

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
**"САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО"**  
Институт компьютерных наук и технологий  
Направление **02.03.01** : Математика и компьютерные науки

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ  
«Моделирование в среде UNIGINE»

Исполнитель: \_\_\_\_\_

Яшнова Дарья Михайловна  
группа 5130201/20002

Руководитель: \_\_\_\_\_

Курочкин Михаил Александрович

« \_\_\_\_ » \_\_\_\_\_ 2025г

Санкт-Петербург, 2025

# Содержание

<b>Введение</b>	<b>3</b>
<b>1 Постановка задачи</b>	<b>4</b>
<b>2 Описание Unigine</b>	<b>5</b>
<b>3 Устройство проекта UNIGINE</b>	<b>6</b>
3.1 Система координат . . . . .	6
3.2 Мир . . . . .	6
3.3 Узел . . . . .	6
3.4 Объект . . . . .	7
3.5 Мэш . . . . .	7
3.6 Поверхность и материал . . . . .	7
3.7 Симуляция физики . . . . .	8
<b>4 Описание сценария</b>	<b>10</b>
<b>5 Реализация сцены</b>	<b>11</b>
<b>6 Результаты работы</b>	<b>21</b>
6.1 Падение в воду . . . . .	21
6.2 Падение на плоскость . . . . .	23
<b>7 Выводы</b>	<b>25</b>
<b>Список литературы</b>	<b>26</b>

## Введение

Графический движок — это специализированное программное обеспечение, предназначенное для управления и визуализации графики в различных приложениях: видеоиграх, симуляторах, анимации, VR-проектах и других интерактивных системах. Он предоставляет разработчикам готовые инструменты и функции, упрощающие создание сложных визуальных эффектов и реалистичных сцен, что значительно ускоряет процесс разработки.

Хотя чаще всего графические движки ассоциируются с игровой индустрией, их возможности востребованы и в других сферах, таких как архитектурная визуализация, кинопроизводство и научные симуляции. Многие движки включают в себя готовые ресурсы, например, текстуры (воды, металла, дерева) и шаблоны материалов, а также поддерживают физические свойства объектов — гравитацию, столкновения, динамику жидкостей и тканей. Это позволяет разработчикам сосредоточиться на творческой составляющей, не тратя время на реализацию базовых механизмов с нуля.

# 1 Постановка задачи

**Целью** данной работы является изучение графического движка UNIGINE.

**Задачи:**

- Ознакомиться с 3D-движком UNIGINE и возможностями, которые он предоставляет;
- Импортировать в UNIGINE модели, разработанные в ходе выполнения лабораторной работы №1;
- Создать сцену с импортированными объектами.

## 2 Описание Unigine

Unigine Engine – это многофункциональный 3D-движок, предназначенный для создания широкого спектра приложений, включая игры, системы виртуальной реальности, интерактивные визуализации и различные типы тренажеров (обучающие, медицинские, военные, транспортные).

Unigine Engine обеспечивает кроссплатформенную совместимость с Microsoft Windows и GNU/Linux, предоставляя разработчикам возможность работать на обеих операционных системах. Он поддерживает графические API Direct3D и OpenGL. Движок использует собственный скриптовый язык UnigineScript, который является объектно-ориентированным и напоминает C++, что облегчает разработку и настройку игровой логики.

Unigine Engine предоставляет передовые возможности для реализации визуальных эффектов, таких как Screen Space Ambient Occlusion (SSAO) и расчет глобального освещения в реальном времени, позволяя создавать высококачественную и реалистичную графику. Встроенный модуль физической симуляции обеспечивает реалистичное моделирование столкновений, физики твердых тел, динамического разрушения объектов, симуляции тканей, плавучести, полей сил и даже обратного хода времени.

Для создания реалистичных ландшафтов и окружения Unigine Engine включает инструменты для генерации рельефа местности и реализации систем растительности, что особенно полезно в симуляторах и играх с открытым миром.

Важной особенностью Unigine является наличие интуитивно понятного редактора виртуального мира, который упрощает процесс создания и редактирования сцен и игровых миров.

## 3 Устройство проекта UNIGINE

Разработка приложений в UNIGINE осуществляется в рамках проекта, который объединяет код, контент и метаданные разрабатываемого приложения. Проект может содержать одну или несколько многокомпонентных 3D-сцен, называемых мирами (worlds). Управление проектами осуществляется с помощью UNIGINE SDK Browser.

### 3.1 Система координат

Трехмерное пространство виртуального мира описывается декартовой системой координат, ориентированной по правилу правой руки: оси X и Y лежат в горизонтальной плоскости, ось Z направлена вверх.

При экспорте 3D-моделей и анимации из 3D-редакторов ось Y используется как вектор направления. Положительные углы вращения соответствуют вращению вектора против часовой стрелки, в соответствии с правилом буравчика (или правой руки).

### 3.2 Мир

Мир в UNIGINE представляет собой трехмерную сцену, служащую пространством для визуализации и взаимодействия. Сцена состоит из узлов (nodes) – элементов, формирующих окружение. Это 3D-модели, источники света, камеры и другие объекты, размещенные в сцене для создания визуальной композиции.

Сцена в UNIGINE организована в виде иерархического дерева узлов, называемого графом сцены. Каждый узел может содержать дочерние узлы, что позволяет устанавливать сложные отношения между объектами. Например, узел, представляющий автомобиль, может иметь дочерние узлы для колес, двигателя и других деталей.

Информация о каждом мире сохраняется в XML-файле с расширением .world. Этот файл содержит данные о расположении и параметрах всех узлов сцены, а также глобальные настройки, такие как параметры визуализации и физики, применяемые ко всему миру.

При создании нового проекта в UNIGINE через SDK Browser автоматически создается мир с именем проекта. Этот мир можно редактировать и создавать дополнительные миры для проекта с помощью UnigineEditor.

### 3.3 Узел

В UNIGINE узлы (nodes) – это базовые элементы, добавляемые в виртуальную сцену. Различные типы узлов определяют внешний вид и поведение объектов. Встроенные типы узлов предлагают широкий набор объектов и эффектов. Пользовательские узлы можно создавать и расширять их функциональность, добавляя компоненты, обогащающие узлы свойствами и поведением.

Каждый узел имеет матрицу преобразования, определяющую его положение, ориентацию и масштаб в виртуальном мире, обеспечивая точный контроль над расположением и ориентацией объектов. Некоторые узлы визуально отображаются как объекты, декали или эффекты, а другие, такие как источники света и игровые объекты, могут быть невидимыми, но при этом играть важную роль в виртуальной среде.

Параметры каждого узла сохраняются в файле .world, но их также можно сохранить в отдельных XML-файлах с расширением .node, что полезно для создания нескольких узлов с одинаковыми настройками и ссылками на них из основного файла .world. Это достигается с помощью узлов "Node References позволяющих создавать множественные экземпляры одного и того же узла.

### 3.4 Объект

Объекты – это один из основных типов узлов в UNIGINE, моделирующие сущности, существующие в реальном мире, такие как предметы (люди, деревья, автомобили, самолеты и т. д.), небо, рельеф, вода и т. п. Объекты состоят из одной или нескольких поверхностей, определяющих их внешний вид. Они также могут иметь форму, представляющую занимаемый объем в трехмерном пространстве, и тело, взаимодействующее с физическим окружением.

### 3.5 Мэш

Мэш (mesh) – это базовая геометрическая структура объекта в трехмерной сцене, состоящая из набора многоугольных полигонов, определяющих его форму и внешний вид.

Каждый мэш может содержать одну или несколько поверхностей, определяющих внешний вид объекта. Поверхности состоят из многоугольников, и максимальное количество многоугольников на поверхности зависит от типа мэша. Статические и скелетные мэши могут содержать до 2 147 483 647 многоугольников на поверхности, а динамические мэши ограничены 65 535 многоугольниками на поверхности.

Мэш поддерживает два UV-канала, используемых для нанесения текстур на объект, определяющих, как текстуры будут отображаться на поверхности. Мэш также хранит информацию о цвете каждой вершины, что позволяет создавать разнообразные визуальные эффекты и контролировать цвет объектов.

Анимация объектов может быть реализована различными способами, включая обтекаемые сетки с использованием костей, морф-цели для создания ключевой анимации и динамические сетки, управляемые программно. Анимационные данные мешей сохраняются в форматах UNIGINE с расширениями .mesh (для статических и скелетных мешей с анимацией) и .anim (для внешних анимационных данных).

### 3.6 Поверхность и материал

Поверхность в UNIGINE – это именованное подмножество геометрии объекта (его сетки), не имеющее самопересечений. Каждая поверхность может иметь собственный материал или свойство, что позволяет индивидуально настраивать внешний вид и характеристики. Независимое включение и выключение поверхностей предоставляет гибкость в управлении визуальным представлением объекта.

Поверхности можно организовать в иерархию внутри общей геометрии объекта, например, для управления уровнем детализации (LOD). Переключение между версиями объекта с разным количеством и детализацией поверхностей позволяет эффективно управлять ресурсами в зависимости от расстояния до наблюдателя или других факторов.

Материал в UNIGINE – это набор правил, определяющих внешний вид поверхности объекта, включая поведение при взаимодействии с освещением, параметры отражения и другие характеристики. Материал состоит из следующих основных компонентов:

- Шейдеры: программы, определяющие, как будет выглядеть материал при рендеринге. В UNIGINE используются вершинные, фрагментные и геометрические шейдеры, работающие на уровне пикселей и вершин, а также манипулирующие геометрией объекта.
- Текстуры: изображения, применяемые к материалу и передаваемые шейдерам, используемые для создания деталей, цвета или блеска поверхности.

- Состояния: условия, определяющие, какие шейдеры будут активированы в разных ситуациях (например, разные шейдеры для отражающих и матовых поверхностей).
- Параметры: значения, позволяющие настраивать поведение шейдеров и их воздействие на материал (например, интенсивность блеска или степень прозрачности).

### 3.7 Симуляция физики

- Имитация идеально упругих столкновений: Реализована имитация столкновений, при которых кинетическая энергия сохраняется, позволяя объектам обмениваться энергией.
- Имитация механизмов с использованием твердых тел и разрушаемых соединений: UNIGINE позволяет создавать механизмы и соединения между объектами, которые могут быть неразрушимыми или разрушаемыми, позволяя имитировать механические системы (подвески, шарниры) и динамические разрушения объектов.
- Имитация основных физических явлений: UNIGINE включает имитацию гравитации, трения и выталкивающих сил. Гравитация определяет вертикальное движение, трение влияет на движение по поверхности, а выталкивающие силы могут имитировать аэродинамическое воздействие.
- Процедурное разрушение мэшей: Возможность процедурного разрушения геометрических объектов (мэшей) позволяет создавать эффект разрушения и деформации объектов в реальном времени. Это полезно для создания сценариев с разрушаемыми зданиями или объектами.

Физика объектов в UNIGINE строится на основе тел и форм, позволяющих имитировать различные физические свойства и взаимодействия. Для задания физики объекту необходимо назначить тело.

Тело (Body) представляет собой основную часть объекта, имитирующую его физическое поведение. UNIGINE поддерживает следующие типы тел:

- Твердое тело (Rigid Body): Имитирует жесткое поведение объекта, взаимодействующего с другими твердыми телами и внешними силами.
- Тряпичная кукла (RagDoll): Представляет объект, который может деформироваться и гибко реагировать на силы, что подходит для имитации мягких тканей или одежды.
- Разрушаемое тело (Fracture): Позволяет объекту разрушаться на части при воздействии силы или столкновении, полезно для разрушаемых зданий.
- Шнур (Rope): Представляет собой гибкий объект для создания веревок, цепей или других подвижных элементов.
- Ткань (Cloth): Позволяет создавать объекты, деформирующиеся под воздействием сил, имитируя текстильные материалы.
- Вода (Water): Представляет собой специальное тело, имитирующее поведение жидкости, включая течение, взаимодействие с объектами и плавание.
- Траектория (Path): Позволяет объекту двигаться по заранее заданной траектории для анимации движения.



Форма объекта определяет его коллизионную геометрию и, следовательно, то, как объект будет взаимодействовать с окружающими объектами и физическими силами. UNIGINE поддерживает следующие формы объектов: сфера (sphere), капсула (capsule), цилиндр (cylinder), куб (box), выпуклый многоугольник (convex hull).

Каждая форма может быть ассоциирована с телом объекта и определять его коллизионные свойства, позволяя точно определить, как объект будет взаимодействовать с другими объектами и средой.

## 4 Описание сценария

### **Сцена с падением на плоскость**

В сцене участвуют три твердых объекта: губка, карандаш и ролик для одежды. Все три объекта изначально находятся на высоте падают на твердую поверхность. Губка и ролик сталкиваются друг с другом в процессе полета, а после этого все предметы упруго сталкиваются с поверхностью.

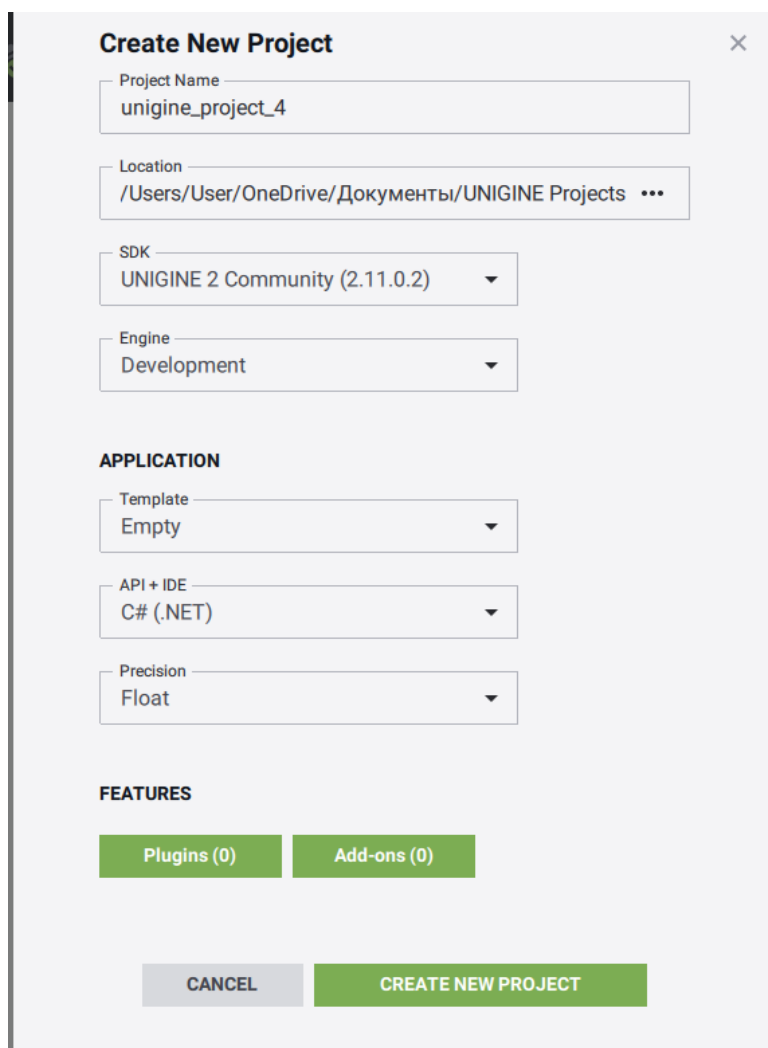
### **Сцена с падением в воду**

В сцене участвуют три твердых объекта: губка, карандаш и ролик для одежды. Все три объекта изначально находятся на высоте падают в воду. Губка после всплывает и выскакивает из воды, карандаш и ролик тонут, но ролик продолжает двигаться по течению.

## 5 Реализация сцены

Для начала работы с UNIGINE необходимо установить его основные элементы: UNIGINE SDK и UNIGINE SDK Browser. В данной лабораторной работе используется SDK 2.18.1 версии и UNIGINE SDK Browse версии 2.0.15. Все необходимые для разработки компоненты можно загрузить с сайта <https://unigine.com/get-unigine/>.

После установки и авторизации необходимо создать проект. UNIGINE проект — это "контейнер" для кода, содержимого и мета-данных приложения. При создании проекта можно указывать необходимые API + IDE, точность, SDK и вид версии (рис. 1).



**Create New Project** [X]

Project Name  
unigine\_project\_4

Location  
/Users/User/OneDrive/Документы/UNIGINE Projects ...

SDK  
UNIGINE 2 Community (2.11.0.2)

Engine  
Development

**APPLICATION**

Template  
Empty

API + IDE  
C# (.NET)

Precision  
Float

**FEATURES**

Plugins (0) Add-ons (0)

CANCEL CREATE NEW PROJECT

Рис. 1: Создание проекта

Затем открываем проект в редакторе рис. 2.

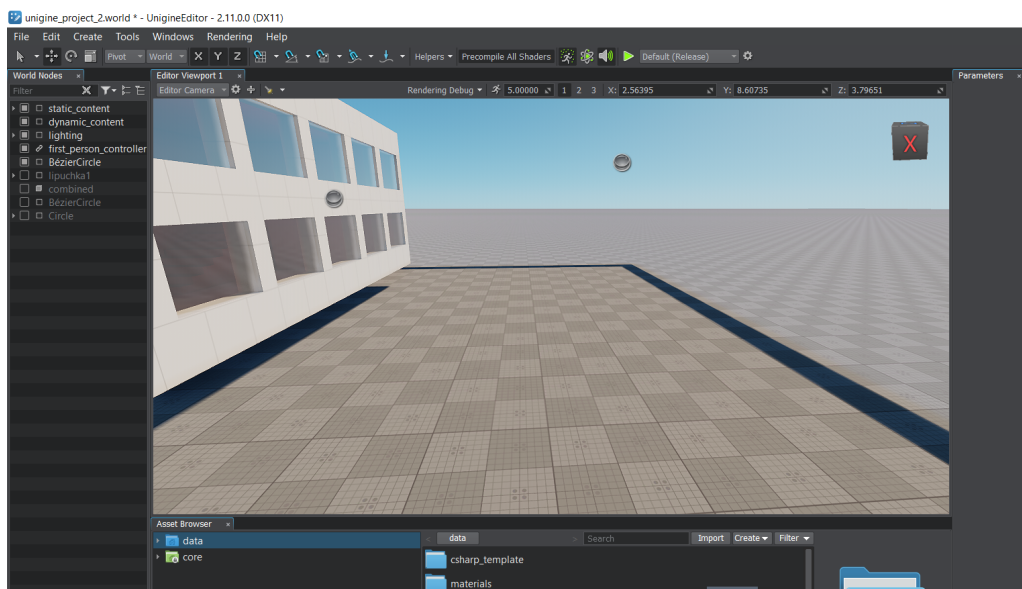


Рис. 2: Проект в редакторе

С предметами полученными в первой лабораторной работе реализуем сцену в UNIGINE. (рис. 3).

Важно помнить, что размер объекта должен соответствовать его реальному размеру и к предмету перед экспортированием должны быть применены все виды трансформации. Если размер не соответствует реальному, то нужно изменить размер. Это необходимо для того, чтобы масштаб был равен 1.

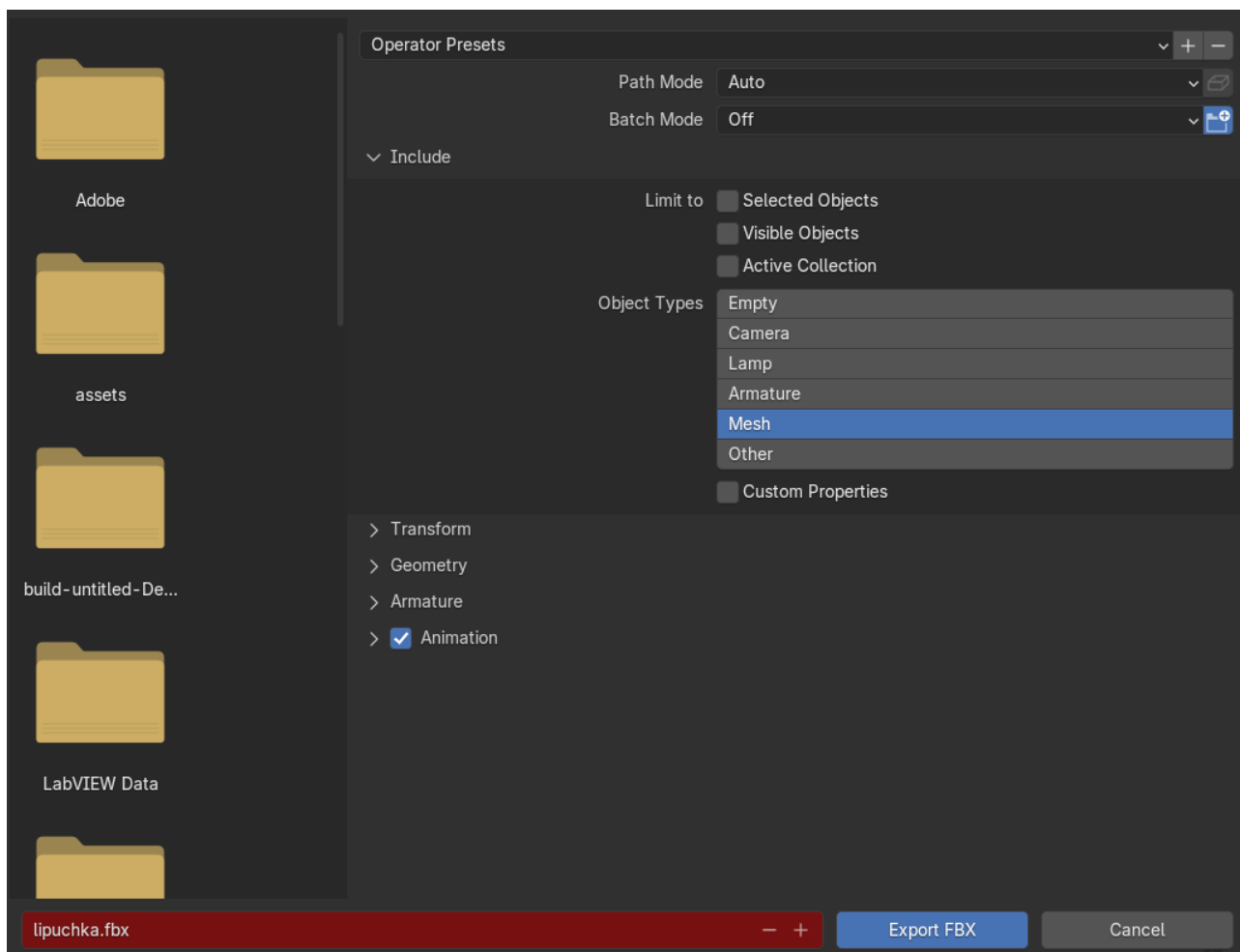


Рис. 3: Экспорт модели из Blender

Далее все предметы необходимо импортировать. Для этого в окне Asset Browser создаем по новой папке для каждого объекта. Заходим в папку, нажимаем ПКМ->импортировать и задаем параметры, как на рис.4.

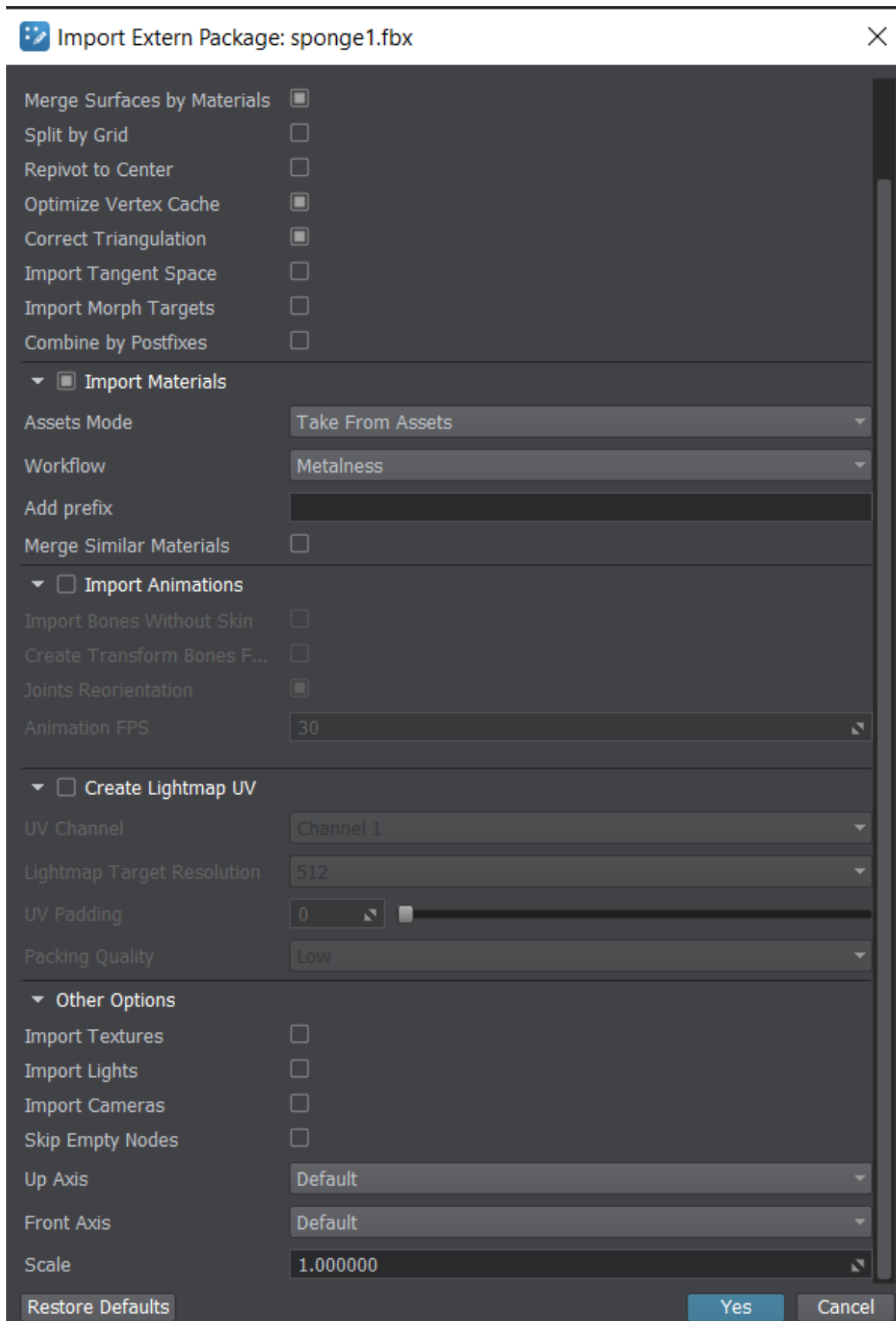


Рис. 4: Импорт модели в UNIGINE

Для того, чтобы модели появились на сцене необходимо зажать ЛКМ на .fbx файле и перетянуть этот файл на рабочую поверхность.

Для того что бы объекты могли взаимодействовать с поверхностью и друг с другом,

необходимо задать физику объектов. Для этого кликаем ЛКМ на объект, далее нужно открыть вкладку Parameters->Node и выбрать Dynamic.

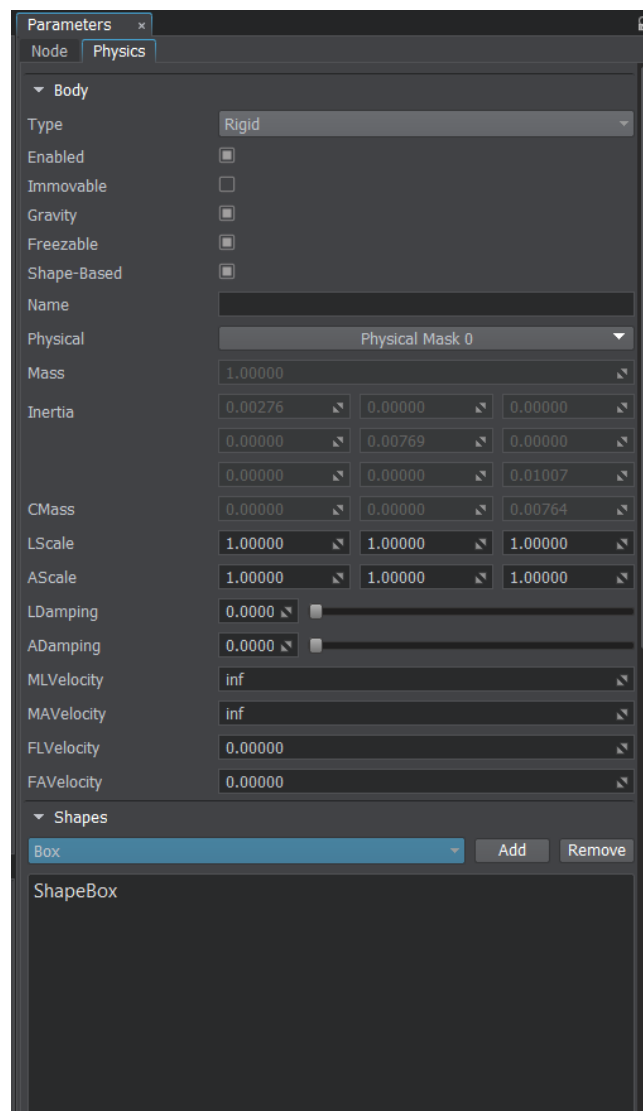


Рис. 5: Настройка физики губки

Центр масс (CMass) смещен по оси Z (0.00764), что означает, что объект может опрокидываться при падении, если физика включена.

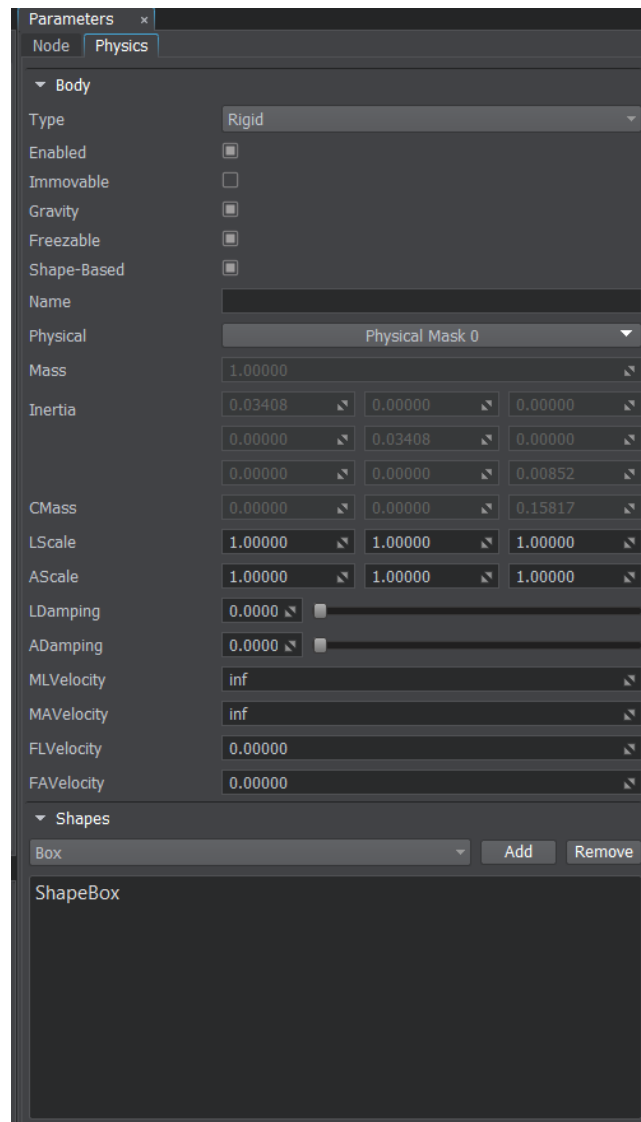


Рис. 6: Настройка физики ролика для одежды

Центр масс (CMass) смещен по оси Z (0.15817), что означает, что объект может опрокидываться при падении, если физика включена.



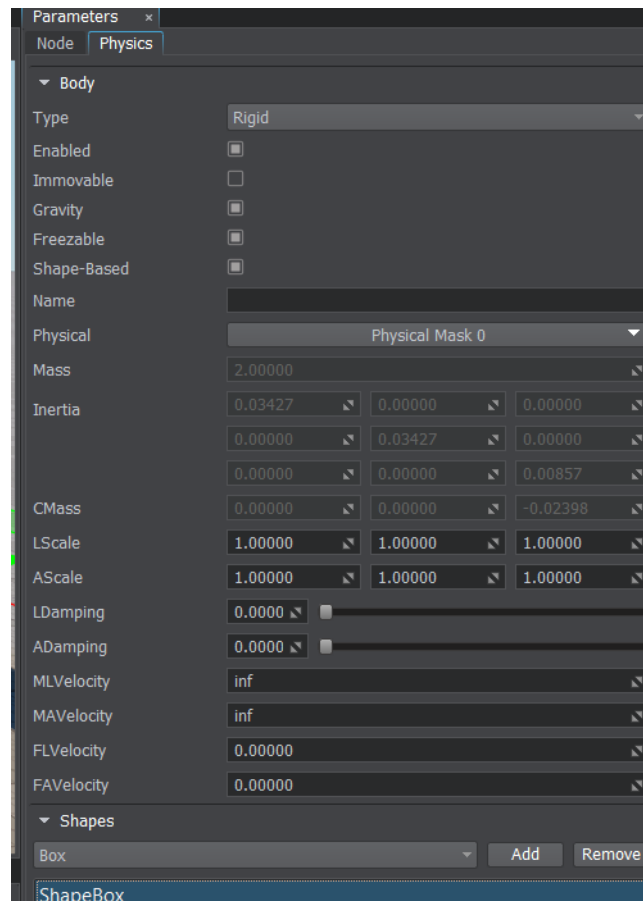


Рис. 7: Настройка физики карандаша

Центр масс (CMass) смещен по оси Z (-0.02398), что означает, что объект может опрокидываться при падении, если физика включена.

- Тип тела (Body Type):

Объект имеет тип Rigid, что означает, что он будет подчиняться законам физики, включая столкновения, гравитацию и инерцию.

- Масса и инерция (Mass & Inertia):

Масса объекта: 1.00000 (стандартное значение).

Момент инерции (матрица 3x3) указывает на то, как объект будет вращаться. В данном случае он симметричен по осям X и Y, но меньше по оси Z, что типично для продолговатых объектов (например, прямоугольной формы).

- Форма коллизии (Shapes):

У объекта есть форма коллизии Box (куб). Это значит, что при столкновениях он будет вести себя как прямоугольный объект, даже если его визуальная модель сложнее.

- Демпфирование (Damping):

Линейное (LDamping) и угловое (ADamping) демпфирование установлены в 0, поэтому объект не будет терять энергию при движении или вращении (если физика включена).

- Скорости (Velocity):

Максимальные линейная (MLVelocity) и угловая (MAVelocity) скорости не ограничены (Inf).

Важно не забыть, что у поверхности тоже должна быть физика. Для настройки переходим во вкладку Parameters->Physics. Для поверхности необходимо установить Rigid body, добавляем пункт Immovable. Это необходимо, чтобы поверхность не двигалась, но могла взаимодействовать с моделями. В качестве формы объекта выбираем Box.

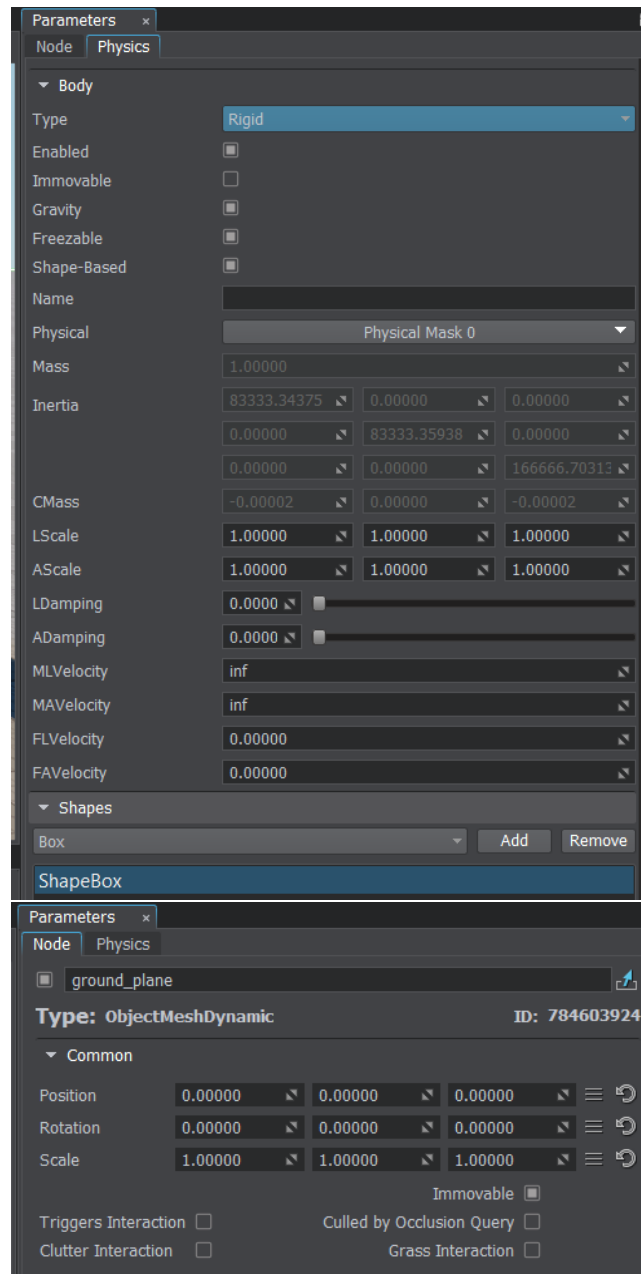


Рис. 8: Настройка физики плоскости

Теперь необходимо расставить объекты в пространстве так, как было задумано в нашей сцене. Для этого переходим во вкладку Parameters->Node и устанавливаем следующие параметры:

Помимо сцены с падением предметов на плоскость была разработана сцена с падением предметов в воду. Для того чтобы создать воду надо нажать Create->Water->Physical Water. Далее в окне Parameters задать воде скорость и плотность. В зависимости от

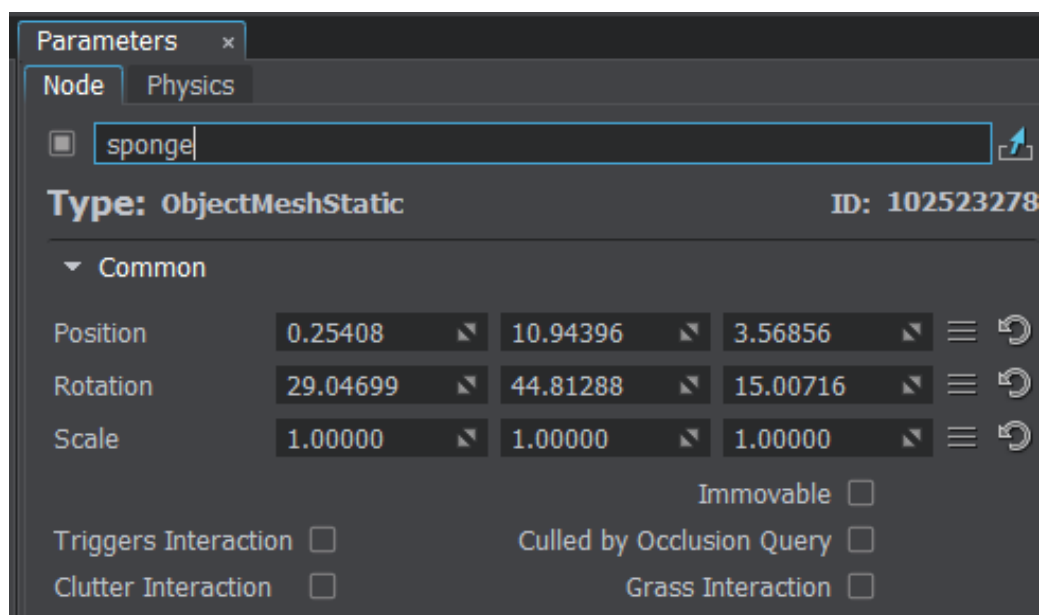


Рис. 9: Настройка положения губки

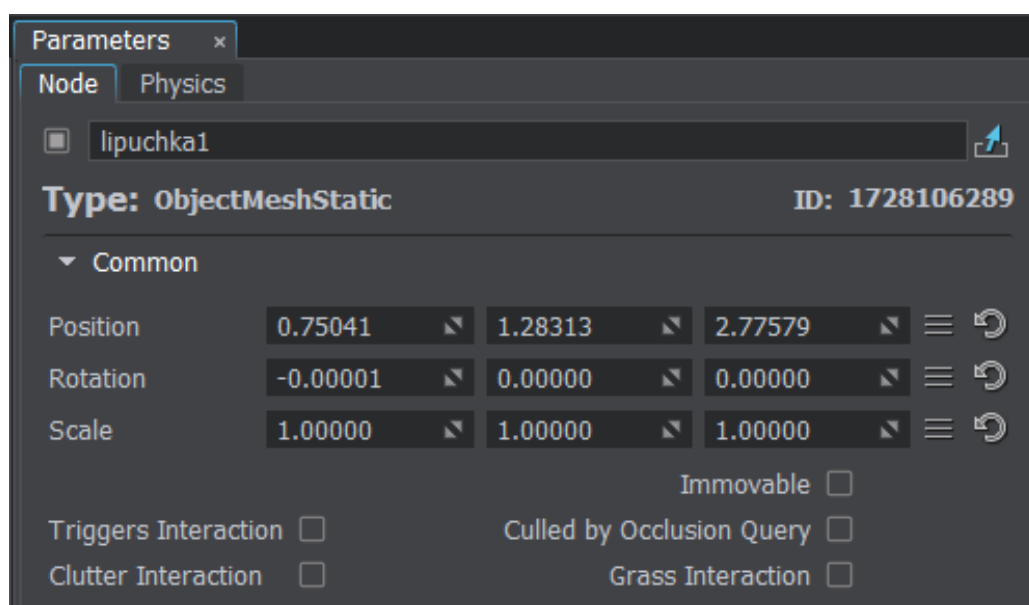


Рис. 10: Настройка положения липучки

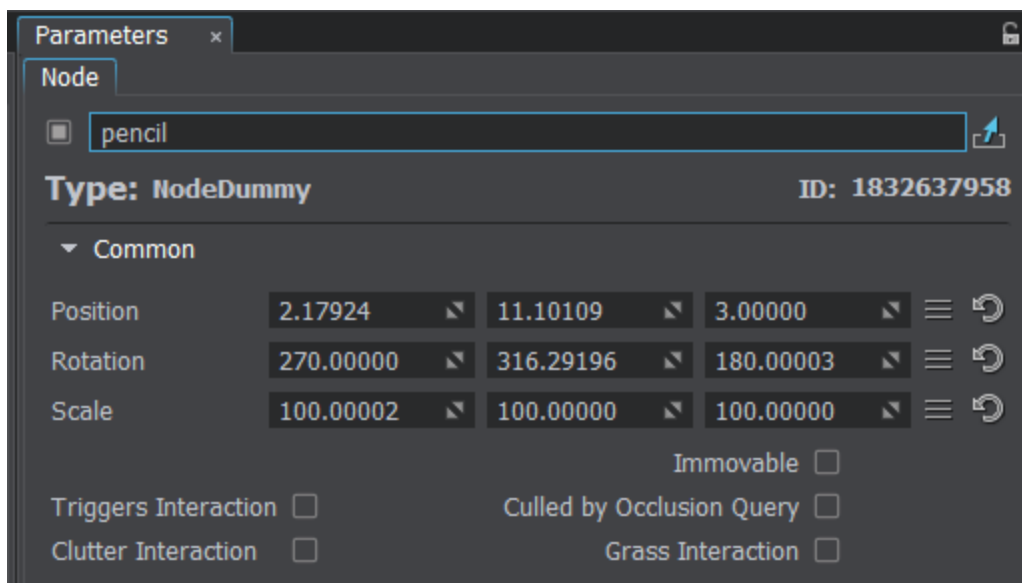


Рис. 11: Настройка положения карандаша

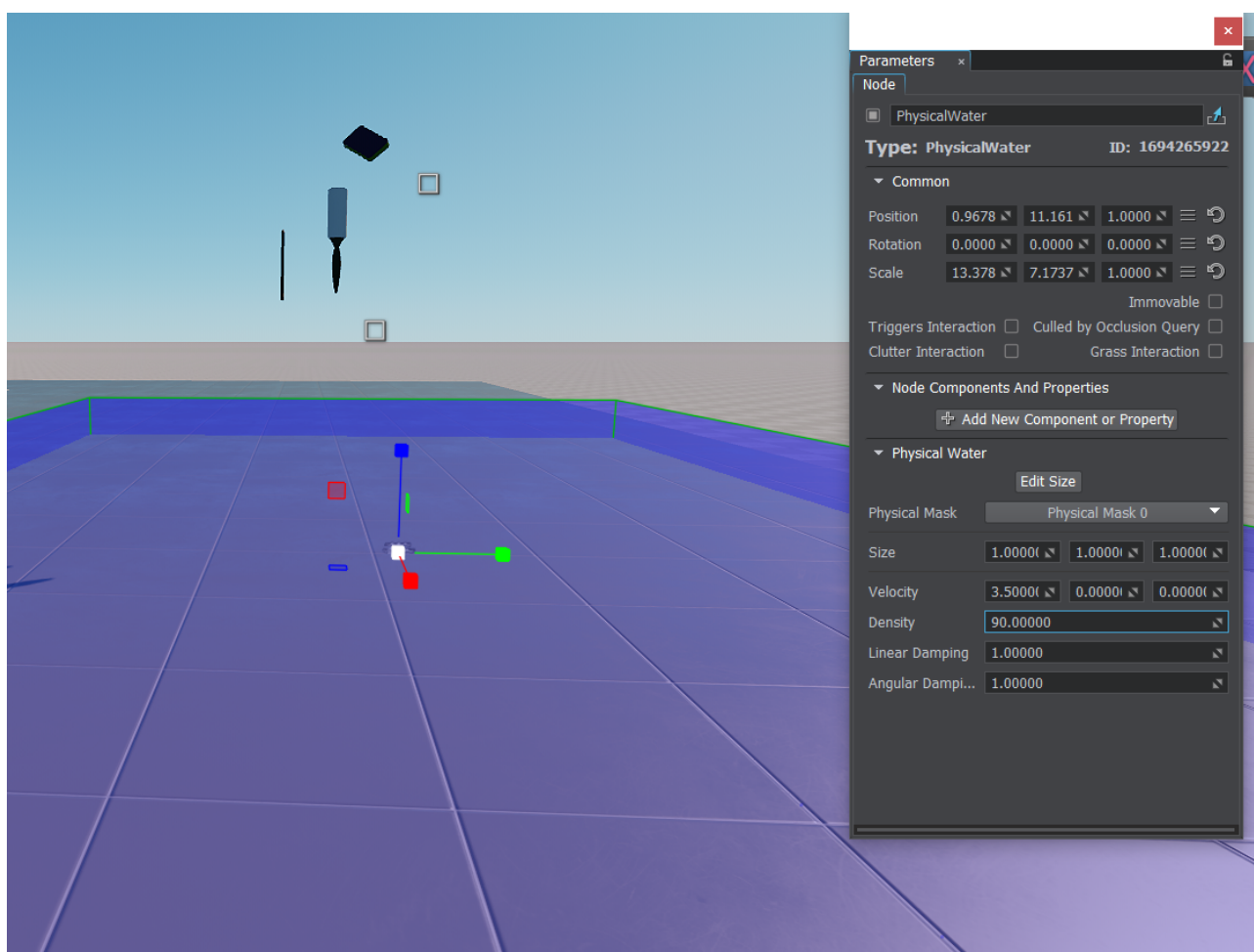


Рис. 12: Настройка свойств воды

свойств предмета разные предметы будут вести себя в воде по разному. Карандаш упадет на дно, губка утонет, а потом всплывет и уплывет назад по оси X, а ролик для одежды утонет на дно и начнет двигаться вдоль оси X по дну.

## 6 Результаты работы

### 6.1 Падение в воду

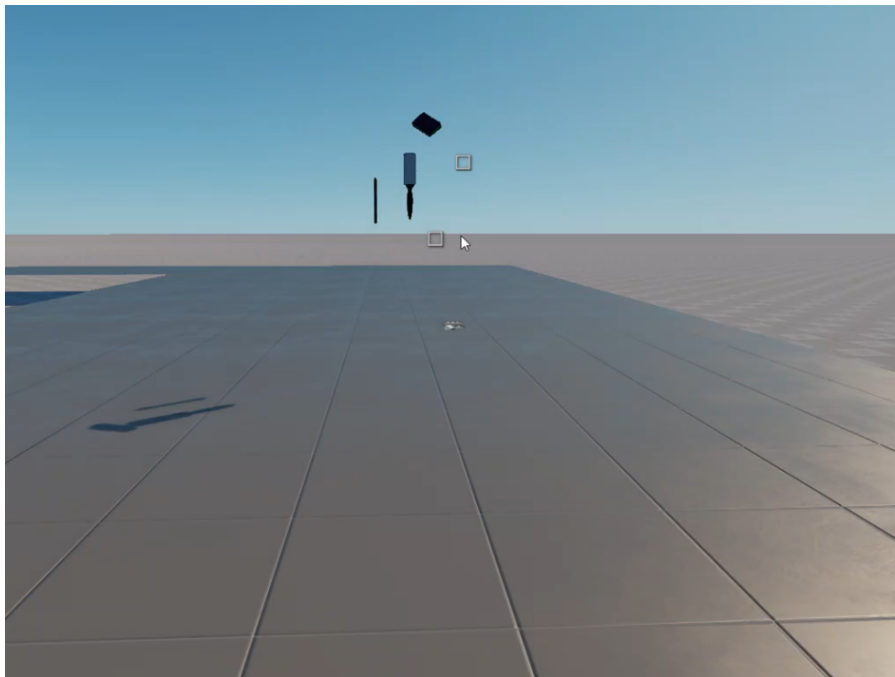


Рис. 13: Начальное положение предметов

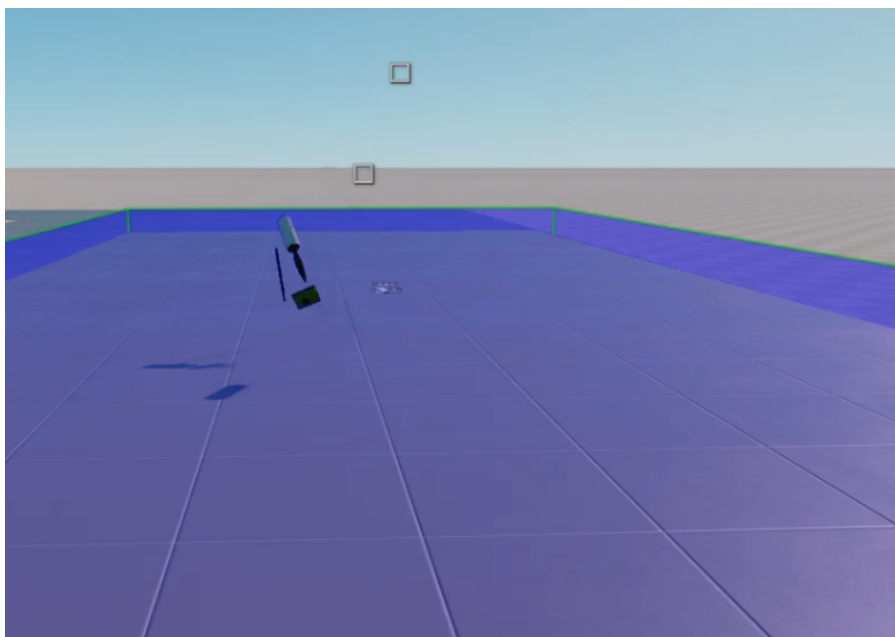


Рис. 14: Все предметы достигли поверхности воды

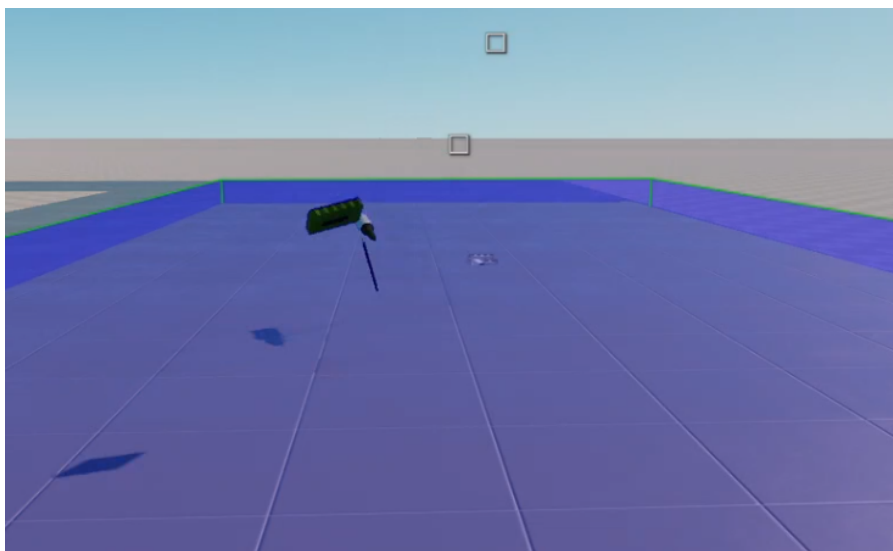


Рис. 15: Карандаш и ролик начинают тонуть, а губка начинает всплывать

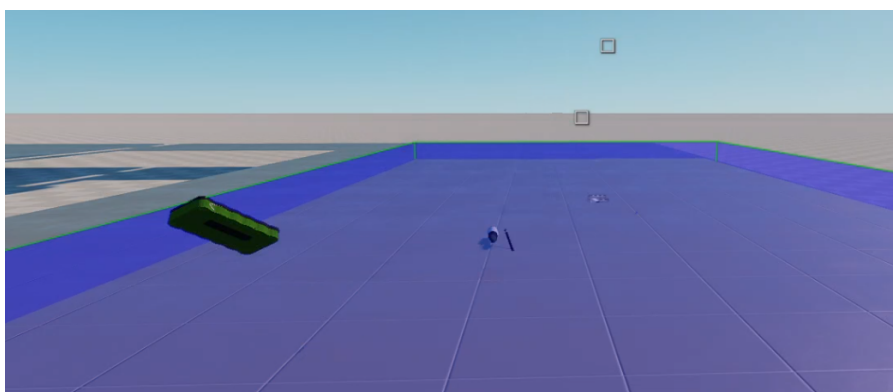


Рис. 16: Карандаш и губка продолжают тонуть, а губка выскочила с поверхности воды

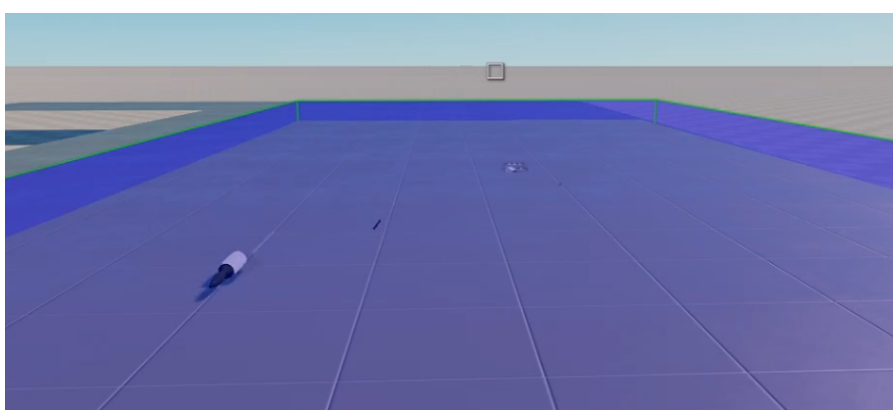


Рис. 17: Ролик продолжает двигаться назад по оси x

## 6.2 Падение на плоскость

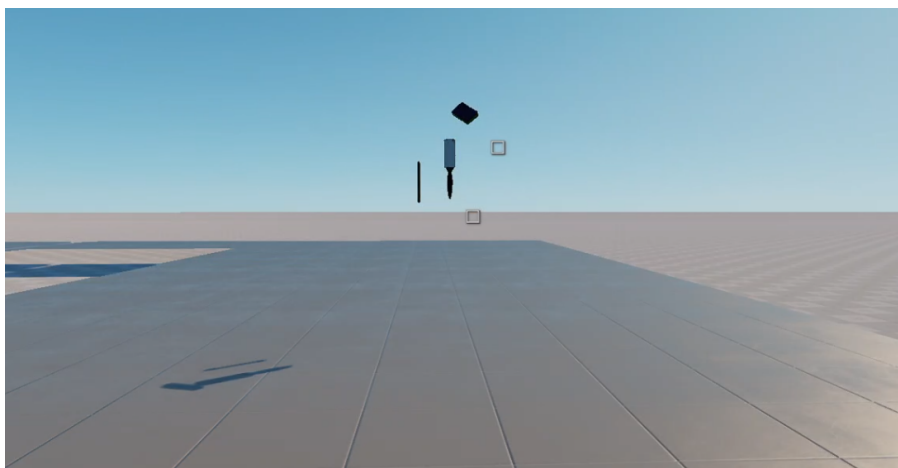


Рис. 18: Начальное положение предметов

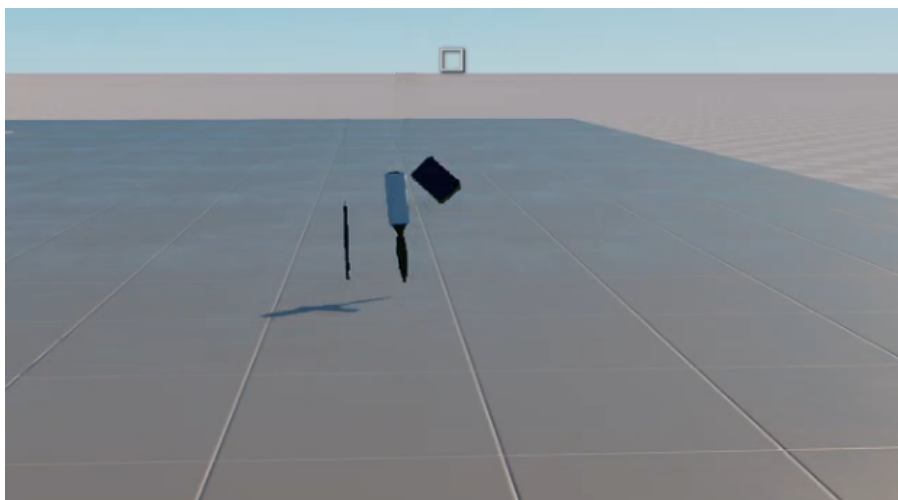


Рис. 19: Предметы в процессе падения

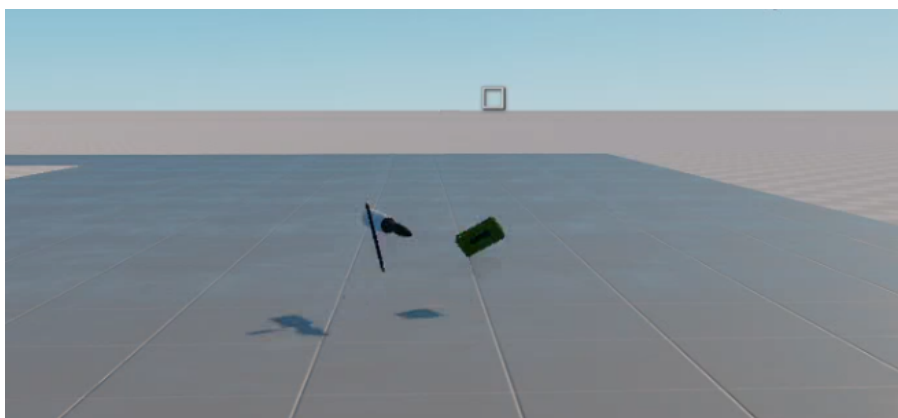


Рис. 20: Предметы после столкновения ролика и губки

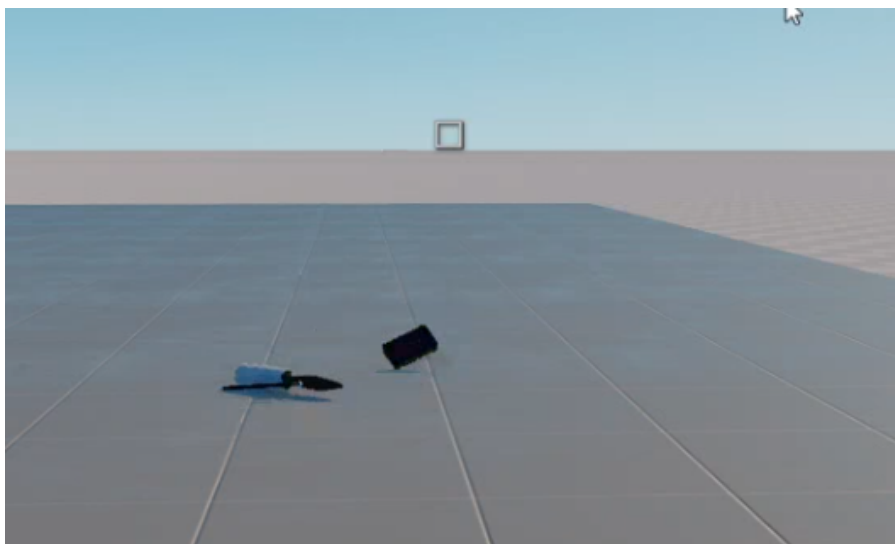


Рис. 21: Предметы после столкновения с поверхностью

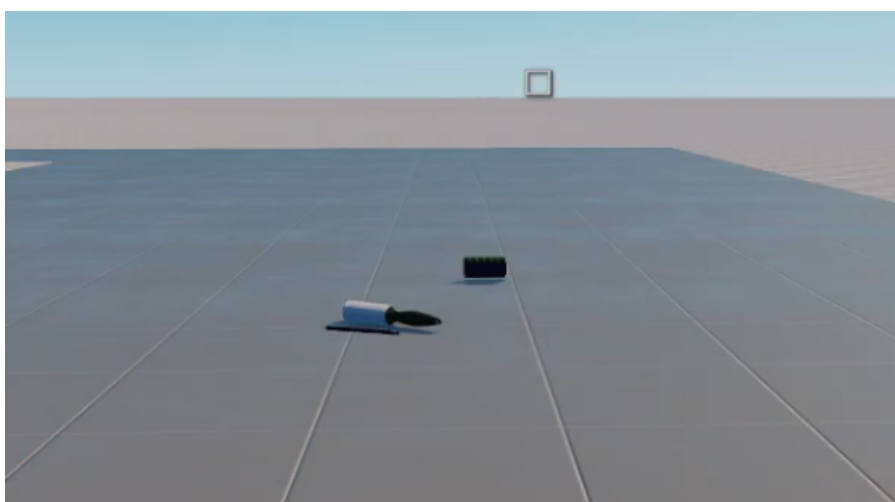


Рис. 22: Карандаш и ролик остановились, губка перекатывается вперед по оси X



## 7 Выводы

В ходе работы были изучены основы использования 3D-движка UNIGINE и его функциональные возможности.

В процессе создания сцены были импортированы три модели, ранее разработанные в Blender 3D: ролик для одежды, губка и карандаш. Каждой из этих моделей были присвоены физические свойства, такие как форма, масса, инерция и другие характеристики, необходимые для взаимодействия объектов в виртуальном пространстве.

Далее были созданы две сцены - сцена с падением предметов в воду и сцена с падением предметов на плоскость. Было продемонстрировано разное поведение предметов в разных физических средах.

В результате проведённой работы была создана интерактивная сцена, демонстрирующая возможности UNIGINE в области рендеринга и физической симуляции, а также улучшены навыки работы с 3D-графикой и моделированием.

## Список литературы

- [1 ] UNIGINE documentation: <https://developer.unigine.com/en/docs/latest/> (дата обращения: 30.04.2024)