УДК 519.2

С. Ю. ПЛЕСНЕЦОВ, Н. Н. ЮДАНОВА, А. С. КОВАЛЕНКО

РЕАЛИЗАЦИЯ СИМУЛЯЦИИ АКУСТИЧЕСКИХ ПРОЦЕССОВ НА БАЗЕ ПРОГРАММНОЙ ПЛАТФОРМЫ UNREAL ENGINE 4

Изложены этапы разработки программного симулятора акустических колебаний. Приведены главные теоретические основы, заложенные в программный продукт при разработке. Изложены допущения, сделанные в рамках теоретической базы с целью оптимизации алгоритма. Приведены элементы алгоритмической логики и технической базы, в рамках которой выполнялась разработка. Указаны и охарактеризованы основные элементы, на основе которых построены классы программного продукта. Охарактеризованы и описаны программные классы, реализованные в рамках процесса разработки. Приведены листинги разработанных классов в формате Unreal Engine Blueprint. Приведены скриншоты реализованного программного обеспечения для сценариев прямой и поперечной волны (прямого и наклонного датчика). Охарактеризованы качественно параметры соответствия полученной симуляционной модели реальным образцам. Охарактеризованы недостатки полученной модели и рассмотрены пути их устранения.

Ключевые слова: симуляция программная, unreal engine 4, движок игровой, классы программные, объектно-ориентированный подход, приложение Windows, контроль акустический

С. Ю. ПЛЄСНЕЦОВ, Н. М. ЮДАНОВА, А. С. КОВАЛЕНКО

РЕАЛІЗАЦІЯ СИМУЛЯЦІЇ АКУСТИЧНИХ ПРОЦЕСІВ НА БАЗІ ПРОГРАМНОЇ ПЛАТФОРМИ UNREAL ENGINE 4

Викладено етапи розробки програмного симулятора акустичних коливань. Наведено головні теоретичні основи, закладені в програмний продукт при розробці. Викладено допущення, зроблені в рамках теоретичної бази з метою оптимізації алгоритму. Наведено елементи алгоритмічної логіки і технічної бази, в рамках якої виконувалася розробка. Вказані і охарактеризовано основні елементи, на основі яких побудовані класи програмного продукту. Охарактеризовані й описані програмні класи, реалізовані в рамках процесу розробки. Наведено листинги розроблених класів в форматі Unreal Engine Blueprint. Наведено скріншоти реалізованого програмного забезпечення для сценаріїв прямий і поперечної хвилі (прямого і похилого датчика). Охарактеризовані якісно параметри відповідності отриманої симуляційної моделі реальним зразкам. Охарактеризовані недоліки отриманої моделі і розглянуті шляхи їх усунення.

Ключевые слова: симуляція програмна, unreal engine 4, движок ігровий, класи програмні, об'єктно-орієнтований підхід, додаток Windows, контроль акустичний

S. YU. PLESNETSOV, N. N. YUDANOVA, A. S. KOVALENKO

IMPLEMENTATION OF ACOUSTIC PROCESSES SIMULATION ON THE BASIS OF UNREAL ENGINE 4 SOFTWARE PLATFORM

The stages of development of the software simulator of acoustic oscillations are outlined. The main theoretical foundations laid down in the software product during development are given. The assumptions made within the framework of the theoretical framework for the purpose of optimizing the algorithm are stated. Elements of algorithmic logic and technical base within which development was carried out are presented. The main elements on the basis of which the classes of the software product are built are indicated and characterized. The program classes realized within the development process are characterized and described. The listings of the developed classes in Unreal Engine Blueprint format are given. The commentary for each listing is provided. Screenshots of implemented software for direct and transverse wave scenarios (direct and oblique sensor) are provided. Qualitative parameters of the correspondence of the obtained simulation model to real samples are characterized. Graphical representation of simulated testing model is given. The shortcomings of the received model are characterized and ways of their elimination are considered.

Keywords: simulation software, unreal engine 4, game engine, software classes, object-oriented approach, Windows application, acoustic testing

Введение. Программная симуляция процессов является эффективным способом упрощения задачи исследования и анализа процессов, сложных для прямой визуализации и экспериментальной оценки (в особенности для случая высокой затратности процесса).

В то же время, существующие решения, такие как Ansys Mechanical Suite и Comsol Multiphysics, как правило предлагают крайне широкий, но, во многих случаях, избыточный инструментарий при высокой стоимости продукта и трудоемкости постановки задачи, а также высокие затраты времени на осуществление вычислений.

1. Разработка общего алгоритма упрощенной симуляции акустических волн

1.1. Общие вопросы программной симуляции акустических импульсов

Для разработки алгоритма в упрощенной форме зададимся рядом утверждений, упрощенно характеризующих модель ультразвукового импульса, описанного в [1]:

1) Волновой импульс можно рассматривать как автономный объект, располагающий скоростью, запасом энергии, расходуемым на рассеивание и поглощение, и несущий в себе информацию о времени, затраченном на перемещение.

2) При преодолении границы раздела сред, объект-импульс распадается на два различных объекта – отраженный, несущий в себе запас энергии, соответствующий коэффициенту отражения, и преломленный, несущий в себе запас энергии, соответствующий коэффициенту прохождения.

3) Для данного исследования будем рассматривать диапазон в пределах критических углов и случай нормального прохождения импульса, т.е., случаи, когда можно пренебречь либо продольной, либо поперечной составляющей волны.

4) Эффектами взаимодействия волновых импульсов также пренебрегаем.

В таком случае мы можем представить импульс как условную точку, имеющую пространственные

координаты, скорость, запас энергии и счетчик времени, а материал – как область пространства, обладающую габаритами, набором пространственных границ, определяющих нормали, номинальной скоростью прохождения продольной и поперечной волн, коэффициентом затухания.

Эффект преломления возникает, когда импульс покидает зону одного материала и входит в зону другого материала.

В данном случае необходимо:

1) Определить нормаль к плоскости вхождения импульса в материал.

2) Определить угол между вектором вхождения импульса и нормалью.

3) По закону Снеллиуса [2] определить изменение направления вектора скорости импульса.

Прием сигнала можно реализовать через специальную область материала, описываемую как приемник. В этой области импульс уничтожается, а несомые им данные о запасе энергии и времени в пути передаются в базу данных для визуализации.

1.2. Математическое описание компонентов алгоритма

1) Перемещение репрезентации импульса.

Для перемещения репрезентации импульса используются законы равномерного прямолинейного движения. Т.е., базовая зависимость имеет вид:

$$S = c \cdot t$$

Поскольку в случае с игровыми движками реализация будет осуществляться дискретно, через определенные (неравномерные) промежутки времени, и через сумму компонентов вектора скорости на оси координат то это выражение примет вид:

$$S_x = c \cdot t \cdot k_x, \quad (1)$$

$$S_y = c \cdot t \cdot k_y, \quad (2)$$

где k_x, k_y – коэффициенты преобразования для проекции вектора скорости на оси координат (рис. 1), рассчитываемые как:

$$k_x = \frac{c_x}{c}$$

$$k_y = \frac{c_y}{c}$$

c_x, c_y – спроецированные компоненты вектора скорости. Их длины определяются по известному изменению координат.

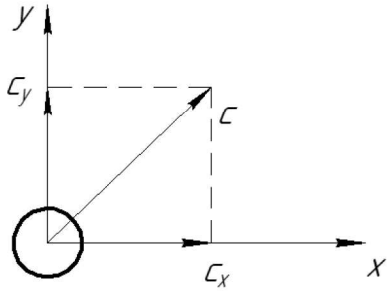


Рис. 1 – Схема движения репрезентации импульса

2) Преломление и отражение репрезентации импульса

Преломление будем рассчитывать согласно закона Снеллиуса (рис. 2):

$$\sin(\beta) = \sin(\alpha)\frac{c_2}{c_1} \quad (3)$$

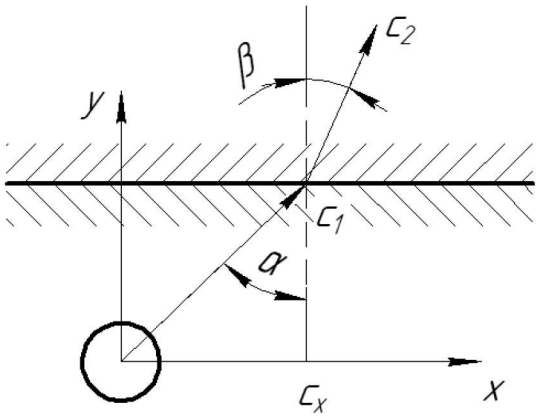


Рис. 2 – Процесс преломления

В случае с отражением сигнала, необходимо порождение дополнительной репрезентации импульса, с соответствующим сокращением несомого данным импульсом запаса энергии:

$$E_1 = \frac{E_1}{k_{pass}}, \quad (4)$$

$$E_2 = \frac{E_1}{k_{ref}}, \quad (5)$$

где k_{pass} и k_{ref} – коэффициенты прохождения и отражения соответственно;

E_1, E_2 – запасы энергии основного (преломленного) и отраженного импульсов.

Угол же отраженного сигнала равен двум углам падения.

3) Затухание волны

Для определения потерь энергии в ходе затухания используется зависимость:

$$E_n = E_o \cdot e^{-2\delta t}, \quad (6)$$

где E_o, E_n – запасы энергии до и после итерации затухания;

δ – коэффициент затухания;

t – длительность данного дискретного интервала.

2. Реализация алгоритма для платформы Unreal Engine 4. Движок Unreal Engine 4 представляет собой совокупность программных библиотек и инструментов, ориентированных в первую очередь на разработку интерактивных мультимедиа продуктов. Однако поставляемые с движком библиотеки позволяют быстро вычислять координаты, скорости, направления движения, моменты и координаты пересечения единовременно множества объектов.

Каждый объект в пространстве сцены является частным представлением (инстансом) объекта определенного класса. Соответственно, описанные в разделе 1.1 составляющие следует представить как соответствующие универсальные классы объектов, объединенные интерфейсом для обмена данными.

Главными объектами являются репрезентация импульса (*soundwave_rep*) и репрезентация объема материала (*material_item*). Для вывода графической информации добавлены класс-структура для хранения сводки данных о параметрах принятых сигналов (*wavedata*) и класс графического виджета интерфейса пользователя (*UI*), а для порождения репрезентаций импульсов добавлен класс условного «генератора» - спаунер (*soundwave_spawner*). Представлены также дополнительные технические классы – общий режим, контроллер движения и класс-пешка, управляемый этим контроллером (для манипуляций с камерой). Все объекты классов отображены на рис. 3.

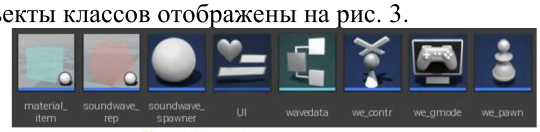


Рис. 3 – Классы в проекте

Основное программирование выполнено с помощью инструментария визуального скриптинга Unreal Engine 4 – Blueprint [3]. Составные структурные схемы визуальных алгоритмов приведены ниже.

1) Спаунер частиц

Спаунер частиц порождает репрезентации импульса с заданной частотой на основе циклически возобновляемого таймера. Длительность таймера определяется из периода, определяемого из задаваемой пользователем частоты.

$$T = \frac{1}{freq}$$

где *freq* – переменная с плавающей точкой, хранящая данные о частоте.

После порождения сигнала в заданных координатах, спаунер присваивает ему изначальные значения запаса энергии, скорости, типа волны (продольная или поперечная) и изначального угла, под которым объект будет перемещаться.

Схема спаунера приведена на рис. 4.

2) Репрезентация импульса

Объект, отвечающий за репрезентацию импульса, является центральным элементом системы.

Ключевыми задачами при разработке данного класса являются система движения (рис. 5) и обработчик столкновений, обрабатывающий процессы преломления и отражения (рис. 6).

При движении обрабатываются процессы собственно перемещения по формулам (1, 2) и затухания по формуле (6).

При обработке преломления нормаль к поверхности столкновения [4] определяется с помощью линейного трейсинга [5] (проведения виртуального луча между двумя точками с выявлением точек столкновения). Процедура трейсинга автоматически включает нормаль в векторной форме в виде составляющей структуры Hit Result, включенной в состав стандартной библиотеки Kismet Library, что упрощает задачу. Угол между нормалью и вектором скорости определяется как арксинус скалярного произведения этих векторов (см. рис. 6).

В результате порождается отраженный импульс и изменяется угол движения и скорость данного.

Необходимость выполнения расчета преломления и отражения определяется на основе изменения текущей скорости звука в материале, данные о которой транслируются самим объектом-материалом (см. ниже), и расчет проводится ретроактивно.

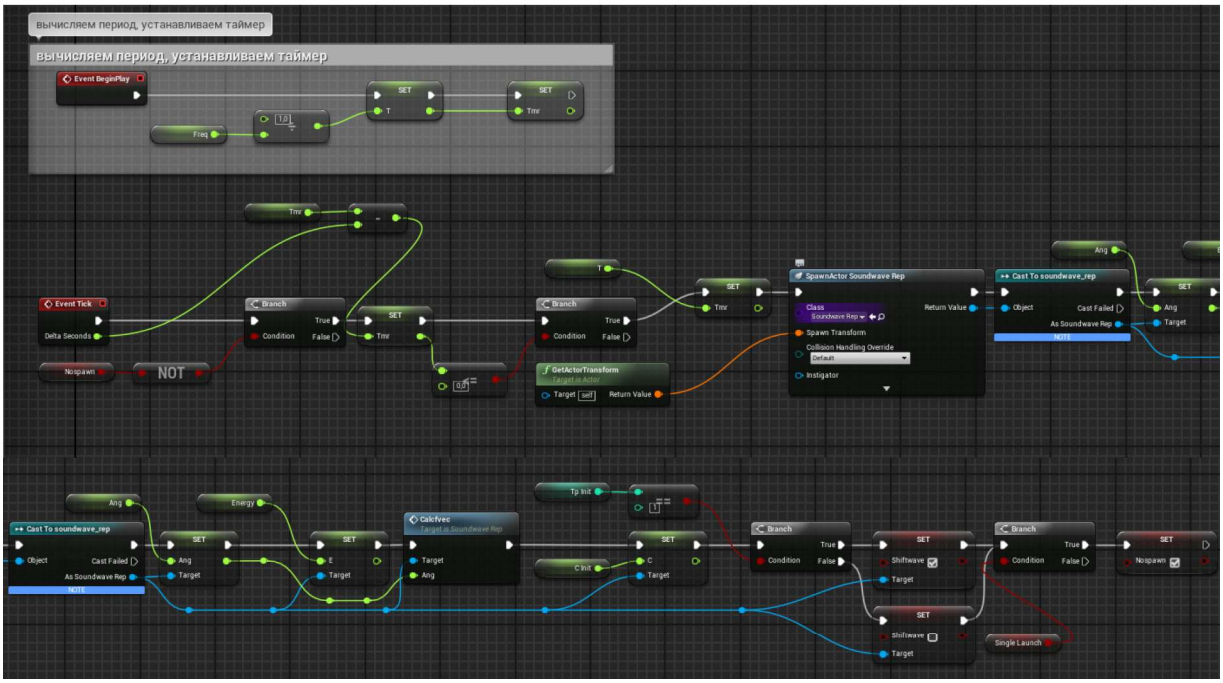


Рис. 4 - Спаунер частиц

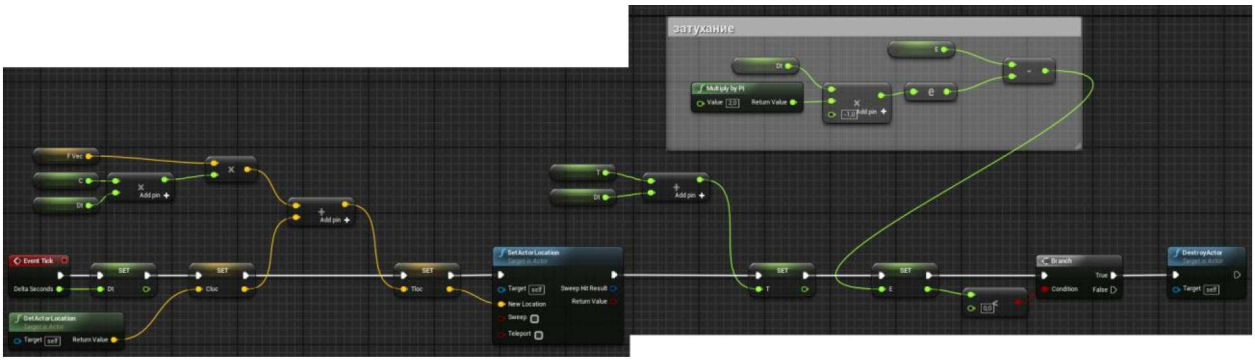


Рис. 5 - Обработка поведения виртуального импульса во времени

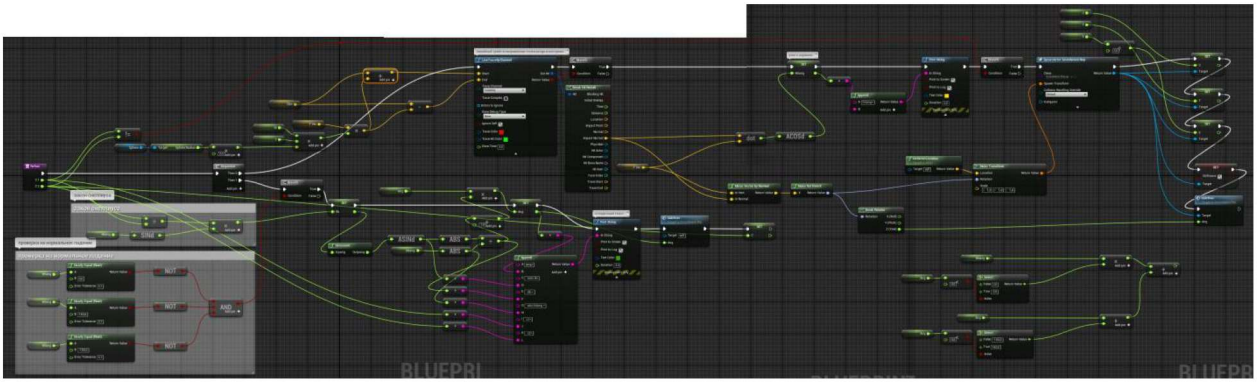


Рис. 6 - Обработка преломления и отражения

3) Объект-материал

Объект-материал в текущей итерации представляет собой параллелепипед, окруженный блокирующими плоскими объемами, ограничивающими его для определения точек пересечения его пространства репрезентациями импульса.

Структура объекта-материала включает шесть компонентов (рис. 7): четыре блокирующих плоских объема (1), настроенные на перехват сигналов линейного трейсинга репрезентаций импульсов, объем-параллелепипед, настроенный на перехват наложений (тип столкновений в Unreal Engine 4) (2), а также визуальное представление объема, реализованное мешем-параллелепипедом. По габаритам все компоненты совпадают.

На базовом уровне материал отслеживает взаимодействующие с ним импульсы и назначает им заданную в себе скорость (для продольной или поперечной волны, со сменой типа волны в случае углового взаимодействия) и инициируя процесс расчета преломления/отражения (рис. 8).

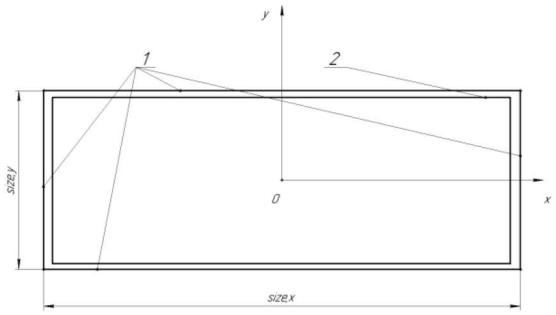


Рис. 7 – Схема построения объекта-материала

В то же время, если для такого объекта установить булев флаг receiver (такой флаг может быть установлен только у одного объекта, остальные будут игнорироваться), данный материал станет приемником и будет принимать сталкивающиеся с ним импульсы, записывая хранящиеся в них данные в массив структур для последующей визуализации (рис. 9).

При этом абсциссой выступает ось времени, ординатой – ось энергии. В случае совпадения времени, энергии суммируются.

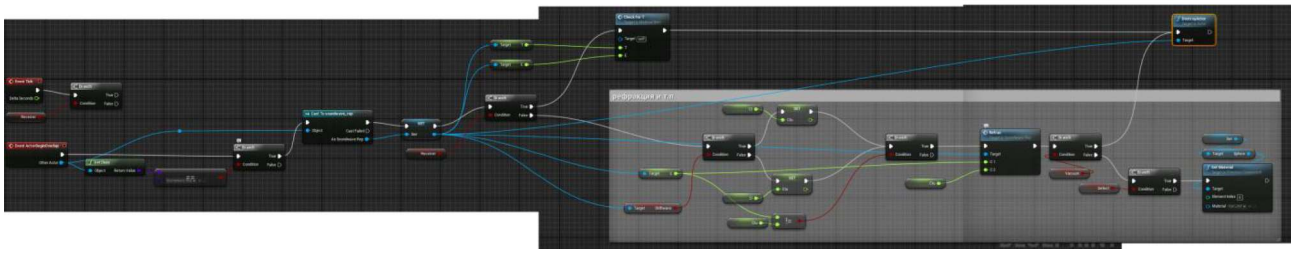


Рис. 8 - Блок материала

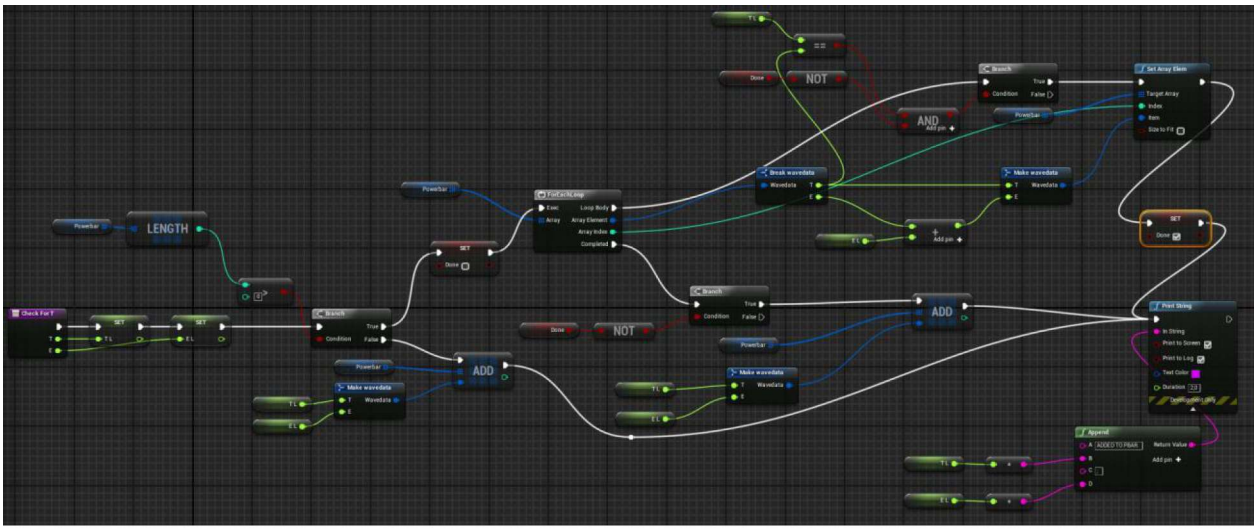


Рис. 9 - Обновление базы данных для построения графика время-энергия

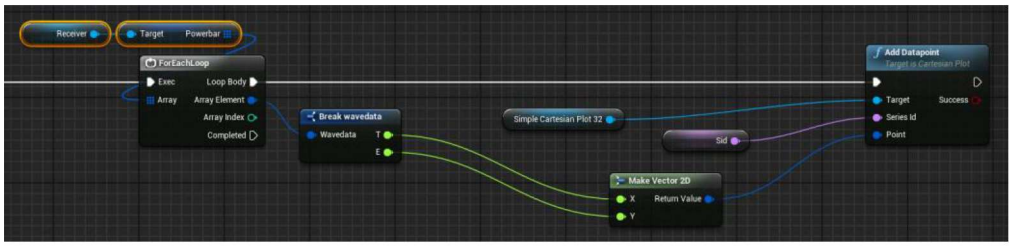
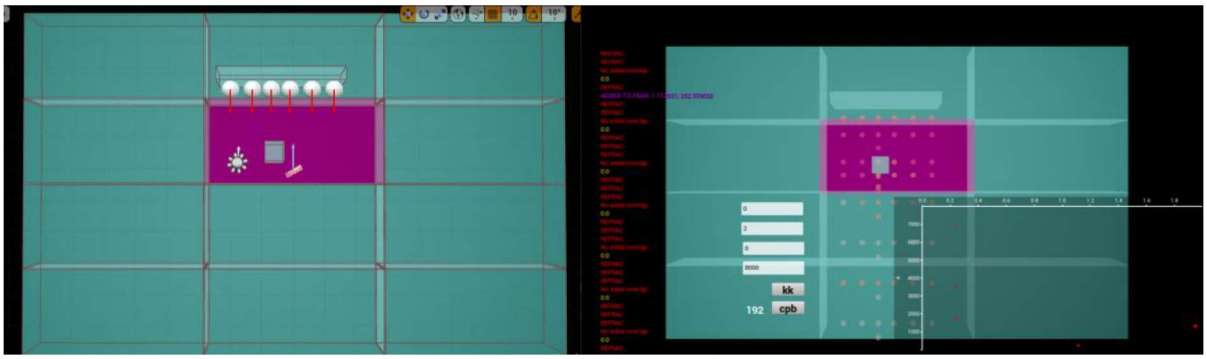
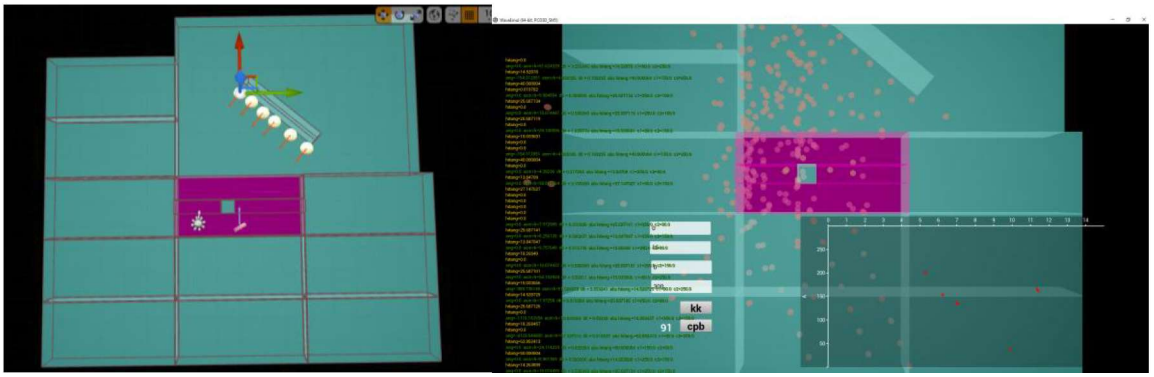


Рис. 10 - Построение графика



а б



в г

Рис. 11 – Реализация приложения

а – настройка сцены для нормального падения волн; *б* – исполнение решения для нормального падения волн; *в* – настройка сцены для падения волн под углом; *г* – реализация сцены для падения волн под углом

4) UI-объект

Графический интерфейс используется для визуализации данных, полученных материалом со свойством receiver, с помощью стороннего плагина Kantan Charts (рис. 9). График строится в виде набора точек, проецируемых на поле, в рамках алгоритма обработки базы данных, формируемой материалом receiver (рис. 10).

Перечисленные выше объекты образуют систему, размещаемую на сцене (в рабочем пространстве редактора движка), и начинают работать немедленно после старта приложения. На рис. 11 приведены конфигурации сцен (а, в) и их исполнение (б, г) для случая нормального падения волн (а, б) и в случае падения волн под углом (в, г)

3. Проверка исполнения алгоритма. Тестирование разработанного приложения проводилась в рамках сопоставления полученного визуального образа сигнала на ПК и для схожего образца, проверяемого на дефектоскопе УД2-12.

Качественная структура распределения точек для условного образца не выявила расхождений по временной развертке (рис. 12).

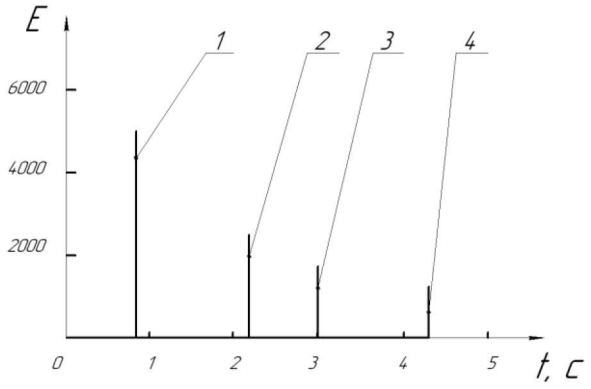


Рис. 12 – Распределение отраженных импульсов во времени

На рис. 12: 1 – отраженный импульс от поверхности образца; 2 – отраженный импульс от дефекта; 3 – переотраженный импульс от дефекта; 4 – донный импульс.

Важно отметить, что при проверке были существенно и пропорционально снижены реальные частоты генерации импульсов и скорости перемещения сигнала для наглядности отображения информации. Так, скорость звука для стали была принята равной 80, для оргстекла – 50. Частота генерации составила 1 Гц.

В то же время выявлен ряд устранимых проблем, связанных с быстрым увеличением числа репрезентаций импульсов при множественных отражениях в малых объемах.

Выводы.

1. Программная платформа Unreal Engine 4 может использоваться для визуализации и симуляции акустических процессов с соблюдением ряда допущений и упрощений, необходимых для получения должной скорости исполнения процессов.

2. Симуляции в рамках полученного решения дают качественно корректное распределение в рамках шкалы время-энергия, что позволяет допускать возможность точной оценки численных характеристик сигнала.

Список литературы

1. *Неразрушающий контроль: Справочник*: В 8 т. / Под общ. ред. В.В. Клюева. Т. 3: И.Н. Ермолов, Ю.В. Ланге. *Ультразвуковой контроль*. – 2-е изд., испр. – М.: Машиностроение, 2006. – 864 с.: ил.
2. Сучков Г. М. *Акустический контроль : учеб. пособие* / Г. М. Сучков, Е. Л. Ноздрачева. – Харьков : НТУ "ХПИ", 2013. – 138 с.
3. *Blueprints Visual Scripting for Unreal Engine* / Brenden Sewell. - Packt Publishing, 2015. – 188 p.
4. *Unreal Engine 4 – Physics - Collision* [Electronic resourse] / Epic Games // Electronic data. – Mode of access: World Wide Web: https://docs.unrealengine.com/en-us/Engine/Physics/Collision (viewed on May 1, 2018). – Title from the screen.
5. *Unreal Engine 4 – Using a Single Line Trace (Raycast) by Channel* [Electronic resourse] / Epic Games // Electronic data. – Mode of access: World Wide Web: https://docs.unrealengine.com/en-us/Engine/Physics/Tracing/HowTo/SingleLineTraceByChannel (viewed on May 1, 2018). – Title from the screen.

References (transliterated)

1. *Nerazrushayuschiy kontrol: Spravochnik* [Nondestructive testing: guide]: In 8 volumes / Ed. V.V. Klyuev. Vol. 3: I.N. Ermolov, Yu.V. Lange. *Ultrazvukovoy control* [Ultrasonic testing]. – 2nd ed., fixed. – Moscow: Mashinostroenie, 2006. – 864 p.
2. Suchkov G. M. *Akusticheskiy kontrol: ucheb. posobie* [Acoustic testing: tutorial] / G. M. Suchkov, E. L. Nozdracheva. – Kharkiv : NTU "KhPI", 2013. – 138 p.
3. Blueprints Visual Scripting for Unreal Engine / Brenden Sewell. - Packt Publishing, 2015. – 188 p.
4. Unreal Engine 4 – Physics - Collision [Electronic resourse] / Epic Games // Electronic data. – Mode of access: World Wide Web: https://docs.unrealengine.com/en-us/Engine/Physics/Collision (viewed on May 1, 2018). – Title from the screen.
5. Unreal Engine 4 – Using a Single Line Trace (Raycast) by Channel [Electronic resourse] / Epic Games // Electronic data. – Mode of access: World Wide Web: https://docs.unrealengine.com/en-us/Engine/Physics/Tracing/HowTo/SingleLineTraceByChannel (viewed on May 1, 2018). – Title from the screen.

Надійшла (received) 05.06.2018

Відомості про авторів / Сведения об авторах / About the Authors

Плєснецов Сергій Юрійович (***Плеснецов Сергей Юрьевич***, ***Plesnetsov Sergey Yurievich***) – кандидат технічних наук, старший викладач каф. КРСКД НТУ «ХПІ»; м. Харків, Україна; ORCID: https://orcid.org/0000-0001-8428-5426; e-mail: s.plesnetsov@gmail.com

Юданова Ніна Миколаївна (Юданова Нина Николаевна, Yudanova Nina Nikolaevna) – ст. викл. каф. КРСКД НТУ «ХПІ», м. Харків, Україна

Коваленко Аліна Сергіївна (***Коваленко Алина Сергеевна***, ***Kovalenko Alina Serhiivna***) – студентка каф. КРСКД, Національний технічний університет «Харківський політехнічний інститут», м. Харків, Україна