**УО «Белорусский государственный технологический университет»**

Факультет **информационных технологий**

Кафедра **информационных систем и технологий**

Специальность **1-40 05 01-03 «Информационные системы и технологии (издательско-полиграфический комплекс)»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОЙ РАБОТЫ**

**по дисциплине «**Компьютерные мультимедийные системы в издательском деле»

**тема «**Разработка компьютерной обучающей системы “Изучение магнитного гистерезиса с помощью осциллографа”»

|  |  |  |
| --- | --- | --- |
| **Исполнитель** |  | |
| Студент 3 курса группы3 | подпись, дата | В. В. Пригодич  инициалы и фамилия |
|  |  |  |
| **Руководитель** |  |  |
| Доцент | подпись, дата | Н. И. Гурин  инициалы и фамилия |

|  |  |
| --- | --- |
| **Курсовая работа защищена с оценкой** | **\_\_\_\_\_\_\_\_\_\_** |
| **Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  (подпись) | Н. И. Гурин  инициалы и фамилия |

**Минск 2022**

**Содержание**

Введение 3

1. обзор методов и программных средств разработки компьютерных мультимедийных систем 4

1.1. средства для разработки компьютерных мультимедийных систем 4

1.2. методологии разработки мультимедийных систем 4

2. последовательность разработки содержания компьютерной мультимедийной системы 5

2.1. веб-сайт 5

2.2. озвученная карта установки и анимация алгоритма работы 13

2.3. трехмерная модель установки 15

3. описание хода работы 3d-симулятора с копиями экрана пользовательского интерфейса основных элементов системы, а также ключевых функций разработанного программного кода на языке c# 19

4. разработка базы знаний и режима диалога с информационной системой 30

5. руководство пользователя 3d-симулятором 33

заключение 37

список использованных источников 38

приложение А 39

приложение Б 54

приложение В 70

# **Введение**

На сегодняшний день информационные технологии используются во всех сферах деятельности людей. Мы не можем представить современный мир без использования мультимедийных технологий и их различных производных. Результаты эволюции и прогресса применяются во всех сферах: начиная от создания компьютерных учебных курсов и заканчивая промышленной отрасли целого государства. Также, новые технологии часто создаются в результате сотрудничества между творческими профессионалами и разработчиками программного обеспечения. В результате получается более дружественное и качественное программное средство, ликвидирующее барьер между пользователем и программой.

Мультимедийные системы и технологии позволяют сделать процесс обучения более эффективным. Используя их, можно добиться оптимального баланса простоты и эффективности в образовании, ведь сочетание классических средств образования с современными мультимедийными технологиями – один из лучших способов модернизировать образовательный процесс с различных сторон [1].

Целью данного курсового проекта будет сделать такое обучающее программное средство, которое, благодаря современным мультимедийным возможностям, позволило бы любому пользователю с легкостью ознакомиться со всеми шагами физического эксперимента, изучить теорию и, с помощью полученных знаний, самостоятельно провести лабораторную работу по теме «Изучение магнитного гистерзиса с помощью осциллографа» на построенном нами симуляторе.

Мультимедийная система будет состоять из трех частей:

* веб-сайт, которых будет содержать теоретическую информацию о проведении опыта;
* обучающий симулятор физической установки, созданный в Unity;
* база знаний, встроенная в веб-сайт.

# **Обзор методов и программных средств разработки компьютерных мультимедийных систем**

## **Средства для разработки компьютерных мультимедийных систем**

Для разработки компьютерной мультимедийной системы будут использоваться следующие программные средства и технологии:

* среда разработки Visual Studio Code;
* язык гипертекстовой разметки HTML и каскадная таблица стилей CSS;
* язык программирования JavaScript;
* открытое программное обеспечение для трехмерного моделирования 3D Studio Max 2022;
* программное обеспечение для создания мультимедиа и анимации Adobe Animate;
* межплатформенная среда для разработки компьютерных игр Unity;
* язык программирования C#.

## **Методологии разработки мультимедийных систем**

Разработка веб-сайта начинается с проектирования разметки и стилей с помощью HTML и CSS. Используется язык программирования JavaScript для создания логики в элементах страницы. Вся разработка и проектирование веб-сайта происходит в Visual Studio Code.

На сайте будут использоваться озвученные карта установки и анимация алгоритма работы, которые создаются в Adobe Animate.

Моделирование установки и окружающей среды происходит в 3D Studio Max. Именно из этой среды интегрируем модели в Unity.

В самом Unity выставляется окружение с симулятором, проектируется работа с самим симулятором (задания, интерфейс, взаимодействие с установкой). Программирование событий установки происходит на языке C# в среде разработки Visual Studio Code.

Для создания базы знаний на сайте используется JavaScript, HTML и CSS для визуального отображения диалогового окна базы знаний.

В конечном итоге должен получиться один веб-сайт, содержащий в себе различные мультимедийные системы, позволяющие сполна изучить выбранную тему.

# **Последовательность разработки содержания компьютерной мультимедийной системы**

## **Веб-сайт**

Начинать разработку мультимедийной системы стоит с сайта. Первым делом стоит приступить к титульной странице, которая будет содержать кнопку перехода к навигационному меню (как и каждая страница), тему проекта и фото разработчика проекта. HTML разметка титульной страницы представлена на листинге 2.1.

*Листинг 2.1 – HTML разметка титульной страницы*

Благодаря подключению CSS файла CSSForTitlePage.css, который добавляет стили и динамические взаимодействия на страницу и листинг которого представлен в приложении А, получаем титульную страницу, скриншот которой представлен на рис. 2.1.

*Рисунок 2.2 – Титульная страница*

К динамическим элементам на главной странице можно отнести затемнение кнопки «Меню» при наведении (рис. 2.3).

*Рисунок 2.3 – Затемнение кнопки при наведении*

При нажатии на кнопку «Меню» осуществляется переход на страницу с главным меню, где расположены ссылки на основные страницы сайта (рис. 2.4).

*Рисунок 2.4 – Страница меню*

К динамическим элементам на странице меню можно отнести изменение положения текста элемента меню при навидении (рис. 2.5).

*Рисунок 2.5 – Изменение положения текста при наведении*

На странице «Теоретическое введение» содержится весь теоретический материал, относящийся к разрабатываемой обучающей системе. HTML разметка страницы «Теоретические введение» представлен в приложении А данной пояснительной записки.

На рисунке 2.6 показано, как выглядит страница «Теоретические введение».

*Рисунок 2.6 – Страница «Теоретические введение»*

К динамическим элементам на данной странице можно отнести увеличение формул по нажатию на них курсором мыши и возвращение к первоначальному положению при уходе курсором мыши с элемента. Состояние до нажатия представлено на рисунке 2.7.

*Рисунок 2.7 – Формула до нажатия курсором мыши*

Состояние элемента после нажатия представлено на рисунке 2.8.

*Рисунок 2.8 – Формула после нажатия курсором мыши*

Следующей страницей является страница «Структура установки». Эта вкладка содержит изображение установки, используемой в разрабатываемой обучающей системе, озвученую карту установки, созданную в Adobe Animate, а также анимационый ролик, поясняющий принци работы установки. HTML разметка страницы со структурой представлена в приложении А данной записки.

На рисунке 2.9 – 2.12 показано, как выглядит страница «Структура установки».

*Рисунок 2.9 – Страница «Структура установки»*

*Рисунок 2.10 – Страница «Структура установки»*

*Рисунок 2.11 – Страница «Структура установки»*

*Рисунок 2.12 – Страница «Структура установки»*

К динамическим элементам на дайнно странице можно отнести, появлние картинки при нажатии на соответсвующий элемент списка элементов установки (рис. 2.13).

*Рисунок 2.13 – Отображение картинки при нажатии на соответствующий элемент списка элементов установки*

Четвертой страницей является страница «Симулятор установки». Эта страница представляет собой набор скриншотов работы симулятора установки, созданной в Unity. HTML разметка страницы с симулятором представлена в приложении А данной записки.

На рисунке 2.14 показано, как выглядит страница «Симулятор установки».

*Рисунок 2.14 – Страница «Симулятор установки»*

## **Озвученная карта установки и анимация алгоритма работы**

Создание озвученной карты установки и анимации алгоритма работы происходит в программном средстве Adobe Animate, которое позволяет создавать различные покадровые анимации [2]. Основой является изображение установки (рис. 2.14).

*Рисунок 2.14 – Схема установки*

Для того, чтобы элементы при нажатии воспроизводили звук, были записаны соответствующие аудиодорожки для каждого из элементов (библиотека со звуковыми элементами в Adobe Aminate представлена на рисунке 2.15).

*Рисунок 2.15 – Аудиодорожки в библиотеке*

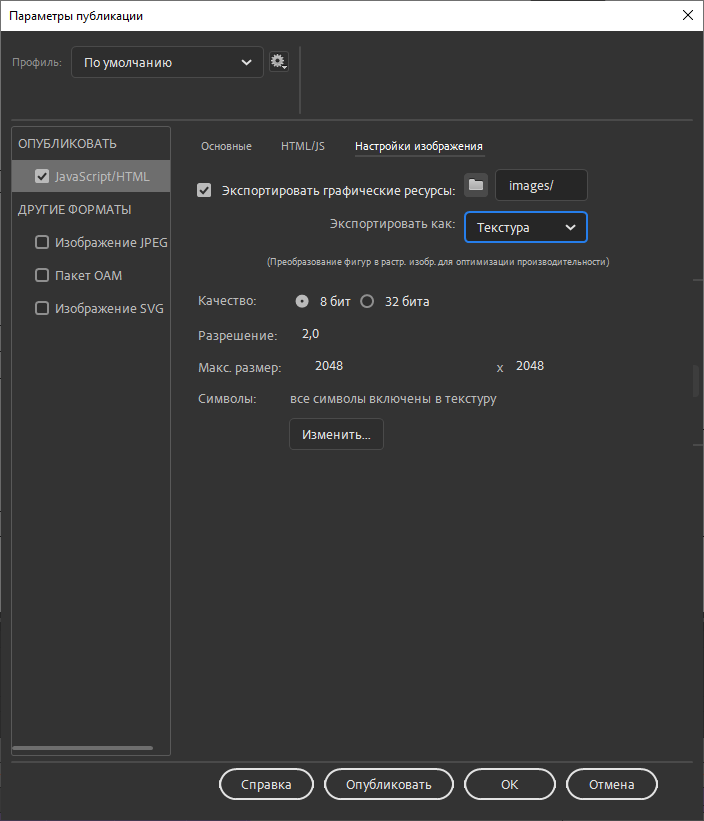
В итоге, после преобразования определенных элементов рисунка в кнопки, по наведению на которые воспроизводится записанные нами ранее аудиодорожки, получается озвученная карта установки, с помощью которой мы можем узнать название каждого элемента.

Далее было необходимо разработать анимацию алгоритма работы, которая показывала бы нам, какие действия необходимо выполнить, чтобы взаимодействовать с установкой. Во-первых, для этого мы разнесли по отдельным слоям каждый из тех элементов, который будет анимирован, и добавили на них аудиодорожки в зависимости от того, какой элемент анимируется и в какой момент времени (рис. 2.16).

*Рисунок 2.16 – Процесс анимации установки*

Для экспорта данной карты и анимации алгоритма работы на веб-сайт необходимо преобразовать их в HTML5 Canvas. После того, как программа соберет проект, его можно вставлять на основной веб-сайт.

Важной особенностью преобразования проекта в HTML5 Canvas является выставление некоторых параметров в «Параметрах публикации». В частности, необходимо графические элементы экспортировать как текстуру (рис. 2.17).



*Рисунок 2.17 – Параметры публикации в HTML5 Canvas*

## **Трехмерная модель установки**

Трехмерная модель установки создается в программном средстве 3D Studio Max 2021 [3]. За основу было взято изображение, представленное на рисунке 2.18.

*Рисунок 2.18 – Фотография установки*

При моделировании установки использовались следующие примитивы и модификаторы:

* Box (платформа, пластина);
* Cylinder (поршень, блок большого цилиндра, блок малого цилиндра, трубки, узел сброса давления, рычаг перемещения поршня, каркас, манометр);
* Loft (для придания объема линии, в нашем случае с целью сделать шланг);
* Capsule (для болтов, рычаг перемещения поршня);
* Extrude (для выдавливания плоских объектов);
* Sphere (впускной клапан, нагнетательный клапан, клапана узла сброса давления);
* Boolean (логический модификатор для объединения объектов);
* Bevel (для создания объемной модели с заданным сечением);
* Editable Poly.
* UVM Map.

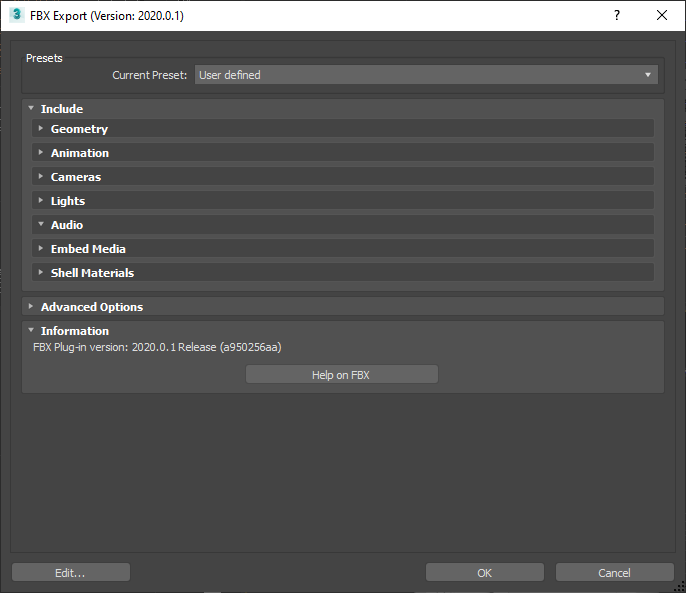
В итоге установка, смоделированная в 3D Studio Max 2022, выглядит так, как показано на рисунке 2.19.

*Рисунок 2.19 – 3D-модель установки*

Далее необходимо было смоделировать окружающую среду: кабинет и интерьер. Для этого мы использовали сторонние текстуры, часть из которых представлена в приложении Б. Кабинет с интерьером и установкой внутри представлен на рисунке 2.20.

*Рисунок 2.20 – 3D-модель установки и кабинета*

Для того, чтобы данную установку добавить на сцену в Unity, необходимо было экспортировать её в формате FBX, так как именно в этом формате можно удобно работать с 3D-моделями в Unity. Процесс публикации в FBX показан на рисунке 2.21.



*Рисунок 2.21 – Параметры публикации в FBX*

# **Описание хода работы 3D-симулятора с копиями экрана пользовательского интерфейса основных элементов системы, а также ключевых функций разработанного программного кода на языке C#**

3D-симулятор создается в среде разработки компьютерных игр Unity [4]. Перед тем, как разрабатывать основной функционал, необходимо импортировать 3D-модели, созданные в предыдущей главе

Импорт происходит путем перетаскивания файлов формата FBX в папку Assets проекта (рис. 3ю1).

Рисунок 3.1 – Папки с импортированными материалами

Далее необходимо выставить наши материалы на сцену. Для этого мы перетягиваем их из папки Models, где они хранятся, на саму сцену. Кабинет и установка будут выглядеть на сцене так, как показано на рисунке 3.2.

Рисунок 3.2 – Кабинет и установка на сцене

Теперь можно приступить к созданию интерфейса. Основными элементами интерфейса у нас будут являться:

* список элементов установки, по нажатию на каждый из них камера будет перемещаться к соответствующему элементу;
* область, в которой будет выводиться информация об элементах установки;
* область, в которой будет отображаться текущее задание;
* кнопка вызова таблицы, в которую записываются значения;
* таблица с числовыми значениями, которые определяются в ходе выполнения лабораторной работы;
* кнопка начала лабораторной работы;
* поле для ввода числовых значений.

Все эти элементы рисуются в Canvas – объекте, хранящий в себе 2D объекты. Следовательно, сначала необходимо создать его на сцене. Вместе с ним создается объект EventSystem, позволяющий обрабатывать события мыши со всеми элементами.

Созданный интерфейс представлен на рисунке 3.3.

Рисунок 3.3 – Созданный интерфейс

Когда интерфейс готов, необходимо реализовать функционал некоторых элементов. Например, при наведении мыши на кнопки в списке элементов установки у нас подсвечивается соответствующий элемент и выводится текст в информационное поле. Эти и многие дальнейшие функции реализованы с помощью компонента «EventTrigger», который позволяет вызывать определенные события в зависимости от действий пользователя (рис. 3.4).

*Рисунок 3.4 – Компонент EventTrigger*

Также было принято решение реализовать функцию перемещения камеры к определенному объекту по кнопки из списка «Установка». Эта функция реализована с помощью созданного нами скрипта «CameraMovement», который принимает на вход определенное положение камеры и по нажатию на кнопку перемещает камеру к определенному объекту. Скорость движения регулируется переменной speed, а перемещение и вращение – функциями Vector3.Lerp и Quaternion.Lerp соответственно. Скрипт «CameraMovement» представлен в листинге 3.1.

*Листинг 3.1 – Скрипт «CameraMovement»*

Необходимо создать статический класс, который показывал бы нам, выполнено ли определенное действие. Это необходимо для того, чтобы невозможно было делать задания не по порядку. Также добавим в этот скрипт порядковый номер эксперимента, так как их можно будет сделать несколько. Скрипт «StaticClassForTasks» представлен в листинге 3.2.

*Листинг 3.2 – Скрипт «StaticClassForTasks»*

Далее была реализована функция перемещения камеры внутри комнаты. Камера осуществляет движение по вертикали и горизонтали, а также выполняет поворот вокруг установки. Скрипт «ScriptForCamera» представлен в листигне 3.3

*Листинг 3.3 – Скрипт «ScriptForCamera»*

Для выделения элементов установки при наведении на соответсвующий элемент списка меню был реализован скрипт *«*ChangeColor*»* представленный в листигне 3.4

*Листинг 3.4 – Скрипт «СhangeColor»*

Для изменения цвета кнопок списка элементов установки был реализован скрипт *«*ChangeColorButton*»* представленный в листинге 3.5

*Листинг 3.5 – Скрипт «СhangeColorButton»*

Для отображаения окна информации, с основными принципами по работе с установкой, при наведении на кнопку с вопросом в левом верхнем углу и скрытии окно с информацие при уходе курсора мыши с его области был реализован скрипт *«*OpenCloseInfoWindow*»* представленный в листинге 3.6

*Листинг 3.6 – Скрипт «OpenCloseInfoWindow»*

Далее займемся разработкой скриптов непосредственно самих действий [5]. Каждый скрипт каждого действия будет привязан к элементу, с которым в процессе выполнения действия пользователь должен взаимодействовать. Таких скриптов на один элемент может быть несколько. Для того, чтобы не вешать несколько скриптов на один объект, будем объединять несколько действий, связанных с одним объектов, в один скрипт.

Основные анимации, с помощью которых в симуляторе реализованы действия:

* изменение местоположения объектов и их вращение;
* изменение текстур объектов;
* изменение текста;
* скрытие и появление основных элементов интерфейса.

Отслеживать выполнение действий будем через скрипт «StateVariables», который был описан выше. Перед каждым действием мы будем разрешать, пользователю выполнить его и запрещать выполнять предыдущее (все остальные действия первоночально ткже запрещены для выполнения). После того, как действие будет выполнено, будем ставить соответсвующей перменной false, а переменной следующего действия будет назначать true.

Значения, которые мы будем получать в процессе работы с симулятором, будем записывать в таблицу результатов. Для этого во время выполнения задания, в процессе которого пользователю необходимо снять какое-то показание с прибора и записать его в таблицу, будем давать ему поле для ввода, в которое он сможет вписывать определенное значение и сразу же юудем рассчитывать вычисляемые поля для всех столбцов соответствующей строки.

Все скрипты заданий представлены в приложении В данной записки.

# **Разработка базы знаний и режима диалога с информационной системой**

Одной из главных задач курсового проекта является создание базы знаний. Сама структура базы знаний представляет собой семантическую сеть, связывающую ключевые понятия смысловыми связями. Под семантической сетью подразумевается граф, в вершинах которого находятся информационные единицы – объекты, а дуги характеризуют отношения – связи между ними, которые интерпретируют эти сущности в заданной области.

Создание такой сети основывается на разбиении членов предложения на триады:

* блок существительного;
* блок глагола;
* блок дополнительных членов.

Для генерации таких триад исходный текст нужно упростить:

* заменив местоимения на соответствующее название объектов;
* удалив из текста обороты без смысловой нагрузки;
* разбив текст на простые предложения.

Основой базы знаний в моей мультимедийной системе является язык программирования JavaScript. Для реализации базы знаний создается отдельный файл формата .js. После формируется двумерный массив, хранящий триады. Фрагмент данного двумерный массив представлен на рисунке 4.1.

*Рисунок 4.1 – Фрагмент двумерного массива базы знаний*

Сама суть диалога с базой знаний заключается в получении ответа на основе заданного вопроса. Следовательно, необходимо реализовать функцию получения ответа.

Функция «getAnswer» принимает заданный вопрос. Дальше происходят следующие преобразования:

* в тексте вопроса дополняются различные символы препинания пробелом (точка, запятая, вопросительный знак, скобки и т.д.);
* все слова разбиваются в отдельный массив слов;
* все буквы в массиве слов преобразуются в нижний регистр.

После того, как преобразования готовы, необходимо сделать массив псевдоокончаний. Данный массив будет иметь различные окончания прилагательных и спряжений, которые будут заменяться на те, что могут быть в нашей базе знаний.

Далее на основе этих окончаний ведется поиск в ранее созданном массиве слов подлежащих, на основе которых будет вестись поиск в базе знаний. Формируется регулярное выражение найденного сказуемого и регулярное выражение найденных подлежащих (+ прилагательных).

В итоге, имея данные регулярные выражения, мы проходимся по базе знаний и ищем на основе наших регулярных выражений в триадах ответ на заданный вопрос.

Код скрипта, отвечающий за базу знаний, представлен в приложении В данной записки.

Затем необходимо интегрировать базу знаний в наш сайт. Для этого в заголовке HTML-документа пропишем связь между текущим документом и файлом knowledge.js, в котором находится база знаний. Не забываем отдельно задать дополнительные стили в файле styles.css, которые будут необходимы для красочного и внятного отображения базы. HTML-разметка интеграции модуля базы знаний представлена в листинге 4.1.

*Листинг 4.1 – Интеграция базы знаний в HTML-документ*

База знаний будет располагаться на странице «Теоретическое введение». Она будет появляться по нажатию на кнопку, расположенную в правой части страницы. Вид этой кнопки представлен на рисунке 4.2.

*Рисунок 4.2 – Кнопка открытия базы знаний*

Сам вид диалогового окна представлен на рисунке 4.3.

*Рисунок 4.3 – Диологовое окно бызы знаний*

Для создания нормальной базы знаний необходимо заполнить ей минимум на 100 триад. Данное заполнение позволит покрыть большинство пробелов в знаниях по данной тематике.

# **Руководство пользователя 3D-симулятором**

В данном руководстве описано, как нужно пользоваться симулятором установки. Начальный вид симулятора представлен на рисунке 5.1.

*Рисунок 5.1 – Начальный вид симулятора*

Для того, чтобы узнать краткую информацию о каждом элементе установки, необходимо навести курсор мыши на соответствующую кнопку в правом верхнем меню. В результате внизу рабочей области отобразиться окно с информацией о выбранном элементе установки, а на сцене подсветится интересующий нас элемент. Данная функция представлена на рисунке 5.2.

*Рисунок 5.2 – Вывод дополнительной информации и подсвечивание элементов*

Если пользователь не знает, где находится конкретный элемент, он может нажать на вышеупомянутые кнопки и камера переместится к тому элементу, кнопку которого он нажал. На рисунке 5.3 представлено, как это выглядит.

*Рисунок 5.3 – Перемещение камеры к определенному элементу*

По нажатию на кнопку «Практика» начинается выполнение практической работы. Появляется нижняя информационная панель, которая говорит пользователю, что он должен сделать. Данная панель представлена на рисунке 5.4.

*Рисунок 5.4 – Начало практической работы*

В процессе выполнения лабораторной работы пользователю будет необходимо вводить определенные значения в поле для ввода. Оно появляется в тот момент, когда это необходимо сделать, чтобы пользователь не вводил лишнюю информацию. Вид поля для ввода представлен на рисунке 5.5.

*Рисунок 5.5 – Ввод значений*

После выполнения лабораторной работы в нижнем информационном поле появляется сообщение, что пользователь закончил выполнение работы. Вид симулятора после выполнения лабораторной работы представлен на рисунке 5.6.

Рисунок 5.6 – Конец лабораторной работы

Итогом выполнения работы является заполненная таблица значений. Стоит отметить, что таблицу можно открывать на протяжении всего опыта, как и перемещать камеру. Заполненная таблица после четырех опытов представлена на рисунке 5.7.

*Рисунок 5.7 – Итоговая таблица*

# **Заключение**

В результате выполнения данного курсового проекта была разработана полноценная компьютерная обучающая мультимедийная система, позволяющая реализовать рактическую сдавливанию элементов, состоящих из различных материалов. Курсовая работа была выполнена с помощью таких программных средств, как Unity, 3D Studio Max 2022, Adobe Animate, Visual Studio 2019, Visual Studio Code.

Данная компьютерная обучающая мультимедийная система состоит из нескольких обучающих модулей разной направленности, заключенных в одном веб-сайте:

* раздел теоретическими сведениями, подробно описывающий все основные и дополнительные аспекты работы с установкой;
* раздел с визуальным отображением структуры установки в виде озвученной карты и видеоролика, описывающий процесс работы с установкой;
* раздел с симулятором установки, созданный в среде Unity.

Такая структура мультимедийной системы позволяет подробно изучить данную тематику без необходимости поиска дополнительных источников информации, а также дает возможность закрепить изученный теоретический материал на практике с помощью симулятора.

# **Список использованных источников**

[1] Гурин Н.И. Компьютерные обучающие системы в издательском деле. В 2 ч. Ч. 1. Разработка компьютерной обучающей среды: учеб. пособие / Н. И. Гурин, О. В. Герман. - Минск: БГТУ, 2015.

[2] Animate Tutorials [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://helpx.adobe.com/animate/tutorials.html> Время доступа: 07.10.2021

[3] 3ds Max Features [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.autodesk.com/products/3ds-max/features> Время доступа: 10.10.2021

[4] Unity Forum [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://forum.unity.com/> Время доступа: 26.10.2021

# **ПРИЛОЖЕНИЕ А**

Файл СSSForTitlPage.css

Файл TheoreticalInformation.html

Файл CSSForTheoreticalInformation.css

Файл Structure.html

Файл CSSForStructure.css

Файл Simulator.html

Файл MenuPage.html

Файл CSSForMenuPage.html

Файл JSForStructurs.js

Файл JSForTheoreticalInformation.js

# **ПРИЛОЖЕНИЕ Б**

Файл ActivDisactivButton.cs

Файл AddRemoveInfoIntoTable.cs

# **ПРИЛОЖЕНИЕ В**

Файл KnowledgeBase.js