Инженерный вестник Дона, №1 (2021)
ivdon.ru/ru/magazine/archive/n1y2021/6783

Расширение и использование редактора визуального программирования для разработки виртуальных тренажеров

О.Г. Ведерникова, Н.А. Москат

Ростовский государственный университет путей сообщения

Аннотация: Рассмотрен виртуальный тренажер осмотрщика вагонов. Отличительная черта данного проекта – специально для этого тренажера был разработан редактор визуального программирования с набором необходимого функционала для разработки других виртуальных тренажеров. Редактор разработан для движка Unity3D . Проведено масштабирование и расширен функционал редактора, добавлены новые вершины в разработанную систему. Использован алгоритм обработки анимации и созданы новые классы. Получившиеся вершины объединены в цепь и проверена их работоспособность.
Ключевые слова: виртуальный тренажер, редактор визуального программирования, осмотрщик вагонов, движок Unity3D, коллайдер.

На текущий момент сфера виртуальных тренажеров, несмотря на свою специализированность и закрытость, довольно сильно развита. Виртуальный тренажер – это специальная программа или приложение, которое позволяет получать навыки и знания в конкретной предметной области или профессии. На данный момент в публицистической речи название «виртуальный тренажер» однозначен тренажеру в очках виртуальной реальности. На самом же деле, виртуальный тренажер – это любой тренажер, разработанный при помощи информационных технологий, а виртуальный тренажер с применением технологии виртуальной реальности уже будет подразумевать наличие очков виртуальной реальности для нормального использования приложения [1]. Тем не менее, независимо от названий или использованных технологий виртуальные тренажеры в современном мире занимают довольно большую нишу в специализированном обучении, в том числе в образовательных учреждениях [2, 3]. А технология виртуальной реальности постепенно занимает доминирующую позицию в сфере виртуальных тренажеров. Она позволяет использовать максимально возможную на данный момент степень погружения в виртуальный мир, что положительно сказывается на скорости и качестве обучения. И сейчас виртуальные

тренажеры начинают применять не только в опасных или требующих больших навыков профессиях (летчик, сапер), но и в достаточно массовых, однако требующих достаточного уровня практических навыков [4].

Несмотря на широкое развитие виртуальных тренажеров, у этой сферы есть существенный недостаток – децентрализованность средств и методов разработки, каждый проект необходимо реализовывать практически с нуля [5, 6]. Ну а сама по себе низкая массовость затрудняет развитие решений в данной сфере. Но данную проблему постепенно решают. Так, например, на базе Ростовского государственного университета путей сообщения был разработан виртуальный тренажер осмотрщика вагонов. Профессия осмотрщика вагонов требует немалого количества практических навыков, но полноценный и частый выезд в парк вагонов для проведения занятий затруднен. Этот недостаток компенсируется виртуальным тренажером, который позволяет полностью осмотреть вагон по всем позициям и вынести заключение о его исправности. Отличительная черта же данного проекта заключается в том, что, специально для этого тренажера был разработан редактор визуального программирования с набором необходимого функционала для разработки виртуальных тренажеров. Авторы выложили разработанный редактор в открытый доступ как альтернативный подход к разработке виртуальных тренажеров. Редактор в данный момент носит название VT Node Editor, но его главная особенность в том, что он по сути, является небольшой платформой визуального программирования для разработки виртуальных тренажеров, хотя и сами авторы называет свой проект редактором.

Редактор разработан для движка Unity3D [7, 8]. По заверению авторов, он содержит минимальный функционал для разработки именно виртуальных тренажеров, а больше всего подходит для тренажеров с технологией виртуальной реальности. Данный редактор является относительно

универсальным средством разработки. Истинность данных утверждений необходимо проверить на практике. Для этого можно расширить предлагаемый функционал редактора, проверить сложность такого расширения и удобство использования в процессе разработки. Так же необходимо понять, насколько данный редактор подходит к сфере виртуальных тренажеров и насколько уместна в нем система визуального программирования [9]. Для начала нужно понять, что представляет собой предлагаемый редактор.

Редактор строиться на системе неориентированных графов. Есть вершины графа (так называемые Nodes) и связи вершин. При этом каждая вершина несет в себе какой-то выполняемый функционал, уже заранее запрограммированный автором. Так, например, существуют такие стандартные вершины как:

1. Стартовая вершина – с нее начинается дерево вершин;
2. Конечная вершина – ей заканчивается дерево вершин;
3. Вершина объекта – посредством нее отображается объект;
4. Вершина звука – посредством нее воспроизводится звук;
5. Вершина коллайдера – посредством нее можно работать с геометрическим объемом объекта.

Этот набор вершин предоставлен разработчиком как минимально необходимый. Так же в плагине есть небольшой пример использования и набор необходимых компонентов для расширения редактора. Предоставленный разработчиками фрагмент примера показан на рис. 1. Целиком весь пример показан на рис. 2.

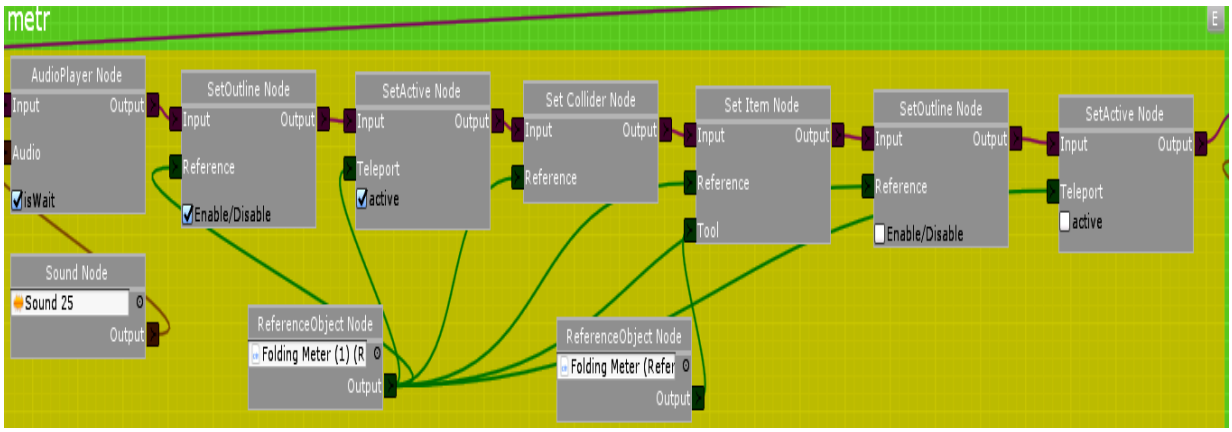

Рис. 1. – Фрагмент примера, предоставленного разработчиками

Рис. 2. – Пример, предоставленный разработчиками

После изучения всех необходимых элементов можно приступить к расширению редактора. Для полного понимания возможностей данной системы необходимо разработать как минимум две новых вершины, и с их использованием собрать небольшой рабочий пример. Эти действия помогут полноценно оценить работоспособность данного расширения.

В виртуальных тренажерах не последнее место занимает анимация. Если нужно лучше увидеть действия каких-либо деталей или объектов, то она просто необходима. Расширим функционал редактора, добавив в него возможность взаимодействия с анимациями объектов [10, 11].

Для этого наследуемся от исходного класса вершины, после этого система сама предоставит нам уникальный идентификатор нашей вершины. Затем опишем в ней необходимые поля для взаимодействия. Для вершины взаимодействия с анимацией будут нужны поля:

1. Входы и выходы вершины;
2. Должна ли анимация переключаться или находится в режиме ожидания;
3. Какая анимация должна быть проиграна.

Далее создаем класс, который будет производить непосредственную обработку действий, а в главном методе вершины создадим экземпляр и вызовем алгоритм обработки анимаций. Полученная вершина представлена на рис. 3.

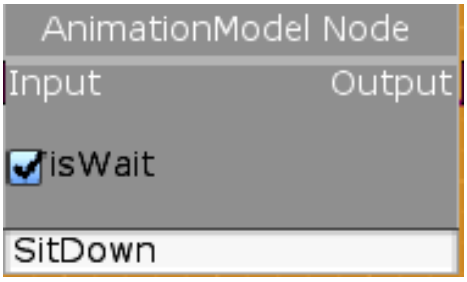

Рис. 3. – Вершина работы с анимацией

Теперь, после того как все анимации проиграны, объекту необходимо двигаться в каком-то направлении. Для этого также необходима соответствующая вершина. Для реализации необходимы следующие поля:

1. Входы и выходы вершины;
2. Цель, к которой двигаться.

Проделав вышеописанные действия, получим вершину движения объекта, представленную на рис. 4.

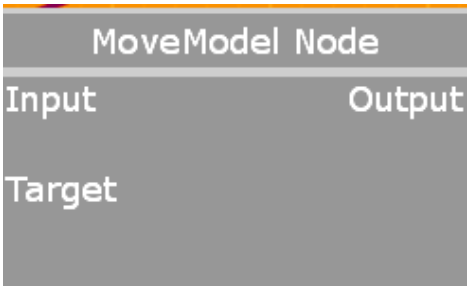

Рис. 4. – Вершина движения объекта

Теперь есть возможность объединить получившиеся вершины в цепь. Например, объект – человек. Он должен сесть, пройти дистанцию до цели и остановиться. Тогда такое дерево вершин будет выглядеть, как на рис. 5.

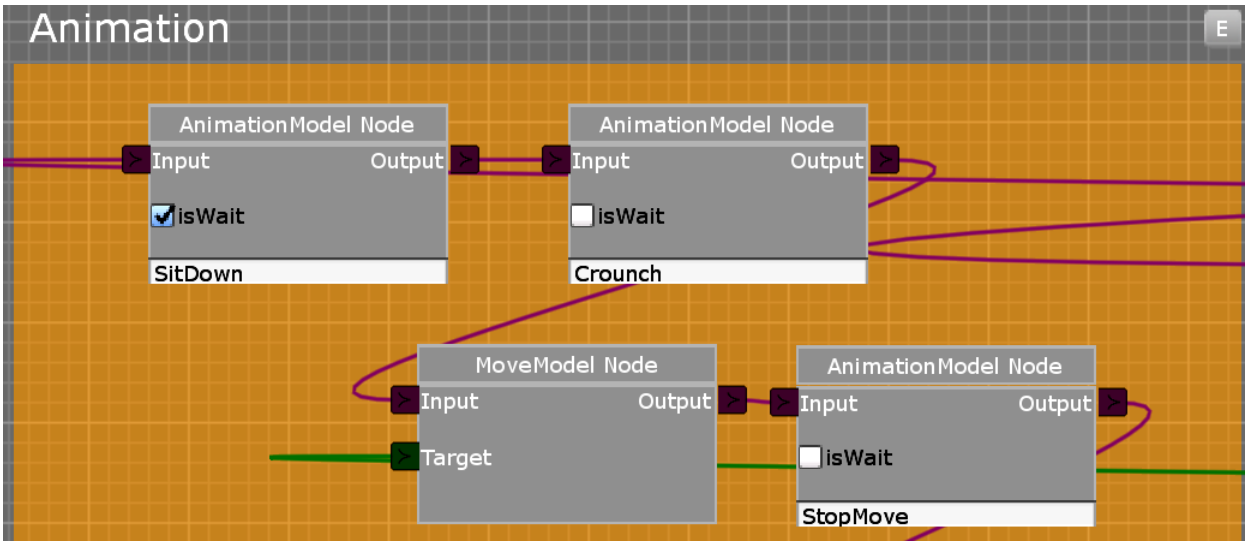

Рис. 5. – Пример дерева вершин

Исходя из выполненной работы, можно сказать, что редактор достаточно работоспособен. Он легко масштабируем и просто расширяем. При этом, добавление визуального программирования – это отличное решение в сфере виртуальных тренажеров. На практике стало понятно, что вершины выглядят достаточно информативно, чтобы даже человек, не

знакомый с программированием мог полноценно выполнять сборку дерева. То есть, в идеальном варианте, именно специалист в данной области, либо специально обученный человек будет заниматься сборкой дерева под необходимый тренажер, а разработчик будет расширять получающуюся платформу под конкретные нужды. Такой формат работы возможен именно благодаря визуальному программированию. Кроме того, визуальное программирование позволяет достаточно быстро и просто выполнять правки приложения, так как нет необходимости в чтении кода и в разборе зависимостей. Достаточно изменить вершину или переделать связи вершин.

Тем не менее, разработанный проект нельзя назвать завершенным. На данный момент у редактора выделяется ряд проблем:

1. Визуальное отображение связей вершин часто отображаются некорректно и налагаются друг на друга. При множестве связей становится тяжело разобрать, какая связь к какой вершине подходит.
2. Нет блока условного ветвления. На данный момент редактор может выполнять действия только последовательно, хотя на практике это далеко не всегда так.
3. Нет возможности цикличного выполнения. Сейчас, если предполагается множество одинаковых действий, то для каждого необходима вершина или система вершин. В перспективе, при больших проектах дерево вершин может стать слишком объемным.

Тем не менее, рассматриваемый проект можно назвать начальным этапом внедрения универсальных методов разработки в сферу виртуальных тренажеров. При этом, решение добавить в данный редактор систему визуального программирования особенно удачно и актуально именно в рассматриваемой сфере.

Литература

1. McLaughlin T., Cutler L., Coleman D. Character rigging, deformations, and simulations in film and game production // ACM SIGGRAPH 2011 Courses. – SIGGRAPH'11. – New York, NY, USA: ACM, 2011. – Pp. 5:1–5:18.

2. Молчанов О.Е., Васильев А.С., Белая Т.И. Компьютерный тренажер-эмулятор учебной цифровой вычислительной машины // Инженерный вестник Дона, 2015, № 2 URL: ivdon.ru/ru/magazine/archive/n2p2y2015/3046

3. Долгова Е.В., Файзрахманов Р.А., Курушин Д.С., Федоров А.Б., Хабибулин А.Ф., Шаронов А. А. Архитектура мобильного тренажера погрузочно-разгрузочного устройства // Инженерный вестник Дона, 2012, № 4 (часть1). URL: ivdon.ru/magazine/archive/n4p1y2012/1327

4. Юренко, И.К., Шепилова Е.Г., Гречук И.А. Совершенствование бортовых систем управления локомотивов на базе технических средств тренажеро-моделирующих комплексов // Инженерный вестник Дона, 2014, № 2. URL: ivdon.ru/ru/magazine/archive/n2y2014/2452

5. Букатов, А.А., Гридчина Е.Е., Заставной Д.А. Методы скелетной анимации для трансформации полигональных поверхностей трёхмерных моделей // Инженерный вестник Дона, 2012, № 3. URL:ivdon.ru/ru/magazine/archive/n3y2012/897

6. Vasilakis, A. A., Fudos I. GPU rigid skinning based on a refined skeletonization method // Comput. Animat. Virtual Worlds. – 2011.–January. – Vol. 22. – Pp. 27–46.

7. Хокинг, Д. Unity в действии. Мультиплатформенная разработка на C#. – СПб.: Питер, 2016. – 336 с.

8. Торн, А. Искусство создания сценариев в Unity. – М.: ДМК 2016. – 360 с.

9. Sung K., Gregory S. Basic Math for Game Development with Unity 3D. - New York: Springer Science + Business Media, 2019. - 414 p.

10. Дикинстон К. Оптимизация игр в Unity 5. Советы и методы оптимизации приложения, охватывающие все аспекты работы с движком unity3D. - М.: ДМК. Пресс, 2017. – 306 с.

11. Аникеев, А.С. Генерация и моделирование 2D ландшафта по контрольным значениям с использованием Unity на языке программирования // Инженерный вестник Дона, 2020, № 4. URL: ivdon.ru/ru/magazine/archive/N4y2020/6433

References

1. McLaughlin T., Cutler L., Coleman D. ACM SIGGRAPH 2011 Courses. SIGGRAPH'11. New York, NY, USA: ACM, 2011. Pp. 5:1–5:18.

2. Molchanov, O.E., Vasil′yev A.S., Belaya T.I. Inzhenernyj vestnik Dona, 2015, № 2. URL: ivdon.ru/ru/magazine/archive/n2p2y2015/3046

3. Dolgova, E.V., Fayzrakhmanov R.A., Kurushin D.S., Fedorov A.B., Khabibulin A.F., Sharonov A.A. Inzhenernyj vestnik Dona, 2012, № 4. URL: ivdon.ru/magazine/archive/n4p1y2012/1327

4. Yurenko, I.K., Shepilova E.G., Grechuk I.A. Inzhenernyj vestnik Dona, 2014, № 2. URL: ivdon.ru/ru/magazine/archive/n2y2014/2452

5. Bukatov A.A., Gridchina E.E., Zastavnoy D.A. Inzhenernyj vestnik Dona, 2012, № 3. URL: ivdon.ru/ru/magazine/archive/n3y2012/897

6. Vasilakis, A. A., Fudos I. Comput. Animat. Virtual Worlds. 2011. January. Vol. 22. Pp. 27–46.

7. Hocking J. Unity v deystvii. Mul′tiplatformennaya razrabotka na C# [Unity in action. Multi-platform development in C#]. SPb.: Piter, 2016. p. 336.

8. Thorne, A. Iskusstvo sozdaniya stsenariyev v Unity [The art of scripting in Unity]. M.: DMK. 2016. p. 360.

9. Sung K., Gregory S. Basic Math for Game Development with Unity 3D. New York: Springer Science + Business Media, 2019. 414 p.

10. Dikinston K. Optimizacija igr v Unity5. Sovety i metody optimizacii prilozhenija, ohvatyvajushhie vse aspekty raboty s dvizhkom unity3D [Optimization of games in Unity 5. Tips and methods for optimizing the application, covering all aspects of working with the unity3D engine]. M.: DMK. Press, 2017. p. 306.

11. Anikeev A.S. Inzhenernyj vestnik Dona, 2020, № 4. URL: ivdon.ru/ru/magazine/archive/N4y2020/6433