

REFERENCES

1. *Boden M. A., Edmonds, E. A.* What is generative art? Digital Creativity, 2009. № 20 (1–2). PP. 21–46.
2. *Lyu Yanru., Wang Xinxin., Lin Rungtai., Wu Jun.* Communication in Human–AI Co-Creation: Perceptual Analysis of Paintings Generated by Text-to-Image System. // Applied Sciences, 2022. № 12 (22). PP. 1–19.
3. *Oppenlaender J.* A Taxonomy of Prompt Modifiers for Text-To-Image Generation. [Электронный ресурс]. URL: <https://arxiv.org/abs/2204.13988> (дата обращения: 26.03. 2023).

ИНСТРУМЕНТЫ СОЗДАНИЯ 3D ГРАФИКИ ДЛЯ ВЕБ-ПРИЛОЖЕНИЙ НА МОБИЛЬНОЙ ПЛАТФОРМЕ

3D GRAPHICS CREATION TOOLS FOR MOBILE WEB-APPLICATIONS

В. В. КАСЬЯНИК, И. Р. ЛУКЬЯНОВИЧ
V. KASYANIK, I. LUKYANOVICH

Белорусский государственный университет
Минск, Беларусь
Belarusian State University
Minsk, Belarus

e-mail: valerialis995@gmail.com, lukianinna12345@gmail.com

Рассмотрены особенности создания трехмерной графики для мобильных приложений и игр на примере трехмерных моделей веб-каталогов и игровых моделей средствами *Unity*, *Blender* и *Adobe Photoshop*. Проанализированы способы интегрирования таких моделей в веб-приложения на основе *API WebGL*, библиотеки *Three.js*, а также программы *Verge3D*.

Ключевые слова: 3D-моделирование; мобильное приложение; API визуализации 3D в браузере; технология WebGL; постобработка в Adobe Photoshop.

The features of creating three-dimensional graphics for mobile applications and games are considered on the example of three-dimensional

models of web catalogs and game models using *Unity*, *Blender* and *Adobe Photoshop*. The ways of integrating such models into web applications based on the *WebGL API*, the *Three.js* library, and the *Verge3D* program are analyzed.

Keywords: 3D modeling; mobile app; 3D rendering API in the browser; WebGL technology; post-processing in Adobe Photoshop.

Долгое время визуализация трехмерных изображений во всемирной паутине осуществлялась, главным образом, посредством внешних плагинов, специальных форматов, апплетов или статических изображений, имитирующих трехмерность. Такой подход зачастую противоречил требованиям безопасности программных и аппаратных клиентских ресурсов¹. С появлением элемента *<canvas>* впервые был получен доступ к пикселю внутри документа, отображаемого браузером, а значит возможность использовать практически все алгоритмические и программные инструменты трехмерного моделирования, адаптировать и реализовывать своеобразный геометрический конвейер для мобильных платформ.

Создание трехмерной графики для компьютерных игр и адаптивных веб-приложений имеет ряд особенностей. Интегрирование таких изображений в веб-документы также предполагает применения специального инструментария и техник. Рассмотрим некоторые из них на примере разработки дома и объектов игры-симулятора. Прорабатываются две версии контента «Алхимик» и «Травница»).

НЕКОТОРЫЕ ОСОБЕННОСТИ РЕДАКТОРА BLENDER.

Для создания трехмерной компьютерной графики используется свободно распространяемый кросс-платформенный 3D редактор *Blender*. Он содержит средства моделирования, скульптинга, анимации, симуляции, рендеринга, пост-обработки и монтажа видео со звуком, компоновки с помощью узлов, а также создания 2D-анимаций. Программное средство устойчиво развивается и имеет и техническую поддержку.

Важнейшей особенностью редактора для этого проекта является наличие инструмента *Композитинг (Node Compositing)*, объединяющего визуальные элементы из разных источников в единое изобра-

¹ GCFGlobal. 2023. Установка и обновление плагинов для браузера [Электронный ресурс] – Режим доступа: https://edu.gcfglobal.org/en/tr_ru-internet-safety/-plugins/1/#. – Дата доступа: 18.03.2023.

жение. Для этого в Blender существуют *Ноды* – узлы, которые выполняют определенные операции и имеют один или несколько выходов и входов. Чаще всего они используются для продвинутой настройки материалов или сцены. Ноды соединяются между собой обычно в соответствии с цветами сокетов. Пример такого структурированного объекта показан на (Прил. 20, рис. 1). Существует три основных типа нодов: входные, обработки и выходные. Входные ноды (*Input Nodes*) предоставляют информацию, но сами не имеют никаких входов. Ноды обработки (*Processing Nodes*) имеют входы, могут преобразовывать поступающую на них информацию, а также имеют выходы. Выходные ноды (*Output Nodes*) принимают результат работы двух вышеперечисленных типов нодов и выводят финальное изображение.

Blender может оперировать системами частиц, контролировать веса отдельных частей при нанесении текстуры, применять направляющие при анимации и использовать внешние силы. Кроме того, в программе имеется симулятор флюидов, который позволяет создавать эффекты текучих тел, таких как дым или жидкости. В режиме реального времени пользователь может просчитывать физические задачи, например моделировать поведение мягких тел. Программа имеет обширную базу документации и вспомогательных расширений – аддонов. *Аддоны* разрабатываются пользователями и разработчиками Blender.

Важнейшей характеристикой игровой модели для адаптивных приложений является ее размер («вес»), на уменьшение которого при сохранении требуемого качества направлена, в частности, оптимизация модели. Blender имеет, наверное самые удобные инструменты для оптимизации модели. В частности, для работы с UV-разверткой, предусмотрен автоматизированный процесс развертки, нанесение размеров модели и развертка вручную. Реализованы все виды запекания текстур: свет, цвет, шероховатости, карты нормали. При запекании для уменьшения веса модели на текстуру была перенесена мелкая геометрия. Игровые объекты подвергались ретопологии – переработка сетки в Blender идет поверх модели отдельным объектом, что очень удобно. Blender-файлы можно открывать в *Unity*.

В качестве референса для разработки дома использовалось следующее изображение (Прил. 20, рис. 2). Для игры он было адаптирован под дом алхимика (травницы) следующим образом:

- добавлены атрибуты для зелий (пустые и заполненные сосуды);
- в интерьере размещено больше цветов, растений и т. п.

- плита в правой части комнаты переработана в глиняную печь;
- уменьшено количество предметов быта для акцента на «растительной части» интерьера;
- убран ствол дерева в левой части комнаты, чтобы не загромождать кадр и разместить дверь;
- перерабатывалась цветовая гамма (по необходимости).

ОСНОВНЫЕ ЭТАПЫ СОЗДАНИЯ МОДЕЛЕЙ ИГРОВЫХ ОБЪЕКТОВ.

Моделирование всех объектов, начиная с *первого этапа* – представления его в виде простых фигур – выполнялось в Blender. Первый этап необходим для того, чтобы было легко воспроизводить пропорции как самого дома, так и предметов в нем. Далее была создана сама коробка (стены, крыша, пол), добавлены деревянные балки, бревна разного размера. создание и расположение окон и двери. Чтобы упростить процесс моделирования отверстия под дверной проем и оконные рамы, были использованы модификаторы – автоматические операции, влияющие на геометрию объекта неразрушающим образом. Они позволяют автоматически выполнять многие эффекты, которые непросто создавать вручную (например, поверхности подразделения) и не затрагивая базовую геометрию объекта. Можно сформировать стек модификаторов и вносить единообразные изменения сразу во все объекты. Вообще в Blender есть четыре вида модификаторов: *modify*, *generate*, *deform*, *simulat*. Они влияют на группы вершин, всю топологию меша, могут изменить общий вид объекта, добавить к нему новую геометрию и симуляцию физики. Использование модификаторов реализует процедурный метод разработки, который позволяет вносить изменения на любом этапе моделирования, при этом каждая правка будет влиять на все последующие пройденные шаги. По аналогии простыми формами с помощью модификаций моделируются другие основные части интерьера: полки, стол, стулья, печь, лестница, подвесной шкаф, часть чердака. Учитывая, что предметы выглядят реалистично, на восприятие влияет как расположение предметов, так и их размеры относительно комнаты и друг друга.

Второй этап разработки представляет собой предварительное нанесение материалов и текстур. Поскольку на изображении много деревянных объектов, в сцене преобладают бежевые, коричневые и охристые оттенки. Так как дом должен соответствовать тематике алхимии и работе с травами, поэтому было проработано сочетание с зелеными цветами. После того, как вся основная мебель была готова,

были расположены растения, книги, колбы и другие предметы декора. После распределения примерных цветов и материалов, на некоторые предметы для придания им большей реалистичности должны быть нанесены растровые текстуры. Например, деревянные предметы хоть и покрашены в приближенный к дереву цвет, но не обладают структурой, как у дерева или изделий из него.

Для мобильной графики с целью уменьшения веса модели до нанесения текстур производится ретопология и UV-развертка модели. *Ретопология* – процесс упрощения и переработки исходной сетки модели с целью уменьшения количества полигонов и разработки правильной UV-текстуры для 3D-объекта. *Hi-poly*- и даже *mid-poly*-модели обычно не используются для адаптивных приложений. *Low-poly*-модели создаются для мобильных устройств, а также дополненной и виртуальной реальности. Они содержат обычно всего 5-10 тысяч полигонов, что экономит ресурсы, но может ухудшить внешний вид, именно для этого игровые объекты были подвергнуты ретопологии. Объекты сложной формы были подвергнуты ручной развертке с указанием швов.

Текстурирование оптимизируемых по весу объектов имеет ряд особенностей, среди которых важнейшими являются наложение готовых изображений на поверхности, создание рельефных карт и др. Кроме того, помимо обычных текстур материалов, для оптимизации модели или для ее правильного импорта в игровые движки, существуют специальные текстуры, важнейшими из которых являются *ambient occlusion (AO)* и *normal map (NM)*, карта нормалей. АО – это способ затенения объектов, когда освещенность объекта зависит от наличия вблизи других объектов². Его можно использовать, чтобы подчеркнуть детали, акцентировать внимание на тенях и пр. NM – этот вариант текстурирования, позволяет получить более детализированный объект с меньшим количеством полигонов. Такие текстуры дают более выраженный рельеф на материале: карты нормалей создают иллюзию глубины, имитируют отражение луча от как будто бы существующей поверхности.

Для качественной оценки готовых материалов и текстур на сцену были добавлены и настроены источники света, для чего предва-

² PG, 2023. Ambient Occlusion в играх. [Электронный ресурс]. URL: <https://ru.pickgamer.com/post/ambient-occlusion-chto-eto-v-igrakh-617> (дата обращения: 13.03.2023).

рительно была настроена камера³ (выбран подходящий ракурс для тестового рендера). Камера – это объект, через который в Blender ведется визуализация изображений. Она определяет часть сцены, которая будет видна на отрисованном изображении. По умолчанию сцена имеет одну камеру: их может быть несколько, но активна – одна.

На сцене может быть несколько (не менее одного) источников света разного типа. Для Blendera это *point*, *sun*, *spot*, *area*. Point (точечный) – свет распространяется из одной точки равномерно во все стороны, с постепенным затуханием. Sun (солнечный) – свет распространяется из бесконечной плоскости прямолинейно в одном направлении, без затухания. Spot (прожектор) – лучи света исходят из одной точки, с ограничением угла распространения, с постепенным затуханием. Area – излучение происходит из плоскости, форму и размер которой можно настраивать. В готовую сцену был добавлен источник солнечного света. Далее вся комната была подсвечена основным точечным светом и отдельным источником света в области печи для имитации отсвета огня.

Когда все материалы, текстуры и UV-развертки сделаны, а освещение настроено, модели подготовлены к запеканию.

Запекание формирует двухмерные растровые изображения полигональной сетки отображаемой поверхности объекта. Эти изображения могут быть повторно спроецированы на объект при помощи UV-координат. Запекание производится для каждой сетки отдельно и может быть выполнено только при наличии ее UV-развертки. Хотя процесс требует некоторого времени для настройки и отрисовки, он помогает экономить время основной визуализации. Если, к примеру, необходимо отрисовать длительную анимацию, время, потраченное на запекание, может быть гораздо меньше времени, требуемого на визуализацию всех ее кадров. Запекание применяется для решений, интенсивно использующих свет и тень, например, для АО или мягких теней от ламп-областей. Различают два основных вида запекания: текстур и света.

Запекание текстур. Наиболее распространенным способом является запекание текстур (или карта запекания). Преимущество запекания текстуры заключается в способности перенести характеристики

³ DTF, 2023. Настройка света в Blender. [Электронный ресурс]. URL: <https://dtf.ru/gamedev/174085-nastroyka-sveta-v-blender> (дата обращения: 16.03.2023).

3D-геометрии на 2D-изображение. Сложные материалы и комплексное освещение могут быть объединены в одной текстуре. Это экономит огромные ресурсы за счет незначительных воспринимаемых изменений между низким и высоким разрешениями.

Запекание света более специфично для игровых движков. Для создания игр могут использоваться два типа освещения: *статическое* и *динамическое*. *Динамическое освещение* реагирует на все изменения в сцене: смещение теней, материалов, на которые они падают. *Статическое освещение* является стационарным и может быть исключено из расчетов для экономии ресурсов. Мобильные и веб-игры часто используют запеченные карты света, которые предоставляют информацию об освещении всей сцены или конкретного уровня игры. При запекании карт света, как статическое, так и динамическое освещения могут быть использованы одновременно, чтобы создать более реалистичную среду. Unity использует систему светового зонда, которая позволяет не статичному (анимированному или динамическому) объекту получать информацию из запеченных карт света, что позволяет экономить ресурсы с их помощью при симуляции динамического освещения. *Adobe Photoshop* применяется для постобработки готовой визуализации и текстурных карт⁴.

По завершении всех вышеперечисленных этапов были начаты тестовые рендеры с целью выявления проблем с материалами, текстурами, светом, пропорциями и пр. Финальный вид сцены показан на (Прил. 20, рис. 3). Таким образом, каждая модель для сцены дома алхимика прошла все этапы проектирования и реализации модели от скульптинга и моделинга до текстурирования, запекания и настройки света.

Следующий шаг реализации игрового приложения – перенос сцены в кроссплатформенную среду разработки игр Unity⁵, ее настройка, а также сбор тестового проекта, описание основных игровых механик, и разработка игрового интерфейса. *Визуализация трехмерных моделей в браузере* происходит на холсте <canvas> посредством ин-

⁴ Adobe Inc., 2023. Документация Adobe Photoshop. [Электронный ресурс]. URL: https://helpx.adobe.com/ru/pdf/photoshop_reference.pdf (дата обращения: 18.03.2023).

⁵ Unity, 2023. Документация Unity. [Электронный ресурс]. URL: <https://docs.unity3d.com/ru/530/Manual/UnityManual.html> (дата обращения: 10.03.2023).

струментария *WebGL*⁶ – программной библиотеки языка *JavaScript*, предназначенной для визуализации интерактивной трехмерной и двухмерной графики без использования плагинов. Чтобы построить и визуализировать корректную модель, надо описать каждую точку, линию, грань – это значительный объем кода, который способен написать только специалист, который, кроме знания языка *JavaScript*, хорошо осведомлен о геометрическом моделировании. Для повышения производительности труда разработчиков и расширения их круга была создана (распространяется под лицензией MIT) библиотека *Three.js*⁷. Библиотека предлагает набор классов для создания и отображения интерактивной 3D графики в *WebGL*. Она имеет декларативный синтаксис, и избавляет от необходимости детально разбираться в создании 3D-моделей.

Эти инструменты не решают проблем разработчиков трехмерных моделей, не знакомых с программированием. Для этой категории разработчиков *Soft8Soft* создал *Verge3D*⁸ – трехмерный рендерер реального времени, использующий технологию *WebGL*, который позволяет создателям изображений в *Blender*, *3ds Max* или *Maya* конвертировать трехмерные сцены для просмотра в браузере. Фактически, *Verge3D* – это движок, предоставляющий визуальный доступ к компонентам *Three.js*. Модель располагается в среде редактора и с помощью его инструментов интегрируется в код страницы.

Кроме анимации в играх, рекламных роликах, он используется на сайтах интернет-магазинов, реализующих, в частности мебель (Прил. 20, рис. 4), смартфоны и аксессуары, кондитерские изделия, спортивный инвентарь, ювелирные изделия, бытовую технику. Благодаря перечисленным технологиям, ориентированным на различные категории разработчиков, область применения трехмерной графики в веб неуклонно расширяется.

⁶ Mozilla, 2022. *WebGL Documentation*. [Электронный ресурс]. URL: https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API (дата обращения: 18.03.2023).

⁷ *Ehree.js r151*. [Электронный ресурс]. URL: <https://threejs.org/> (дата обращения: 18.03.2023).

⁸ *Verge3D*. [Электронный ресурс]. URL: <https://www.soft8soft.com/verge3d/> (дата обращения: 18.03.2023).