

**Автор:**

Матвієнко Богдан Олександрович

Студент 41 ІІЗ групи

**Науковий керівник:**

Селін Юрій Миколайович

Доцент

Кандидат технічних наук

## СИМУЛЯЦІЯ ФІЗИЧНИХ ЯВИЩ ЗА ДОПОМОГОЮ ФІЗИЧНИХ РУШІЇВ

**Анотація.** Метою дослідження є знайомство з принципами роботи фізичних рушіїв двох типів. Завдання дослідження є розбір проектування та розробки фізичного рушія ігрового типу У дослідженні використано методи математичного моделювання

**Ключові слова:** фізичний рушії, наукові рушії, ігрові рушії, зіткнення(collision)

**Постановка задачі.** У сучасному світі все частіше постає питання потреби симуляції природних фізичних явищ. Це потрібно як для досліджень у різних наукових галузях (як приклад симуляція аеродинаміки літаків або високошвидкісних машин), так і для розважальних індустрій (кіно, мультиплікація, комп'ютерні ігри тощо). Питання стоїть у тому як створити ті самі симуляції та що нам для цього потрібно.

**Основна частина.** На сьогоднішній день є два шляхи розв'язання цієї задачі. Перший це створення свого фізичного рушія, а другий це використання готового. Що до першого шляху тут є і плюси і мінуси. Плюсами є гнучкість задач які виконує рушії, адже ми і є розробник тому робимо рушії так як потрібно для нашого конкретного завдання. Мінусом є затрати на розробку, тому як цей процес дуже скрупульозний та трудомісткий, коли в той же час вже готові рушії можуть виконувати потрібні нам задачі. Плюсами другого шляху є вже згадана доступність та економія часу і сил на розробку. Мінусом же є можлива відсутність потрібного функціоналу для поставленої задачі. При виборі другого шляху нам можуть допомогти такі фізичні рушії як «nVidia Phys», «Bullet Physics Library», «Open Dynamics Engine» «Navok».[1]

Усі фізичні рушії діляться на два типи: ігрові й наукові[2]. Ігровий тип використовується в іграх як частина ігрового рушія. Для цього виникає потреба його роботи у реальному часі тобто відтворювати фізичні процеси в грі з тою ж швидкістю, з якою вони відбуваються в реальному світі. Та для ігрового рушія не важлива точність фізичних обчислень, його головна вимога це візуальна реалістичність. В такому випадку не потрібна точна симуляція. Наукові рушії створюються для наукових розрахунків, а саме тут важлива точність обчислень. В такому випадку час грає менше значення так як потрібен точний результат, а не тільки його візуальна реалістичність.

Для того щоб зрозуміти принцип фізичних рушіїв, розглянемо принцип роботи стандартних фізичних явищ у ігровому рушії. Можна думати про фізичний рушії як про безперервний цикл. Починається він з імітації зовнішньої сили, такої як сила тяжіння. На кожному новому кроці часу він виявляє будь-які зіткнення, потім обчислює швидкість і положення об'єкта. І нарешті, відправляє дані про місцезнаходження на графічний процесор. Цей безперервний цикл створює ілюзію, що предмет падає внаслідок сили тяжіння. Фізичний рушії відповідає за обчислення отриманих прискорень, швидкості та переміщення об'єкта від сил і крутних моментів, що діють на тіло. У розробці фізичного рушія найважливішими рівняннями є:

- Другий закон Ньютона
- Сила обертання (моменти, крутний момент)

Ці два рівняння відомі як рівняння руху.

Рушії інтегрує рівняння руху, щоб отримати результуючу швидкість і зміщення об'єкта.

І робить це у безперервному циклі, який складається з наступних кроків:

1. Визначаємо всі сили та моменти, що діють на об'єкт.
2. Беремо векторну суму всіх сил і моментів.

3. Розв'язуємо рівняння руху для лінійного та кутового прискорення.
4. Інтегруємо прискорення щодо часу, щоб знайти лінійну та кутову швидкості.
5. Інтегруємо швидкість щодо часу, щоб знайти лінійне та кутове переміщення.
6. Якщо до об'єкта прикласти гравітаційну силу і крутний момент, цикл рушія створює ілюзію, що об'єкт падає і обертається.

#### Виявлення зіткнень (Collision)

Хоча виявлення зіткнень є відповідальністю фізичного рушія, ця відповідальність покладена на систему виявлення зіткнень. Обидві ці системи працюють рука об руку для визначення зіткнення. Мета системи виявлення зіткнень - повідомити про зіткнення двох об'єктів. Він повідомляє просту відповідь ТАК / НІ. Однак ця операція надзвичайно трудомістка. Для пришвидшення виявлення зіткнень рушії розділяють процес на широко фазний та вузько фазний.

Коли фізичний рушія передає об'єкти в систему зіткнень, об'єкти обгортаються обмежувальними обсягами. Найбільш традиційні обмежувальні об'єми - сфери, Axis-Aligned Bounding Boxes (AABB) та Oriented Bounding Boxes (OBB)[3]. Під час виявлення зіткнень із широкою фазою кожен об'єкт обгортається обмежувальним об'ємом сфери. Рушія аналізує простір кожного об'єкта і створює ієрархію граничного об'єму (Boundary Volume Hierarchy або BVH). BVH створює деревоподібну структуру, де кожен вузол містить об'єкти, які найімовірніше зіткнуться. Фізичний рушія тестує кожен вузол і створює список пар зіткнень. Широка фаза є швидкою, і вона може повідомляти про хибно позитивні зіткнення. Але навіть незважаючи на це, він усуває всі неможливі пари зіткнень. Зіткнення AABB-AABB виявляється швидше, ніж зіткнення OBB-OBB. Однак він повідомляє про помилкові спрацьовування. Потім список пар зіткнень передається до виявлення зіткнення вузької фази. На цій фазі кожен об'єкт обгортається опуклим об'ємом корпусу, що приблизно відповідає його формі. Потім виявлення зіткнення виконує тест на зіткнення між цими опуклими корпусами за допомогою алгоритму Гілберта Джонсона Кірти (GJK)[4]. Цей алгоритм точний, але дорогий.

Реакція зіткнення обчислює результуючі швидкості (лінійні та кутові) об'єктів після зіткнення і миттєво змінює їх швидкості відповідно. Для цього розрахунку потрібна така інформація, як маса, коефіцієнт відновлення, точки зіткнення та вектор нормалі зіткнення.

У момент зіткнення найзначнішою силою, що діє на об'єкти, є сила зіткнення (Сила удару), тому всі інші сили на цей момент ігноруються. Сила удару велика за величиною, але короткочасна. Після зіткнення сила удару згасає, і зовнішні сили знову діють на об'єкт. Рівняння руху вирішується, забезпечуючи нове положення та швидкість. Фізичний рушія виконує цей процес безперервно, і це створює ілюзію, що об'єкт падає через гравітацію.

**Висновки.** Отже коли встає питання створення фізично коректних симуляцій за допомогою комп'ютерної графіки можна створити свій або використати готовий фізичний рушія відповідно до нашої задачі. При правильному підході до постановки задачі і подальшого вивчення можливостей рушіїв, можна очікувати фізично коректний і потрібний нам результат.

#### Список використаних джерел

1. Unity documentation : [Електронний ресурс]. URL: <https://docs.unity3d.com/ru/current/Manual/Physics3DReference.html> (дата звернення: 23.02.2021).
2. Holthe O. PhysX Programming / за ред. Grid Publishing. Trondheim , 2013. 96 с.
3. Darnell K. Visual Simulation: With Student Vissim / Karen Darnell., 1996. – 278 с.
4. Menard M. Game Development with Unity / Michelle Menard., 2011. – 480 с. – (1).