

РЕЗНИКОВ С. А., ЧЕРЕМИСІН А.Ю.

## ДОСЛІДЖЕННЯ ТА РОЗРОБКА МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ІГРОВОГО ДВИЖКА МОВОЮ PYTHON

REZNIKOV S. A., CHEREMISIN A.Yu

### RESEARCH AND DEVELOPMENT OF MATHEMATICAL SOFTWARE FOR GAMING ENGINE LANGUAGE PYTHON.

*Since cont last years game industry in the world has become very popular, and its funding increased considerably, there is a need for new gaming products, gaming ideas and implement them. Thus the idea to create a game engine to accelerate and improve the quality of the creation of game products.*

*And given the popularity of the latest gaming products that create boule, a leading developer gaming products companies and independent developers, we can conclude that the game industry already plays an important role in people's lives. You can also note that the popularity of a game product does not depend only on the developer's company, but on the realization of an idea vdalosti game, as the inner world of the game, entertaining story and gameplay successful implementation.*

*Quite difficult is to implement the right algorithm for the solution of a problem with the software as necessary to foresee possible errors in the program, as well as errors in the input that the program will receive from the user.*

*It is important to create a game engine to provide intuitive user class names playing.*

#### **Вступ**

Кожен програміст або фірма яка займається створенням ігрових продуктів хочуть, щоб їхні ігрові продукти були цікавим, якісними там мали достатньо широкий функціонал. А для цього вони повинні або самі розробляти ігровий движок для своїх продуктів або використовувати вже існуючі. Кожен із варіантів має свої плюси та мінуси. Наприклад створення свого ігрового движка є достатньо трудомісткою працею и не завжди виходить таким вдалим, як спочатку задумував розробник. Часто ігровий движок особистої розробки новачка програміста або фірми має ряд недоліків такі як малий функціонал, невдала реалізація тієї чи іншої функції, що в свою чергу тягне за собою низьку оптимізацію, але найчастіше, якщо програміст початківець вирішив написати свою гру на своєму ігровому, може з'явитися недолік, коли певний ігровий движок може розкрити свій потенціал лише в запланованій грі.

З іншої сторони, при створенні власного ігрового движка можна отримати декілька переваг, такі як: повне розуміння процесів, що проходять у грі, можливість змінювати по своєму розумінні деякі блоки ігрового движка, а також повна відповідність ігрового движка запланованій грі.

#### **Постановка завдання**

Дослідити та розробити математичне забезпечення для ігрового движка мовою Python. На основі розробленого ігрового движка дослідити поняття ігрового движка, різницю між поняттями ігровий движок та ігровий продукт а також розробити математичне забезпечення для фізичної підсистеми ігрового движка.

### Основна частина

Існують розбіжності з приводу того, чим саме є ігровий движок, іноді з фундаментальними відмінностями у визначеннях. Сімпсон повідомляє про плутанину між ігровими движками і самими іграми, а також про помилковий опис ігрових движків як ігрового компонента для відображення графіки ігор.

Визначення ігрового движка, які не обмежені окремими компонентами движків, часто дуже широкі і розпливчасті. Наприклад, «Фреймворк складається з набору різних інструментів, утиліт і інтерфейсів, які приховують деталі низького рівня різних завдань, які складають відеогру» (Sherrod 2007). Виникає головна проблема в питанні, де лежить межа між ігровим движком і самою грою.

Є ознаки того, що загальний консенсус може бути злиттям у визначення ігрового движка. Більш конкретні описи були запропоновані, наприклад від Льюїса і Якобсона, що ігрові движки це «набір модулів симуляції коду, які безпосередньо не задають поведінку гри (логіку гри) або оточення гри (дані про рівень)». Хоча це, безумовно, наближає нас до розуміння межі ігрового движка і ігровий логіки, цей розрив все ще чітко не визначений і залишається безліч перекриттів.

Можна сказати, що ігровий движок - це програмний компонент, що дозволяє нам створювати і запускати відеоігри. Він надає розробникам інструменти для створення більшості компонентів гри, а потім дозволяє їм зібрати їх воедино. Движок гри зачіпає всі компоненти гри, починаючи від рендеринга, фізики, звукового оформлення, скриптинга, створення штучного інтелекту і закінчуючи мережевими аспектами. А якщо, щось створити з його допомогою і не можливо, можна створити це в спеціалізованій програмі та потім імпортувати це в гру. У кожному разі, движки гри це робочі конячки сучасних відеоігор.

Також можна сказати, що ігровим движком є набір систем, які спрощують найбільш часто використовувані функції гри. Движок складається з підсистем, контролюючих певні частини гри. Більшість ігор мають наступні підсистеми(архітектуру):

- Графічна підсистема (візуалізатор);
- Підсистема вводу;
- Фізична підсистема;
- Звукова підсистема;
- Системне ядро.

Деякі ігри можуть мати більше підсистем в залежності від їх потреб. Наприклад, додаткова підсистема може працювати з мережею. В основному все, що може бути згруповано в категорії, може стати підсистемами.

В якості першого кроку до дослідження ігрових движків та їх компонентів, дослідження в цій галузі повинні будуть вивчити

комп'ютерні ігри та ігрові движки, щоб визначити програмні компоненти, які є загальними і унікальними для різних типів ігор. Це повинно привести до розуміння чіткої відмінності між кодом ігрового движка і ігрового коду, допомагаючи встановити цю межу.

Існує велика різноманітність архітектури ігрових движків, по своїй суті кожен зі створених ігрових движків має свою особливу архітектуру.

Для кращого розуміння функцій розробленого ігрового движка потрібно детальніше розглянути його складові.

Ядро системи виконує низько рівневі операції по створенню вікна, управлінню ігровими станами і процесами.

Візуалізатор в ігровому движку виконує одну з самих головних ролей, а саме відображає на вікні графічні елементи ігрового движка, такі як модель гравця, модель противників, карту рівня та графічні ефекти.

Формат зберігання карти рівня і моделей є функцією візуалізатора, а не частиною додатка/гри. Логіці гри не потрібно знати, як об'єкти представляються у пам'яті, або як візуалізатор буде їх відображати. Гра просто повинна бути в курсі, що саме відображає візуалізатор, чи в правильному ракурсі він відображає, і ті чи він показує моделі, що потрібно.

Фізичний движок в ігровому движку є дуже важливою частиною. Він дозволяє зробити за нас дві важливі речі:

- виявляти зіткнення (колізії) між нашими ігровими об'єктами;
- імітувати сили та рухи в результаті зіткнень.

Розглянемо виявлення зіткнень. Ігри були б не дуже цікавими, якщо ваш персонаж провалився б крізь підлогу, не зробивши жодного кроку або стрибка. Функції фізичної частини ігрового движка, що виявляють колізії дозволяють позбавити ігровий продукт від таких недоліків та значно розширюють його функціонал. Виявлення зіткнень дозволяє дуже точно визначати взаємодію двох тіл.

Розглянемо імітацію сили. Після зіткнення можуть відбутися різні явища: може стрибнути персонаж, може стрибнути якийсь інший ігровий об'єкт, а може ви просто будете рухатися в якомусь напрямку. Все це виконується движком за кадром. Але сили, які може імітувати фізичний движок, не обмежені зіткненнями. Існують такі поняття, як гравітація і імпульс. Вони можуть застосовуватися до тіл, навіть якщо ті не стикалися. Також такі сили можуть впливати на дії, які відбуваються в грі, на рухи героїв і навіть на сам світ.

Для найкращої симуляції руху об'єктів в фізичному модулі ігрового движка (падіння з висоти, переміщення об'єктів під дією сил гравітації та імпульсу) мною були використані формули руху тіла під кутом до горизонту.

У такому випадку рух тіла можна розглядати як суму двох рухів: в горизонтальному напрямі – рівномірного і у вертикальному напрямі – рівноприскореного.

Якщо спроектувати вектор  $V_0$  на кожен з координатних осей  $x$  та  $y$  [1].

$$V_{0x} = V_0 * \cos \alpha; \quad V_{0y} = V_0 * \sin \alpha; \quad (1)$$

Тоді координати точки для будь-якого моменту часу будуть [2] (без урахування вітру та супротиву повітря):

$$x = V_0 * \cos \alpha * t; \quad y = -g * \frac{t^2}{2} + V_0 * \sin \alpha * t; \quad (2)$$

Якщо з рівнянь [2] виключимо час, то дістанемо рівняння траєкторії [3]:

$$t = \frac{x}{V_0 * \cos \alpha}; \quad y = \tan \alpha * x - g * \frac{x^2}{2 * V_0^2 * \cos^2 \alpha}; \quad (3)$$

Отже об'єкт буде летіти по параболічній траєкторії. За допомогою вище наведених формул симуляція фізичного руху об'єктів в фізичній частині ігрового движка буде максимально наближеною до реальних показників.

Звукова підсистема займає велику роль в ігрових продуктах так як надає можливість розробникам створювати в ігрових продуктах потрібну їм ігрову атмосферу, реалізувати озвучку ігрових персонажів, дій гравця та інші звуки.

### Висновки

Теоретичні дослідження руху тіла під кутом до горизонту дозволяють створити реалістичну симуляцію руху об'єкта в координатній системі під дією сил гравітації, інерції та імпульсу. Дослідивши розроблений ігровий движок можна зробити висновок, що використані для симуляції руху тіла формули дають достатньо правдоподібні результати, а для поліпшення якості симуляції можна додати декілька додаткових функцій, які будуть симулювати вплив вітру та супротиву повітря при русі тіла у просторі.

### Література

1. Рух тіла, кинутого під кутом до горизонту. Рух тіла, кинутого горизонтально [Електронний ресурс] – Режим доступу до ресурсу: <http://posibnyky.vntu.edu.ua/fizika/1.1.13.htm>

2. Игровые движки [Електронний ресурс] – Режим доступу до ресурсу: <http://developerpro.ru/>