия кости на геометрию до самого сильного.

Глава 2. Мифы современного CGI-сообщества

Проблема современного CGI-сообщества заключается в том, что сам технологический процесс становится порой важнее непосредственно результата. Иными словами, специалист получает большее удовольствие, применяя более изощренные технические новшества, чем непосредственно эстетическая проблематика интеграции математическо-художественного образа в «полотно» того или иного произведения искусства новых медиа.

О данной проблеме также упоминает Лев Манович, один из самых известных исследователей новых медиа. В своей книге: «Язык новых медиа» он утверждает:

«Программное обеспечение стало нашим интерфейсом к миру, к окружающим, к нашей памяти и нашим фантазиям – универсальном языке, на котором мир говорит, и универсальном мотором,двигающим современный мир вперед. Подобно электричеству и двигателю внутреннего сгорания в начале XX века, программное обеспечение стало локомотивом в XXI веке»⁷.

Данная проблематика порождает 2 мифа в современном мире компьютерной графики.

2.1. Миф об идеальном инструменте

Первый – это миф об идеальном инструменте (программном обеспечении).

На протяжении многих лет среди компаний, а также многочисленных адептов пользователей того или иного программного обеспечения не стихают споры о том, что именно выбранный ими когда-то, по какой-то причине инструмент является неоспоримо лучшим во всей индустрии.

⁷Manovich L. The Language of New Media MIT Press, 2001.

Причем, что интересно, в основном все эти споры построены исключительно на технических особенностях того или иного инструмента. Иными словами, предположим, что два резчика по дереву спорят, какая бензопила лучше, или художник, написав определенное произведение искусства, хвастается не своим мастерством, а производителем красок, которыми он пользовался.

По большей части важно осознавать основной пласт теоретического понимания процесса. Если углубляться в то или иное программное обеспечение, то становится понятно, что практически все широко распространенные программные продукты, что представлены на рынке, как правило, являются коммерческими предложениями с набором среднестатистических инструментов, предназначенных для широкого круга специалистов. Нужно понимать, что многие функции, рекламируя которые маркетологи привлекают все новых и новых пользователей, уже минимум 3 года использовались в качестве закрытого программного обеспечения, разработанного той или иной студией для определенных задач. В качестве примера можно взять ситуацию с той же Maya. Поразив всех в 1997 году гибкостью и легкостью работы, тогда еще Alias Maya показала, каким может быть универсальное и удобное для пользователя программное обеспечение.

Глубочайшее заблуждение кроется в том, что многие думают, что до 1997 года вся графика в голливудском кинематографе была лишена нововведений технического характера, которые предлагала Maya. Конечно нет, все эти нововведения, разумеется, в немного искаженном виде все же имели достаточно широкое распространение среди закрытых голливудских студий. Сама же революция, которая произошла в 1997 году с Maya или которая происходит каждый год с анонсами новых версий того или иного программного обеспечения, носит более чем маркетинговый характер и побуждает здоровую конкуренцию среди производителей «массового» программного обеспечения. Тенденция XXI века заключается в том, что будущее за универсальными комбайнами, то есть программами-гигантами,

которые позволяют пользователю в одном приложении максимально пройти весь пайплайн. Стоит отметить, что тенденция это не нова, если не брать в расчет закрытое программное обеспечение студий, то в период 1990-х годов компания Discreet представила на рынке универсальный комбайн DiscreetFlame. Программа поражала своими возможностями. Flame изначально предназначалась как инструмент для монтажа, цветокоррекции, композитинга, 3D-трекинга, ротоскопинга, частиц, эффектов, анимации... Вот далеко не весь список ее умений. На тот момент это было абсолютно уникальное торговое предложение. Позиция маркетологов Discreet была проста: зачем студии использовать отдельные небольшие узкоспециализированные программы, когда их продукт позволял пройти весь пайплайн, не выходя из «него». С таким набором функций Flame должна была вытеснить 80 процентов программного обеспечения с рынка. Но... Вы сейчас о нем слышали?! Так что-же пошло не так? Прежде чем продолжить, стоит оговориться, что Flame до сих используется на некоторых студиях и каждый год нынешний ее владелец, компания Autodesk, пытается оживить этого «Франкенштейна», но доля рынка Flame не превышает 2–5 процентов.

Первая и весьма существенная проблема данного программного обеспечения была в излишней ресурсоемкости, помимо того, что Flame не поддерживала приоритетные на тот момент операционные системы (Windows/MacOs) (работало все это только на Linux/RadHat или SiliconGraphics/IRIX), она еще и требовала колоссальные ресурсы машины, на которой должна быть установлена. И если в случае конкурентов, когда машина была менее мощная, чем на то указывали системные требования, происходило просто проседание производительности, то Flame использовала SystemChecker, и если хоть один пункт ее системных требований не совпадал с машиной, то она просто отказывалась запускаться. Но и это не было основной проблемой этого продукта.

Самая главная проблема Flame была в том, что хоть программа и умела технически выполнять каждую из заявленных операций, то на практике

продукт оказался неудобен. По сути говоря, это был просто склад с кучей функций, весьма плохо отсортированных между собой. Иными словами, чтобы просто изменить цвет на одном кусочке изображения, надо было, пробираясь сквозь дебри всех остальных инструментов смежных профессий, подключать необходимые операторы. Обратив внимание на многочисленные жалобы и недолго подумав, компания приняла решения создать 4 узкоспециализированные программы в комплект к Flame – так появился DiscreteIFFFS (Inferno / Flame/ Flint/ Flare/Smoke).

В этом плане Flame немного облегчил интерфейс, Flare взял на себя сложный композитинг, работу с 3D, а также мощный менеджер проектов.Smoke досталась монтажная часть, Flint получил систему ротоскопии и трекинг, Inferno работу с метаданными и процедурные эффекты, при этом сам Flame также оставил себе усредненную работу со всеми этими функциями. Позже они оторвали от Flame систему цветокоррекции, назвав Lustre, а элемент организации проектов Flare перерос в комбайн серверно-сетевого управления Burn.

Иными словами, компания пошла в разрез уникальности их изначального торгового предложения. Сначала она собрала все возможности отдельных программ в одну, громко заявив, что за универсальностью будущее, а потом, когда большинство пользователей не смогли разобраться в излишнем нагромождении функций, разработчики опять разделили возможности Flame на отдельные программные пакеты.

Стоит также сказать, что компания до сих пытается протолкнуть весь IFFFS на рынок массового программного обеспечения домашних платформ. В 2010-м, запустив мощную рекламную кампанию, что Smoke адаптирован под Mac, правда, не упомянув, что с тем же SystemChecker и только на топовые конфигурации MacPro. Однако особой популярностью Smoke это не принесло. Спустя 3 года, в последней попытке популяризации SmokeAutodesk приняла решение убрать ограничение мощностей, и с версии

2013 любой пользователь Macintosh мог наслаждаться творчеством на платформе Smoke... Но зачем ему это было нужно – непонятно. Даже эта последняя попытка найти общий язык с массовыми специалистами не принесла Autodesk ничего хорошего. От того, что теперь любой желающий мог запустить Smoke на своей машине, сама программа удобнее не стала.

И такова тенденция не только на рынке графического узкоспециализированного программного обеспечения. Фактически такое понятие, как узкоспециализированное ПО, снова уходит в прошлое. Компании сейчас снова нацелены на расширение своих основных продуктов. Причины этого кроются сразу в нескольких вещах.

2.2. Узкоспециализированные решения программного обеспечения

Первое – это интеграция усовершенствований в уже привычный графический интерфейс, что выражается, прежде всего, в удобстве взаимодействия между специалистом и программным обеспечением. Каким бы хорошим по функционалу не было стороннее программное обеспечение, пользователю нужно время, чтобы привыкнуть к его интерфейсу и особенностям работы. Это и плюс, и минус. Второе, как ни странно, это финансовая сторона вопроса. Компаниям или пользователю финансово выгоднее заплатить сразу за комбайн широкого спектра, чем докупать отдельные функции других производителей за отдельные деньги. Подобную ситуацию можно рассмотреть на примере трех приоритетных программных продуктов.

ToroGun – система создание ретопологии.

UvLayoutPro – создание UV-развертки.

Vray – система визуализации.

ToroGun – одно из лучших приложений по созданию топологии. Произведенный студией PixelMachine в 2009 году, ToroGun сразу начал завоевывать рынок, предоставляя пользователю передовые инструменты для ретопологии созданных при помощи других программных продуктов моделей.

По сравнению с другими трехмерными пакетами ToroGun имел неоспоримое преимущество. Это очень простой интерфейс, мощный инструментарий и, что важно, с очень высокой оптимизацией.

Даже сейчас в Maya-2018 нетпредпросмотра высоко детализированной топологии (HiPolytoLowPolyProjection) до момента физического запекания карты. И в тех случаях, когда производительность ToroGun выдает стабильную скорость работы, та же Maya проседает в 2, а то и в 4 раза.

UvLayoutPro – передовой инструмент для создания UV-разверток.

Инструменты Uv, как правило, всегда были достаточно скудными в арсенале сложных трехмерных комбайнов, т.к. большинство задач оптимизируются вокруг относительно простой и примитивной геометрии. Сама постановка условия перед этим инструментом строится на реализации UV среднестатистического пользователя, что практически полностью исключает выполнение и реализацию нестандартных задач сложной геометрии, созданной не в рамках трехмерного комбайна.

Именно поэтому такая узкоспециализированная программа, как UVLayoutPro дает возможность пользователю сосредоточиться исключительно на UV путем глобального переосмысления представления о манипуляции данной операции, вплоть до того, чтобы выйти за рамки двухмерного представления и дать возможность пользователю снимать карты прямо на трехмерной модели. А такого и по сей день практически нигде не встретишь.

И наконец, неоспоримый лидер рынка визуализации – Vray от компании ChaosGroup. Увидевший свет в начале 2000-х, Vray быстро приобрел

популярность, как стандарт визуализации разных областей применения графики. Было это связано даже не с тем, что Vray выдавал потрясающий результат, а как раз наоборот: даже близкого результата при тех же усилиях не могли дать конкуренты, что уж говорить о встроенных рендерах интегрируемых теми же 3DSMax или Maya.

Эти три узкоспециализированные программы возникли в результате общего массового нагромождения функционала основных трехмерных пакетов.

Производитель массового ПО, каким и является любой общий трехмерный пакет общего пользования (Maya, 3DsMax, Cinema 4D, Houdini), вынужден сосредоточиться не на узкоспециализированных вещах, а на дополнении широкого спектра общего функционала. Так или иначе, если и представить, что эти функции были бы реализованы, то пользователю прежде чем начать просто работать с ними, пришлось бы хотя бы базово переучиться на новый инструмент, потому что даже получив доступ к ним, нужно пробираться через дебри графического интерфейса того или иного программного продукта.

Начиная с 2016-версии, Autodesk кардинально продолжил развивать Maya во всех узкоспециализированных областях.

С момента приобретения Maya компанией Autodesk закрепленный успех под управлением Alias не претерпел дальнейшего развития. Еще вчера, являющаяся неоспоримым лидером рынка, Maya стала сдавать позиции под натиском конкурентов.

Помимо трех вышеперечисленных систем не стоит забывать, что есть и менее популярные комбайны от других производителей.

2.3. Ассоциативная подмена профессий программным обеспечением

Тенденция конца XX века сказалась еще и в том, что технические инструменты стали эстетически ассоциироваться с профессиями.

Моушн-дизайн практически занимал Maxon с их ПО Cinema 4D.

Если необходимы сложные визуальные эффекты, то вам к Houdini.

Mayasmee былой славой все еще оставила себе пьедестал анимации и риггинга.

3DsMax отошло поприще статичного и интерьерного моделирования.

В период с 2015 по 2018-й версию Autodesk решили разом усовершенствовать узкоспециализированные системы. Стояла задача, как сделать систему удобной для пользователя не просто технически, а еще и эргономически.

Чтобы достичь этого, и был максимально переработан интерфейс взаимодействия с пользователем. Появилась система Toolkit. Toolkit представляет собой не просто функционал инструментария, а целую мини-программу внутри основной программы, которая позволяла пользователю максимально сосредоточиться не на массивном программном продукте, а

только на необходимых функциях. Работало это все намного лучше, чем до этого, и это, пожалуй, было наилучшее решение для упрощения среды. Мини-TopoGun – интеграция с ModelingToolKit, мини-UvLayoutPro-UvToolkit со своим внутренним интерфейсом были интегрированы в основное тело Maya. Интерфейс каждого модуля также предлагал текстовое описание внутри модуля иза что какая функция отвечает.

Заброшенный MentalRay заменил купленный у SonyArnoldRender. Тем самым, это создает хоть какую-то конкуренцию Vray. И да, в основных вещах что Vray, что UvLayout, что тот же TopoGun будут куда лучше, чем то, что предоставляют пусть и переработанные гиганты.

Выживают сильнейшие. Каким бы ни был хорош TopoGun, но без поддержки и развития он постепенно сойдет на нет.

Однако, в современных версиях MacOSX (начиная с HighSierra) уже прекращена поддержка 32-битных программных продуктов. Так что отказ от

ТороGun в пользу той же Maya или какого-то другого программного обеспечения – это всего лишь вопрос времени.

Нетрудно догадаться, что следующее крупное обновление Maya-2020 будет нацелено на устранение основного конкурента Houdini и, видимо, будет полностью переработан блок симуляции и динамики.

Время идет, и компании, опираясь на не совсем удачный опыт прошлых лет, задумываются не просто о технических совершенствованиях разрабатываемого им ПО. Сейчас самая основная задача, которая стоит перед разработчиками, – это не привлечь новых пользователей путем дополнения функциональности, а не отпугнуть старых. Впервые на передний план выходят не технические совершенствования того или иного продукта, а эргономика и удобство пользователя.

В качестве примера можно взять две попытки устроить «маленькую революцию» в разных областях CGI.

Первая «произошла» в области монтажных систем, вторая – в области цифрового создания персонажей.

Глава 3. Революционные решения развития эстетики и технологии цифрового программного обеспечения

3.1. «Маленькая революция» систем видеомонтажа

Начиная с конца 1970-х годов, начали появляться первые цифровые монтажные системы, представлявшие собой огромные комплексы, а к концу 1980-х цифровая революция систем видеомонтажа стала доступна на персональных компьютерах. Так появился лидер рынка на тот момент и по сей день – AVID.

Но так было не всегда: в начале 2000-х Apple переманила пару специалистов из Avid и Adobe, и путем совместного труда на свет появился FinalCut. FinalCut – единственная монтажная система, составлявшая на момент своего активного распространения реальную конкуренцию Avid в Голливуде. К 2008 году половина голливудских фильмов была сделана с использованием FinalCut. Основное преимущество FinalCutPro перед тем же Avid было как раз удобство и эргономика. Каким бы ни был удобным Avid, он стал первопроходцем на рынке цифровых систем, выстроив неоспоримые алгоритмы и принципы работы цифровых NLE (нелинейных систем монтажа). Неважно, какой программой вы пользуетесь – Avid, FinalCut, Premiere, Edius, – вы все равно найдете там такие каноничные вещи, как, скажем, трековая таймлайн, окно просмотра эксплорера, окно просмотра таймлайна и т. д.

В 2011 году пользователи FinalCutPro предвкушали обещанное обновление.

Каково было удивление, когда в новой версии создатели решили полностью уйти от структуры трековой таймлинии, заменив ее на «магнитическую», которая использует не привычную линейную иерархию треков, как это было до этого, а Connection-клипы, где главной особенностью становится некая иерархическая привязка «родительского» клипа к

«детскому». Система трековой таймлинии была полностью переработана. Классическо-иерархическое представление дорожек и треков сошло на нет в пользу Storyline, которая в свою очередь представляла собой новую неиерархическую систему PrimaryandSecondaryStoryline. Каноны, заданные Avid в 1989-м (которые, в свою очередь, были заимствованы у других профессиональных комплексных пакетов цифрового кинопроизводства), канули в лету в один миг. Apple сняла все каноничные ограничения с любой монтажной системы, выходявшей до этого. Ни Divider (разделителя треков), ни приоритета дорожек, ни самих дорожек в новой системе FinalCutPro, которую также переименовали в FinalCutProX, не было. Хотите ставить аудио над видео – пожалуйста, перетащить первую половину фильма в одно касание в любое место – да запросто. Это была попытка отойти от всей условности философии цифрового монтажа, дав пользователю абсолютно новую среду.

Данное нововведение не только не привело к популяризации FinalCutX среди домашнего использования, но и сделало невозможным «старым» пользователям работать в ней. Компания фактически заставляла их изучать программу заново.

Помимо всего, в ней отсутствовали критические для любого специалиста функции, которые, что самое интересное, присутствовали в старом FinalCut. На момент релиза в ней отсутствовал экспорт цифровых монтажных листов (EDL,XSML),многокамерный монтаж же вообще был добавлен спустя пару лет.

Стоит отметить, что классический FinalCut имел расширенную версиюFinalCutStudio, котораявключалавсебятакиепрограммы,какAppleMotion, AppleColor,DVDStudioPro,SoundtrackPro, Compressor.В отличие от того же IFFFS, FCP (FinalCutPro)Studio строго и узкоспециализированно разделял свои продукты. Например, в программе Color пользователь не мог делать ничего другого, кроме цветокоррекции. Его не отвлекало нагромождение

функций, которые так или иначе не были связаны с основной задачей пакета: коррективкой цвета.

В новой версии FinalCutProX от ряда программ было решено отказаться в силу общей унификации. В итоге Color, SoundtrackPro, DVDStudioPro были переработаны и интегрированы в новый интерфейс самого FinalCutProX. Color, который всегда представлял собой топовую систему работы с цветом, превратился в обычный фильтр, делающий простую регулировку цвета по трем каналам, что сразу поставило крест на какой-либо профессиональной работе.

Изначально хорошая идея развалилась из-за того, что создатели просто не поняли, для какой целевой аудитории сделали свое новое детище.

Для любителей он по-прежнему был сложен, а для профессионалов это обновление исключало работу по привычной и отлаженной системе пайплайна.

Стоит сказать, что на фоне провала FinalCutProX конкуренты, в лице тех же Adobe со своим Premiere и Avid с MediaComposer, начали сразу же переориентировать рынок под себя. Они понимали, что их главный конкурент совершил большой промах и огромная целевая аудитория пользователей старого FinalCutPro фактически осталась без главного инструмента. Ведь вопрос устаревания FinalCut— это был лишь вопрос времени.

Adobe сразу начала выстраивать рекламную компанию, оперируя тем, что Premiere всегда был и остается решением для профессионалов. После чего маркетологами была разработана рекламная компания, которая гласила: «Сдайте FinalCut и получите скидку на Premiere» или «PremierePro – всегда был и остается Pro». Фактически им этого было и достаточно, ведь именно Premiere больше всего схож интерфейсом с классическим FinalCutPro.

Avid просто не мог сделать также, сама философия программы всегда была не дружелюбна к массовому пользователю —это есть и главное

преимущество, и недостаток. То мировоззрение профессиональной среды, выработанное годами, всегда являлось неким камнем преткновения между обычными пользователями и их инструментом. В целях заполучить потерянную аудиторию FCPAvid пошел на кардинальные меры в лице упрощения MediaComposer, чтобы пользователям FinalCutPro было легче на него перейти.

Начиная с версии 5, в MediaComposer было добавлено интерактивное меню выбора режима монтажа, что-то подобное было и в оригинальном FinalCutPro. На презентации Avid при показе этой функции пользователи MediaComposer тоже начали возмущаться, что Avid со своим MediaComposer идет по тому же пути, что и Apple с FinalCutProX. Даже стали появляться баннеры в интернете MediaComposerX, где приставка X стала нарицательным слова «любительский». Понимая, что помимо того, что надо привлечь аудиторию FinalCut, надо не потерять еще и собственную, Avid сделал панель автоматического режима отключаемой. Не хотите, чтобы программа сама выбирала режим выбора клипа на таймлиннии, не надо.

3.2. «Маленькая революция» принципов создания цифровых персонажей

Удачным примером можно считать решение компании Pixologic, об интегрировании стандартного HardSurface-моделинга в структуру их основного программного продукта Zbrush. Как и в случае с FinalCutPro, тут стояла похожая задача – внедрить совершенно другую систему в уже существующую. Вопрос стоял в том, как это сделать. Как интегрировать среду классического трехмерного моделирования в 2.5-мерную программу цифрового скульптинга? Учитывая то, что алгоритмы этих двух сред взаимоисключают друг друга.

Разработчики понимали, что не так важны возможности самого нового инструмента, как его интеграция в основную среду.

Создатели сразу решили не переносить основные элементы представления классического трехмерного моделирования. Речь даже не об

инструментарии, а о глобальном представлении. Такие неотъемлемые атрибуты любого классического 3D-редактора, как представление модели в неортогональных видах (фронтального(frontview)/вида сверху (top view)/вида сбоку(side view)). От них было решено отказаться, чтобы не переориентировать основную среду 2.5-мерного представления пространства. Это решение было принято, чтобы не отпугнуть старых пользователей. Zbrush останется Zbrush. Теперь, когда с приоритетом того, чем можно пожертвовать ради новых функций, закончено, то есть – ничем, как быть с самими функциями? Как интегрировать огромный функционал взаимодействия с геометрией классического трехмерного редактора так, чтобы и он не нарушил основную экосистему программы? Решение не заставило себя долго ждать. Инструмент классического моделирования в Zbrush получил название ZModeler.Zmodeler не более чем кисть, и вся эта система интегрирована и интерфейс Zbrush таким образом, что если вам не нужен функционал по моделированию HardSurface, то вы даже не заметите нововведений в программе, но если он вам понадобился, то вы просто достаете нужную кисточку и... вуаля.

На данный момент мы можем наблюдать весьма интересную тенденцию: в современном мире программного обеспечения происходит некий реверс. Как говорится, с чего начали, тем и закончили.

Причем речь не идет просто в сторону технического внедрения инструментария...

Тенденция на многофункциональные комбайны снова стала актуальной, но если раньше сам принцип добавления функций одного программного продукта в другой выстраивался с учетом приоритета значимости, то сейчас можно наблюдать нарушение и этого канона. Программные продукты стали интегрировать новые возможности не по принципу добавления вспомогательных функций к основной. В этом плане первая попытка интеграции невспомогательных инструментов, которые должны облегчить решение многих задач. Речь идет о внедрении и

интеграции совершенно разноплановых экосистем друг в друга, как ToolKit в случае с Autodesk.

Если раньше интеграция происходила по принципу: как к основному рабочему процессу подключить и добавить узкоспециализированные профессиональные модули второго плана, то теперь ситуация стала другой.

Компания Blackmagic в 2015 году начала интегрировать монтажную систему в свой профессиональный редактор DavinciResolve. Да, да, вы не ослышались. Не цветокорректор в системе монтажа, а полноценный монтаж в системе цветокоррекции.

Можно сказать, что подобный опыт опять же был в 1990-х с тем же IFFFS, но Blackmagic по-другому подошли к задаче интеграции, нежели их предшественники. Они не стали никак вмешивать новые и канонично монтажные операции в привычную систему профессиональной цветокоррекции. Все нововведения так же, как и в случае с Zbrush и его ZModeler, были представлены в отдельном блоке.

AdobePremiere пошел дальше: в 2014-й версии Adobe стал внедрять в него элементы из своей другой программы – AdobeAfterEffects. Появление таких канонично композерских функций, как трекинг, вызвало весьма неоднозначную реакцию. Классические пользователи Premiere, то есть режиссеры монтажа, не видели изменений, которые напрямую касались бы работы их отделов. На самом деле все кардинальные обновления, добавленные в Premiere за последние 5 лет, — это перенесенные элементы из AfterEffects. Среди специалистов данного программного обеспечения вообще бытует мнение, что лучшее, что есть в современном Premiere, это части, добавленные из AfterEffects и не более. Старая платформа, весьма скудная библиотека эффектов и переходов по сравнению с тем же Avid и FinalCutPro, причем обеих версий– и классической, и новой X.

Система взаимоинтеграции разносторонних инструментов нашла свое видение и в британской компании TheFoundry, в одном из своих программных продуктов Nuke, являющимся неоспоримым лидером в области

композитинга. В 2015 году TheFoundry представила более расширенную версию своего программного продукта Nuke, получившее название NukeStudio. В ней также, как и в Davinci, была сделана монтажная система, правда, намного менее удобная. Стоит отметить, что TheFoundry с 2009 года разделила Nuke на две рабочие версии –Nuke и NukeX, – главное отличие одной от другой являлось наличие в NukeX 3D-трекинга моделей, расширенного композитинга и ротоскопии, и расширенных базовых плагинов. Данное разделение было оправдано тем, что функция трехмерного трекинга является частью производственного процесса классического цифрового композитинга, инструментом чего и является Nuke. Что не скажешь о NukeStudio, получившем название NukeS, позиционировавшем себя как самая расширенная версия Nuke. В отличие от NukeX, в NukeS были добавлены инструменты, не относящиеся к профессии композера: монтаж, цветокоррекция и работа со звуковым мастерингом.

Можно вспомнить, что и у Maya тоже были версии своего программного продукта.

MayaPersonal – бесплатный пакет для некоммерческого использования, с ограничениями по сравнению с полной версией, в том числе и на размер (разрешение) выводимых изображений.

MayaComplete –полная версия программы, включающая такие блоки, как Modeling и Animation, но в которой отсутствуют режимы физических симуляций и расширенного рендеринга⁸.

Полный комплект функций можно было найти только в MayaUnlimited.

MayaUltimate – самый полный и самый дорогостоящий пакет. Содержит расширения Hair, Fur, Maya Muscule, Fluid Effects, Cloth, а также систему полного рендеринга и физических симуляций.

⁸СергейЦыпцын. «Понимая Maya».

После покупки фирмы AliasAutodesk в 2009 году собрала все модели в единый пакет. Даже сейчас в современных версиях Maya еще можно найти отголоски старой системы, выраженные в интерактивном меню выбора режима работы.

Глава 4. Миф об универсальной CGI-профессии

4.1. Пайплайн и его прохождение на киноигровом проекте

Каким бы ни было производство продукта новых медиа – это все равно производство, т. е, конвейер, где главная задача – выпустить готовый «продукт». Эта задача очень четко распределена между различными отделами. И в данном случае ни о какой универсальности или «эталонности», превозвышающей одну профессию над другой, и речи быть не может. Приведу несложный пример. Предположим, перед командой стоит задача сделать трехмерного антропоморфного персонажа для анимационно-игрового проекта. Если разделять весь пайплайн на поэтапные периоды, то получается следующая схематическая взаимосвязь специалистов:

в идеале все начинается с образа, по сути, это является самой сложной задачей всего производства, т. к. именно эта задача становится отправной точкой. Структура работы всех последующих отделов (их уровень сложности и особенности) строятся, исключительно исходя от придуманного образа.

Как правило, на этапе раннего концентрирования образ представляет краткое литературное описание. За это отвечают сценаристы, которые помимо общего сценария игры пишут краткий литературный синопсис с характеристиками каждого персонажа. Если уходить от эстетической проблематики, важности драматургического сценария, то, по сути, для последующих отделов, которые занимаются именно персонажным пайплайном, важен только краткий литературный синопсис.

После того как описание персонажа готово, концепт-художники делают первые варианты эскизов (быстрых набросков) данного героя.

После этого, когда базовый образ утвержден, концепт-художники начинают более детальную проработку данного образа.

Для последующего пайплайна из всех концептов, по-сути, важны только так называемые продакшн-референсы. Эти референсы изображают персонажа в трех основных видах: фас, профиль и три четверти.

Создание продакшн-референсов для 2D- и 3D-персонажей немного отличается.

4.2. Основные отличия продакшн-референсов для 2D- и 3D-персонажей

Первое, что нужно учитывать, это то, что не все двухмерные персонажи будут смотреться в соответствии с изначальным образом при добавлении третьего измерения – т. е. физической глубины. Физическая глубина, ось «Z», помимо непосредственного преимущества из-за добавления третьего измерения, для большинства образов – как «палка о двух концах». А именно: излишняя информация фактически разрушает оригинальный образ того или иного персонажа.

В первую очередь, это отражается на цветовом решении концептов. Если концепт предназначен для трехмерной игры, то художник должен апеллировать бóльшим значением полутонов, тем самым выходя за рамки цветовой монотонности «сплошных» цветов, которыми можно описать стандартные двухмерные объекты.

Второе: это проблематика восприятия данного образа. Не все двухмерные персонажи сохраняют эстетику оригинального образа после приобретения информации о глубине.

Добавление полутонов предполагает добавление цветовой информации. И тут возникает другая проблема: какую именно информацию

нужно добавить? Художник фактически балансирует на грани между стилизацией и реализмом.

Компиляцией данного утверждения считается то, что с добавлением третьего измерения из круга все равно выйдет шар, из квадрата куб и т. д. Если проводить данную аналогию на примере американского игрового кинематографа, то в качестве примера можно рассмотреть фильм Джоната Лебисмана «Черепашки ниндзя» (*англ.* TeenageMutantNinjaTurtles) 2014 года. В оригинальном мультсериале 1987 года использовалась классическая рисованная анимация, тем самым закрепив оригинальный образ главных героев – четырех черепашек – с конкретным художественным изображением, которое благодаря оглушительному успеху мультсериала во всем мире стало каноничным.

А теперь давайте проанализируем на этом примере, как третье измерение Z изменило главных героев некогда детского мультсериала.

Если рассматривать двухмерный антропоморфный образ, каким, собственно, и являются черепашки, можно их представить в виде простейших двухмерных графических примитивов: где голова, нос, глаза, туловище и другие художественные элементы образа представлены в виде примитивных кругообразных форм. Ранее мы говорили о том, что добавление полутонов приводит постепенному усложнению самой формы примитива, и это получается уже не круговая форма силуэта, а более сложная геометрия.

Поэтому нет ничего удивительного, что картина «Черепашки-мутанты ниндзя» образца 2014-го и тем более последующий сиквел «Черепашки ниндзя 2» (*TeenageMutantNinjaTurtles: Outoftheshadows*) 2016 года не увенчались большим успехом ни у критиков, ни у поклонников оригинальной саги.

Помимо неадаптированной графики стоит упомянуть, что все же основной провал фильма Лебисмана связан с тем, что сценаристы картины Джош Аппелбаум, Андре Немец и Ивен Догерти так и не поняли, на какую целевую аудиторию рассчитан их сценарий. Они пытались сделать из

достаточно детской истории боевик, но вот проблема то в том, что до мощного боевика картина воспринималась все еще как детский фильм, а для мультика как раз наоборот, излишне жестокой.

Стоит сказать, что фильм Либсмана – не первая попытка разрушить канон. Еще в далеком 1990 году вышла первая экранизация мультсериала от Стива Бэрона «Черепашки-Ниндзя» (Teenage Mutant Ninja Turtles). Тогда для создания черепашек были задействованы актеры, переодетые в костюмы культовых героев. Но не все так просто, как кажется. Актеры служили, по сути, силуэтом для аниматронической конструкции лицевой анимации, которой с помощью дистанционного управления координировала целая команда специалистов по аниматронике, присутствующая на площадке.

Относительно удачным примером переноса двухмерного образа непосредственно в трехмерное цифровое пространство можно считать картину «Черепашки мутанты ниндзя» (TMNT) 2007 года режиссера Кевина Манро. Вопрос заключался в следующем: почему при практически аналогичной технологии производства картина 2007 года выглядела более органично по сравнению с оригиналом, нежели фильм 2014 года. Ответ как раз кроется не столько в процессе создания модели путем строения, получения топологии, UV, хотя стоит отметить, что для этого анимационного фильма в процессе моделинга персонажей за основу были взяты как раз наиболее примитивные формы оригинала. Но все равно за счет добавления физической оси Z, как было написано выше, форма хоть все же и имела похожие очертания, но была изменена. Так как на этапе рендера грани трехмерного объекта будут отражаться и затенять друг друга, создавая дополнительную цветовую информацию. Ответ как раз кроется не только в строении модели, сколько в процессе, который максимально завязан на работе цвета, а именно методе текстурирования и шейдинга.

Чтобы продолжить анализ давайте рассмотрим, что такое шейдер. Шейдер, если говорить простым языком, это материал, но не простой: в качестве примера можно рассмотреть кувшин с молоком. У кувшина с

молоком есть две составляющие — это сам кувшин и молоко, которое в него налито. Вопрос кроется в том, что может существовать друг без друга. Кувшин без молока или молоко без кувшина. Разумеется, правильным ответом будет кувшин. Так как молоку нужна сопряженная поверхность, какой, собственно, и является кувшин. Нужно это для того, чтобы сохранить свои физические составляющие, скажем, форму. И это правило существования применяется в нашем мире, но перенося модель кувшина и молока в мир компьютерной графики, шейдером называется именно молоко, сформировавшее вид кувшина, без самого кувшина. Так как на месте молока может быть совершенно любой физический материал, а на месте кувшина любая трехмерная геометрия. Шейдеры могут иметь помимо цветовой информации, которая относится к каналу цвета, еще многочисленные каналы, и чем каналов больше, тем реалистичнее выглядит шейдер и тем самым – трехмерная модель. В фильме 2014 года специалистами очень долго создавались различные шейдеры, чтобы их модели выглядели реалистично. Это же было и основное отличие от фильма 2007 года, где на той же по структуре и технологии трехмерной модели вместо трехмерно сложных материалов были использованы двух с половиной мерный шейдера (2.5DShaders), практически однотонного цвета с практически полностью отсутствующим градиентом. Тем самым снизив уровень информации оси Z до минимума.

Именно поэтому черепашек ниндзя 2007 года можно вполне считать максимально близкими к оригинальному мультсериалу 1990 года с использованием 2.5-мерных шейдеров. Главное – перенести двухмерный образ в трехмерное пространство с минимальными потерями.

Заключение

На протяжении последних 30 лет у работников профессиональных отраслей CGI-индустрии произошли изменения восприятия программного обеспечения. Технологическое вмешательство современного прогресса изменяет ядро исконно гуманитарно-творческих профессий, забирая их основу в качестве базиса и создавая абсолютно новый симбиоз сочетания творческой профессии и технологической подноготной в целом.

Меняется сама эстетика восприятия не только инструмента (софта), а еще и самоопределения пользователя. Если раньше программное обеспечение опиралось исключительно на технический функционал, то теперь это кажется недостаточным ввиду неподготовленности массового профориентированного пользователя к этому инструменту. Гуманитарная составляющая начала напрямую влиять на эстетику оболочки рабочей среды. Прикладываю к набору технических способов реализации того или иного процесса, еще и эстетически удобные способы взаимодействия пользователя с инструментом.

Современные способы выражения творческих идей стали очень зависимы от многочисленных технических инструментов их реализации.

Различные компании предлагают многочисленные инструменты для решения тех или иных задач творческого направления. Техническая составляющая в виде программно-математических комплексов постепенно интегрируется с сугубо творческими направлениями различных проявлений искусства. И если раньше эти сферы практически не имели точек пересечения, то в современном мире постепенно стирается граница жесткого разделения между гуманитарными и прикладными профессиями. Так как современная тенденция переосмысливает каждое направление деятельности в некий симбиоз, где от творческого остается зачастую только постановка той или иной задачи, а от технического требуется только набор инструментов для его воплощения.

Список использованной литературы

1. Воль Майкл. Техники монтажа в FINAL CUT PRO. — Издательство АСТ. — 454 с.
2. Дмытрык Эдуард (Edward Dmytryk). О монтаже фильма (On Film Editing). — Издательство Focal Press, 1984.
3. Кэмпбэлл Джозеф (Joseph Campbell). Герой тысячи лиц (The Hero with a Thousand Faces). — Издательство Princeton University Press.
4. Manovich L. Language of new media. — MIT Press, 2001. — 307 p.
5. Мараффи Крис. Создание персонажей в Maya. — Издательство «Вильямс», 2004. — 442 с.
6. Марч Уолтер (Walter Murch). В мгновение ока (In the Blink of an Eye). — Издательство Silman-James Press, 2001.
7. Мелби Калев. Религия черной водолазки. — Издательство: «Э».
8. Олдхэм Габриэлла (Gabriella Oldham). Первый монтаж: разговоры с кино монтажерами (First Cut: Conversations With Film Editors). — Издательство University of California Press, 1992.
9. Чанг Джастин. Профессия режиссер киномонтажа. — Издательство «Рипол Классик»: Москва, 2014.
10. Цыпцын С. Понимая Maya. — М.: Арт Хаус медиа, 2006. — 703 с.
11. Янг Джеффри С., Саймон Вильям Л. Икона Стив Джобс. — Издательство «Эксмо». — 464 с. // электронный доступ: http://www.vgik.info/science/bulletin/recommendations_of_the_authors_of_the_journal_vestnik_cinematography_quot.php, дата обращения: 01.04.2018.

Фильмография

1. Faces and Body parts – Utah Head(США),1974г.
Режиссер: Fred Parke
2. «Черепашкининдзя» (*англ.* TeenageMutantNinjaTurtles), 2014 г.
Режиссер: Джонатан Лебисман
Сценарий: Джош Аппелбаум, Андре Немец, Ивен Догерти
Оператор: Лула Карвальо
В ролях: Меган Фокс, Уилл Арнетт, Уильям Фихтер, Алан Ричсон
3. «Черепашки ниндзя» (*англ.* TMNT), 2007 г.
Режиссер: Кевин Манро
Сценарий: Кевин Манро, Питер Лэрд, Кевин Истмен
Оператор: Стефан Дж. Ламли
В ролях: Крис Эванс, Сара Мишель Геллар, Мако, Кевин Смит
4. «Черепашки-Ниндзя» (*англ.* Teenage Mutant Ninja Turtles), 1990 г.
Режиссер: Стив Бэррон
Сценарий: Бобби Хербек, Толд В. Лаген, Кевин Истмен
Оператор: Джон Феннер
В ролях: Джудит Хоаг, Элиас Котеас, Джош Пейс, Дэвид Форман