Міністерство освіти і науки України

Департамент освіти і науки Дніпропетровської облдержадміністрації

Дніпропетровське територіальне відділення МАН України

Відділення: комп’ютерні науки

Секція: мультимедійні, графічні та навчальні програми

«Моделювання пружного зіткнення на площині двох куль із різними радіусами»

Роботу виконав:

Каманцев Артем Сергійович

учень 11 класу Дніпровського ліцею інформаційних технологій

при Дніпропетровському національному університеті

імені Олеся Гончара

Науковий керівник:

Ентін Й. А. ,

вчитель-методист, к. ф.-м. н.

Дніпро – 2018

**ТЕЗИ**

**Каманцев Артем Сергійович**

учень 11 класу Дніпровського ліцею інформаційних технологій

при Дніпропетровському національному університеті імені Олеся Гончара

Науковий керівник: Ентін Й. А. , вчитель-методист, к. ф.-м. н.

**«Моделювання пружного зіткнення на площині двох куль із різними радіусами»**

Актуальність роботи - у моделюванні задачі абсолютно пружного та жорсткого не центрального зіткнення куль на площині, коли площина, що містить точку зіткнення й центри куль перпендикулярна площині, по якій рухаються кулі.

Мета роботи - створення програмного проекту, що дозволяє моделювати абсолютно пружне й жорстке зіткнення куль різних радіусів на площині й допомагає краще зрозуміти закони фізики, що є основою розрахунків при моделюванні.

Програма складається з п’яти основних розділів, головними з яких є «Меню моделювання» та «Моделювання», які надають користувачу можливість моделювати і порівнювати варіанти руху куль після зіткнення в залежності від вхідних параметрів. Розділ «Тести» дозволяє користувачу вдосконалити навички вирішення кількох типів задач, повторюючи їх розв’язування стільки разів, скільки потрібно завдяки випадковій генерації параметрів цих завдань. В останній частині тестів користувачу пропонується вирішити задачу абсолютно пружного та жорсткого зіткнення куль із різними радіусами на горизонтальній площині.

Розроблено алгоритм розрахунку абсолютно пружного та жорсткого зіткнення куль із різними радіусами за умови розташування точки зіткнення куль у площині, що містить центри куль і перпендикулярна до поверхні Землі. Створено проект моделювання зіткнення куль, який дозволяє користувачу краще зрозуміти явища абсолютно пружного і жорсткого удару та закони збереження імпульсу і енергії.

Зміст

[УМОВНІ ПОЗНАЧЕННЯ 5](#_Toc502613643)

[ВСТУП 7](#_Toc502613644)

[ОСНОВНА ЧАСТИНА 9](#_Toc502613645)

[1. Теоретичний матеріал 9](#_Toc502613646)

[1.1. Рух кулі, кинутої під кутом до горизонту 9](#_Toc502613647)

[1.2. Застосування розрахунків у програмі 10](#_Toc502613648)

[1.3. Зіткнення 12](#_Toc502613649)

[1.3.1. Зіткнення з горизонтальною площиною 12](#_Toc502613650)

[1.3.2. Зіткнення з іншою кулею 12](#_Toc502613651)

[1.3.2.1. Розкладання швидкостей 13](#_Toc502613652)

[1.3.2.2. Куля ліворуч від дотичної площини 14](#_Toc502613653)

[1.3.2.3. Куля праворуч від дотичної площини 15](#_Toc502613654)

[1.3.2.4. Центральне зіткнення куль 17](#_Toc502613655)

[1.3.2.5. Додавання швидкостей 18](#_Toc502613656)

[1.4. Умова перельоту однієї кулі через іншу 19](#_Toc502613657)

[2. Програмний проект 21](#_Toc502613658)

[1.5. Розділи 21](#_Toc502613659)

[1.5.1. Загальні відомості про роботу 21](#_Toc502613660)

[1.5.2. Меню 21](#_Toc502613661)

[1.5.3. Пояснювальна інформація 22](#_Toc502613662)

[1.5.4. Карта додатку 22](#_Toc502613663)

[1.5.5. Теорія 23](#_Toc502613664)

[1.5.6. Тести 23](#_Toc502613665)

[1.5.7. Меню моделювання 24](#_Toc502613666)

[1.5.8. Створення/Редагування елементу 25](#_Toc502613667)

[1.5.9. Налаштування 27](#_Toc502613668)

[1.5.10. Моделювання 29](#_Toc502613669)

[1.6. Структура файлів 30](#_Toc502613670)

[1.6.1. Файл налаштувань 30](#_Toc502613671)

[1.6.2. Файл даних 31](#_Toc502613672)

[1.7. Параметри за умовчанням 32](#_Toc502613673)

[1.7.1. Налаштування 32](#_Toc502613674)

[1.7.2. Дані 33](#_Toc502613675)

[1.8. Програмні та апаратні вимоги 34](#_Toc502613676)

[1.9. Структурна схема програми 35](#_Toc502613677)

[3. Використані програми 35](#_Toc502613678)

[4. Комплектація програми 36](#_Toc502613679)

[ВИСНОВКИ 37](#_Toc502613680)

[ДЖЕРЕЛА 38](#_Toc502613681)

[ДОДАТОК А 39](#_Toc502613682)

# УМОВНІ ПОЗНАЧЕННЯ

Далі використано наступні умовні позначення.

, м – радіус більшої кулі.

, м – радіус меншої кулі.

, м/с **–** проекції швидкостей на осі X та Y відповідно будь-якої з куль.

, м/с **–** швидкість першої кулі до зіткнення.

, м/с – проекція швидкості більшої кулі на вісь **X.**

, м/с- проекція швидкості більшої кулі на вісь **Y.**

, м/с **–** швидкість другої кулі до зіткнення.

, м/с – проекція швидкості меншої кулі на вісь **X.**

, м/с – проекція швидкості меншої кулі на вісь **Y.**

, м/с **–** швидкість першої кулі після зіткнення.

, м/с **–** швидкість другої кулі після зіткнення.

, с **-** інтервал часу, через який куля впаде на землю.

, с **–** час падіння кулі на землю.

**,** м **–** попередні координати центра одної з куль.

**,,** м – координати центра більшої кулі.

**,,** м **–** координати центра меншої кулі.

**B –** кут нахилу швидкості кулі.

**g**, м/– прискорення, що надає сила тяжіння. У наведених розрахунках дорівнює прискоренню вільного падіння (≈-9.80665 м/). Розрахунки залишаються справедливими для будь-якого від’ємного значення прискорення.

**M**, кг **–** маса другої кулі.

**m**, кг **–** маса першої кулі.

**R****,** м– радіус будь-якої з кулі.

**t**, с **–** інтервал часу, за який розраховується переміщення кулі.

**V**, м/с– модуль швидкості кулі.

**V1**, м/с **–** складова швидкості кулі, направлена вздовж прямої, що з’єднує центри куль.

**V2**, м/с **–** складова швидкості кулі, направлена вздовж дотичної прямої.

**V3**, м/с – швидкість, отримана внаслідок додавання векторів **V1** та **V2**.

**x, y,** м - координати центра одної з куль.

**–** кут, прямої, що сполучає центри куль при зіткненні.

– будь-який кут. Використовується у формулах, що відображують загальні правила тригонометрії.

**–** кут між векторами **V1** та **V2**.

**–** кут між векторами **V2** і **V3**.

Кутом нахилу вектору називається кут, утворений вектором з віссю абсцис

Усі кути нахилу векторів розраховуються відносно додатного напрямку осі абсцис проти годинникової стрілки і належать проміжку , вимірюються у радіанах.

# ВСТУП

**Актуальність**

У шкільному курсі фізики школі вивчають ідеально пружне центральне зіткнення куль однакових радіусів на горизонтальній площині. Мабуть, тільки при підготовці до олімпіад вивчають нецентральне пружне зіткнення. Такий тип задачі більш наближений до реальності: важко забезпечити умови для здійснення центрального зіткнення куль. Моделювання даної задачі допоможе учням краще зрозуміти специфіку законів збереження імпульсу та енергії. Тому актуальне створення проекту моделювання пружного зіткнення двох куль із різними радіусами на горизонтальній площині.

**Мета роботи**

Створити програмний продукт, що моделює ідеально пружне та жорстке (без супроводжуючої деформації) зіткнення на площині двох куль із різними радіусами за умови перпендикулярності горизонтальної площини, по якій рухаються кулі, та площини, що містить точку зіткнення куль і їх центри, і надає можливість порівняти вплив на особливості руху початкових швидкостей, радіусів та мас куль.

**Задачі роботи**

Задачею роботи є створення розрахунків, що описують повністю пружне та жорстке зіткнення на горизонтальній площині двох куль довільних радіусів і програмне моделювання цього зіткнення а також порівняння результатів моделювання при таких вхідних даних.

1. Сильно відрізняються значення радіусів та мас куль.
2. Маси куль відрізняються слабко, різниця радіусів велика.
3. Радіуси куль приблизно однакові, маси сильно відрізняються.

**Характеристика роботи**

У роботі використовуються такі фізичні поняття, як імпульс, кінетична енергія, рівномірний прямолінійний та рівноприскорений рух, відносний рух а також закони збереження імпульсу та енергії. При побудові математичної моделі і виконанні розрахунків використано властивості кулі й дотичної прямої, проведеної до неї, властивості квадратичного рівняння, методи вирішення квадратичних рівнянь і систем рівнянь з двома змінними, основні властивості тригонометричних функцій.

Ці поняття, закони, властивості враховано в алгоритмі моделювання абсолютно пружного та жорсткого зіткнення куль на площині.

Програмний проект виконаний мовою програмування C# із застосуванням технології WPF, що внесло деякі корективи у характер розрахунків та їх опису у роботі. Це також надало можливість створити зручний інтерфейс користувача, організувати динамічне створення тестів і наочну візуалізацію зіткнення куль, забезпечити взаємодію програми з файловою системою комп’ютера і роботу з базою даних для збереження змін, внесених користувачем.

# ОСНОВНА ЧАСТИНА

## Теоретичний матеріал

Створення рисунків було виконано за допомогою програмних продуктів GeoGebra, для створенні креслень, та paint.net, для їх подальшого редагування.

### Рух кулі, кинутої під кутом до горизонту

Якщо тіло рухається під кутом до горизонталі, його швидкість можна спроектувати на осі **X** та **Y**:

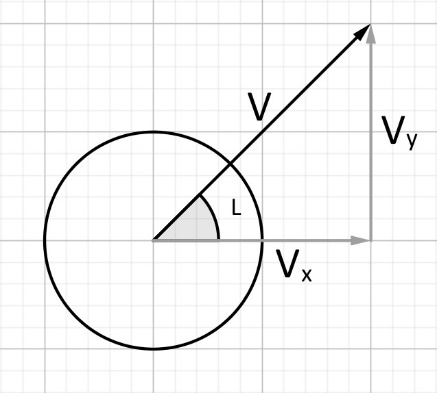


Рис. 1.1 Розкладання вектору швидкості

Рух проекції тіла на вісь **X** рівномірний та прямолінійний, а проекції на вісь **Y** – рівноприскорений. Тому координати тіла у просторі в будь-який момент часу розраховуємо за формулами:

### Застосування розрахунків у програмі

Один метр вважається рівним ста пікселям на екрані.

Кулі у програмі представлені об’єктами, що зберігають усі дані про їх стан та визначають позиціонування кулі у момент часу **t**.

Переміщення вздовж осі **X** розраховується за відповідною формулою рівномірного прямолінійного руху.

Перед розрахунком переміщення вздовж осі **Y** та зміною цієї складової швидкості необхідно перевірити, чи перебуває тіло у стані польоту, бо інакше переміщення вздовж **Y** не має сенсу.

Якщо вибрати координату поверхні землі за нуль, а радіус кулі позначити **R**, можна знайти час, через який відбудеться її зіткнення з землею:

Оскільки і **g** завжди спрямоване униз, то буде завжди більше або рівний за модулем, тому шуканий час завжди буде визначатись за формулою: , оскільки буде мати від’ємне значення.

Якщо **t <**, куля не впаде за поточний інтервал часу **t**. Отже, можна застосувати формули для знаходження швидкості та координати у наступний момент часу.

Інакше застосовуємо наступні розрахунки.

1. Знаходимо проекцію швидкості на **Y**, яку набуде тіло, доки впаде на землю (падіння відбудеться за час ):. Беремо її за модулем, оскільки важливим є модуль швидкості, а не напрям її вектору.
2. Віднімаємо від часу, за який необхідно змоделювати переміщення, час, протягом якого тіло знижувалося:
3. Знаходимо інтервал часу (**)** через який відбудеться повторне зіткнення кулі з землею, якщо на неї не будуть діяти зовнішні сили окрім сили тяжіння.

Оскільки , а в даному випадку , то

1. Якщо знайдений час менший від мінімального врахованого часу, залишаємо незмінною вже розраховану швидкість і враховуємо дрібні стрибки, що повинна виконати куля до кінця заданого часу.
2. Інакше знаходимо час, проведений кулею в польоті після численних стрибків. Нехай куля за час **t** виконує **N** цілих стрибків. **N** визначимо за формулою: . Тоді час, проведений кулею в польоті після стрибків дорівнює **.** Розраховуємо швидкість та координату кулі через цей проміжок часу, вважаючи, що початкова координата (:

Для завершення розрахунків лишається знайти нову швидкість кулі за модулем та визначити кут нахилу її вектору до осі абсцис. Модуль обчислимо за формулою: , а кут нахилу визначаємо за допомогою функції класу Math – Atan2: .

### Зіткнення

Вважається, що кулі рухаються у стані вільного падіння чи ковзання по горизонтальній поверхні, доки не відбудеться зіткнення з горизонтальною площиною або іншою кулею.

#### Зіткнення з горизонтальною площиною

Оскільки зіткнення з горизонтальною площиною вважається ідеально пружним, кут падіння дорівнює куту відбиття. Враховуючи, що кут швидкості кулі - **B**, який відраховується від додатного напрямку осі абсцис проти годинникової стрілки, у момент зіткнення з землею належить проміжку **(-π;0)**, для його віддзеркалення достатньо змінити знак кута вектору відносно додатного напрямку осі **X**:

#### Зіткнення з іншою кулею

Момент зіткнення двох куль можна розглядати як їх зіткнення з дотичною площиною (Рис. 1.2), яка перпендикулярна прямій, що з’єднує центри куль (за властивістю дотичної площини до кулі).

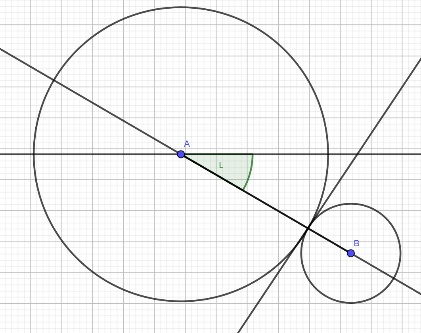


Рис. 1.2 Зіткнення куль

Відзначимо, що вектори швидкостей куль належать площині, що містить центри куль і є перпендикулярною горизонтальній площині, якою рухаються кулі. Для розрахунку нових значень швидкостей куль необхідно розглянути два випадки: куля знаходиться ліворуч або праворуч від дотичної площини. Розглянемо розкладання вектору швидкості кожної з куль згідно з цими випадками і визначимо для кожної кулі усі складові швидкості, набутої внаслідок удару

Далі розглядаємо випадки зіткнень куль при від’ємному значенні кута (перша куля більша від другої). Розрахунки коректні при будь-яких співвідношеннях радіусів.

##### Розкладання швидкостей

Для подальших розрахунків слід визначити проекції швидкостей кожної з куль на дотичну площину та пряму, що сполучає центри куль і є перпендикулярною до дотичної площини.

Проекції на пряму, що з’єднує центри куль, чий кут нахилу належатиме проміжку **[-π/2;π/2]** вважатимемо за модулем додатними, інші ж – від’ємними. Проекції на дотичну вважатимемо завжди додатними за модулем, оскільки після їх розрахунку і до моменту визначення нової швидкості кулі після зіткнення саме ці проекції визначатимуть модуль швидкості кулі, а їх напрям буде визначатись кутом нахилу швидкості – **B**.

##### Куля ліворуч від дотичної площини

У момент зіткнення вектор **V** розкладається на складові **V1** та **V2,** як показано на наступних рисунках.

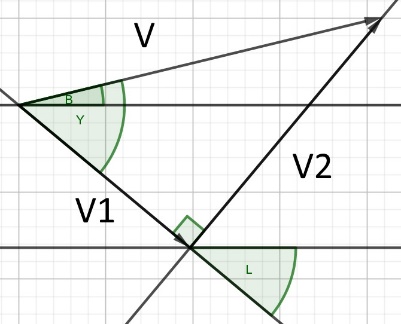


Рис. 1.3 Швидкість напрямлена до дотичної площини і B>L

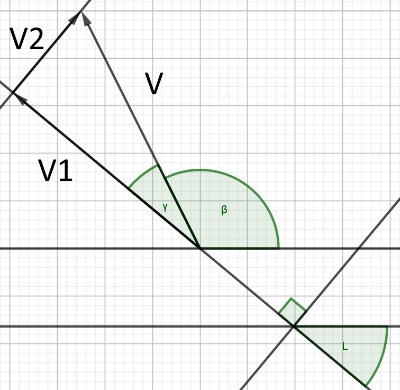


Рис. 1.4 Швидкість напрямлена від дотичної площини

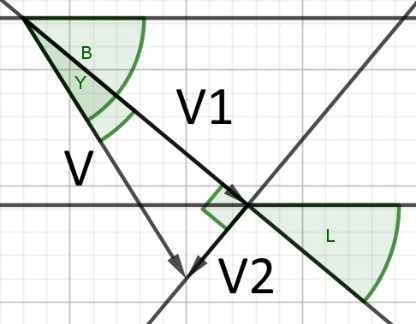


Рис. 1.5 Швидкість напрямлена до дотичної площини і B<L

Для кута **Y** маємо:

для Рис. 1.3 і Рис. 1.5

**–** для Рис. 1.4

У випадку, зображеному на Рис. 1.3, значення не від’ємне: **L<0**, **B>0**. Модулі векторів **V1** і **V2** додатні, їх можна знайти за формулами:

**.**

Враховуючи, що у випадку, зображеному на Рис. 1.5, від’ємне, оскільки **L<0**, **B<0,** **|B|>|L|**, формули для модулів векторів **1** і **2** (додатних) дещо змінюються:

У випадку, зображеному на Рис. 1.4, модуль вектору **1** від’ємний, а модуль вектору **2** додатний. Тому розрахунки лишаються вірними:

Обчислимо кут нахилу вектору **2**. Якщо кут нахилу вектору більший від кута **L**, то кут вектору **2** дорівнюватиме **L+π/2**, інакше його значення дорівнюватиме **L-π/2**.

Вектор швидкості **1** буде використано при подальших розрахунках, вектор **2** на даному етапі розрахунків визначає швидкість поточної кулі.

##### Куля праворуч від дотичної площини

В момент зіткнення вектор має складові: **1** та **2,** як показано на наступних рисунках.

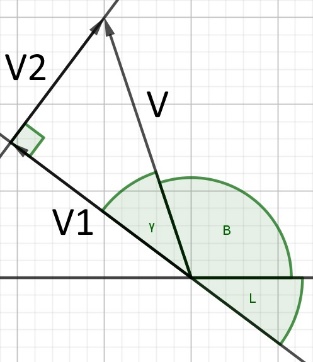


Рис. 1.7 Швидкість направлена до дотичної площини і π>|B|+|L|

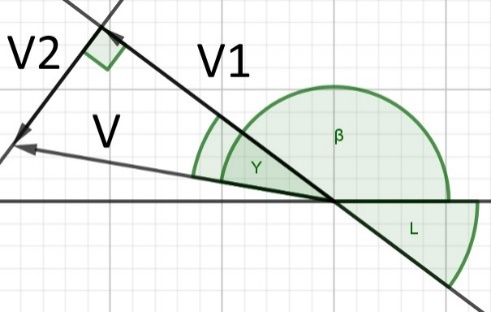


Рис. 1.6 Швидкість направлена до дотичної площини і π<|B|+|L|

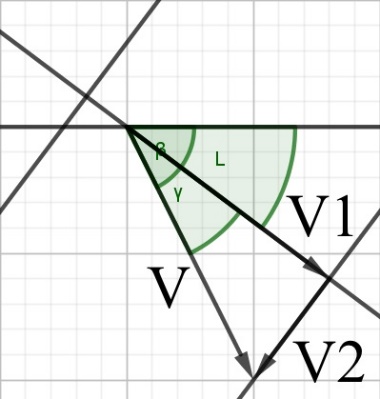


Рис. 1.8 Швидкість направлена від дотичної площини

Кут **Y** дорівнює:

**–** для Рис. 1.6 і Рис. 1.7

**–** для Рис. 1.8

У випадку, зображеному на Рис. 1.8, належить проміжку **.** Оскільки модулі **1** і **2** додатні, вони обчислюються за формулами:

У випадку, зображеному на Рис. 1.7, значення належать проміжку **[0; π/2).** Оскільки **1** має бути від’ємним, а модуль **2** додатнім, залишаються вірними розрахунки:

**–** оскільки **cos(π-)=-cos()**

**–** оскільки **sin(π-)=sin()**

У випадку, зображеному на Рис. 1.6, **.** Оскільки модуль вектору **1** має бути від’ємним, а модуль **2** додатнім, формули залишаються вірними:

Обчислимо кут нахилу вектору **2**. Якщо виконується умова:

,

кут вектору **2** дорівнює **L+π/2**, інакше його значення дорівнює **L-π/2**.

Вектор швидкості **1** буде використано при подальших розрахунках, вектор **2** на даному етапі розрахунків визначає швидкість поточної кулі.

Після виконання розрахунків, описаних у розділах «Куля ліворуч від дотичної площини» і «Куля праворуч від дотичної площини», для кожної з куль складові їх швидкостей **V2** належать дотичній площині, залишаючись також у площині, що містить центри куль і перпендикулярна горизонтальній площині, якою рухаються кулі, і не впливають на взаємодію куль; складові **V1** напрямлені вздовж прямої, що з’єднує центри куль. Отже, можемо знайти швидкості куль після зіткнення, розглянувши зіткнення вздовж прямої, що з’єднує їх центри.

##### Центральне зіткнення куль

Знайдемо швидкість руху першої кулі відносно другої вздовж прямої, що з’єднує їх центри, і позначимо її як **V**:

**.**

З закону збереження імпульсу отримуємо рівняння:

***,***

де і  **–** швидкості куль після зіткнення.

З закону збереження енергії отримуємо:

Помножимо його на два й напишемо систему рівнянь:

Розділивши друге рівняння на перше, маємо:

**.**

Тепер знайдемо швидкості першої і другої куль після зіткнення:

Оскільки дані розрахунки вірні для системи відліку, пов’язаної з другою кулею, остаточні формули для знаходження модулів швидкостей куль вздовж прямої, що з’єднує їх центри, у нерухомій системі відліку будуть такі:

**,**

та - значення, яких набудуть модулі векторів тавідповідно, тобто складові **V1** швидкостей кожної кулі.

##### Додавання швидкостей

Після виконаних розрахунків, кожна куля має дві складові швидкості: **V1**, яка належить прямій, що сполучає центри куль, і **V2**, що належить дотичній площині і площині, що містить центри куль і є перпендикулярною до горизонтальної площини, якою рухаються кулі.

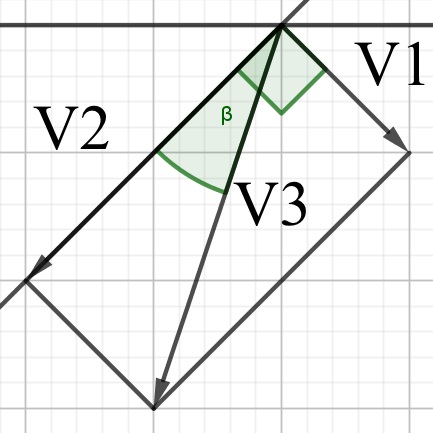


Рис. 1.9B<0 |V1|>0

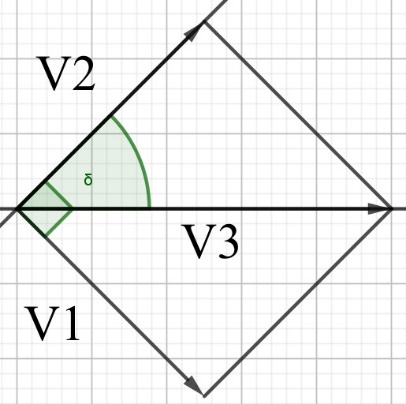


Рис. 1.10 B>0 |V1|>0

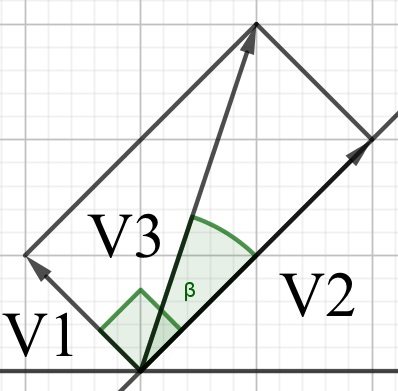


Рис. 1.12 B>0 |V1|<0

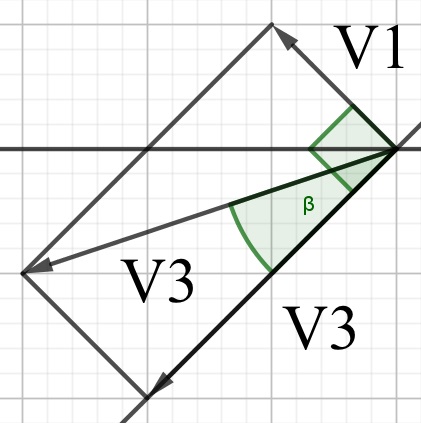


Рис. 1.11 B<0 |V1|<0

Очевидно, що оскільки ці вектори перпендикулярні (за властивістю прямої, перпендикулярної до площини), модуль сумарного вектору буде обчислюватися за формулою:

**.**

Кут знайдемо за формулою:

**.**

Оскільки ми вважали проекції, чий кут нахилу належить проміжку **[-π/2;π/2]** за модулем додатними, а інші – від’ємними, то, проаналізувавши усі можливі взаємні положення векторів **V1** та **V2** (Рис. 1.9, Рис. 1.10, Рис. 1.11, Рис. 1.12)**,** можна стверджувати наступне.

Якщо кут нахилу вектору **V2, B**,та модуль вектору **V3** однакові за знаком, то кут нахилу вектору **V3** дорівнюватиме  **,** інакше ж, якщо та **V3** різні за знаками, кут на нахилу вектору **V3** дорівнюватиме  **.**

Тепер швидкість кулі визначається вектором **V3.**

### Умова перельоту однієї кулі через іншу

Тільки більша куля може перескочити через меншу, що обумовлено їх формою. Щоб визначити умови перельоту більшої кулі через меншу, розглянемо складові швидкостей по осях **X** та **Y** обох куль після зіткнення, координати центрів більшої та меншої куль а також їх радіуси.

Враховуючи, що рух проекцій куль по осі **X** є рівномірним, то першою умовою є виконання нерівності: , бо інакше більша куля не зможе перегнати меншу.

Розрахуємо час **t**, через який центр більшої кулі переміститься за центр меншої.

Обчислимо нові координати центрів куль відносно осей **X** та **Y** згідно з формулами рівномірного прямолінійного та рівноприскореного рухів.

Наступною умовою перельоту більшої кулі через меншу є знаходження більшої через час **t** на висоті, не меншій від , над площиною, на якій відбувається зіткнення куль. Для цього має виконуватися нерівність:

Наведені умови є необхідними для перельоту більшої кулі через меншу, проте не є достатніми, оскільки не враховують можливі зіткнення куль впродовж часу t: вони важко передбачувані.

## Програмний проект

### Розділи

Проект складається з вікна, що містить загальні відомості про роботу, вікон головного меню, п’яти основних розділів: «Теорія», «Карта доданку», «Меню моделювання», «Тести», «Моделювання» та двох допоміжних розділів: «Створення/Редагування елементу» і «Налаштування» а також вікна з пояснювальною інформацією, відкриття якого можливе з деяких розділів програми.

Розглянемо детальніше кожну частину програми.

#### Загальні відомості про роботу

При відкритті програми користувач бачить вікно з загальними відомостями про роботу. В цей час програма виконує зчитування налаштувань з файлу Settings.dat та даних із файлу, який є базою даних XMLі шлях до якого є одним із параметрів файлу налаштувань. При помилці на будь-якому етапі зчитування налаштування приймають значення за умовчанням та статус не збережених.

При наявності помилки у процесі завантаження даних будуть відображені елементи, які вдалось зчитати до виникнення помилки. Якщо не вдається знайти файл за шляхом, вказаним у файлі налаштувань, програма намагатиметься зчитати дані з файлу за умовчанням. Тоді налаштування мають статус не збережених. Дані є збереженими навіть у разі невдалого завантаження.

#### Меню

Звідси користувачу доступні переходи до розділів «Карта додатку», «Меню моделювання», «Теорія» та «Тести» чи повернення до вікна із загальними відомостями про роботу.

#### Пояснювальна інформація

Вікно, що містить пояснювальну інформацію, не є розділом програми. Текст, який воно містить, залежить від розділу програми, з якого його викликано. Вікно відкривається натисканням клавіші «F1», про що користувача сповіщають відповідні написи в нижньому лівому куті розділів, котрі мають пояснення. Закриття відбувається за натисканням клавіші «Esc».

#### Карта додатку

Це вікно візуалізує усі наявні розділи в програмі, а також зв’язок між ними.

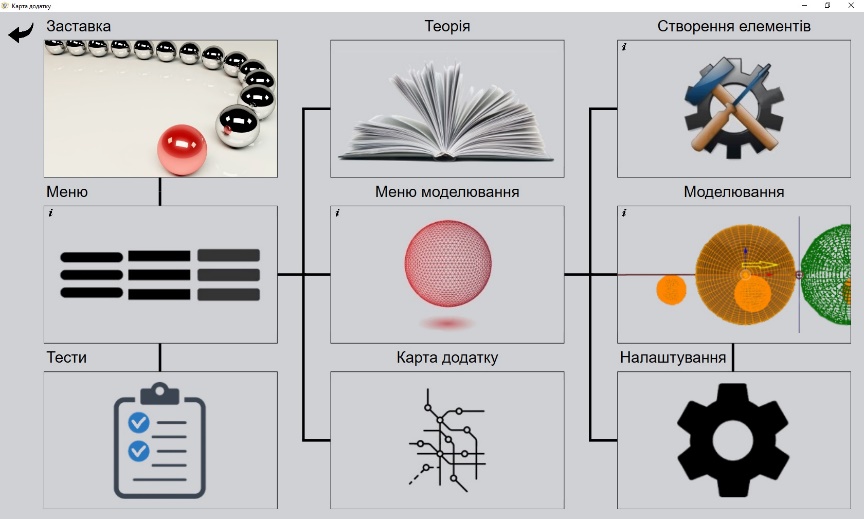


Рис. 2.1 Карта додатку

Розділи, з яких доступне відкриття пояснювальної інформації, мають у лівому верхньому куті своєї позначки на схемі картинку .

Звідси можна перейти до розділів «Теорія», «Меню моделювання» чи «Тести», повернутись до «Головного меню» чи розділу «Заставка» (останні два переходи еквівалентні переходам за допомогою знаків  ), відкрити вікна розділів «Створення елементів» та «Налаштувань». Перехід до розділу «Моделювання» недоступний, бо такий перехід потребує попереднього вибору ситуацій для моделювання у розділі «Меню моделювання». Про неможливість переходу до розділу користувача сповіщає ToolTip на відповідному елементі схеми. Спливаючі підказки містять коротку характеристику відповідного розділу.

Якщо користувач переходить до якого-небудь розділу, то, натиснувши кнопку повернення, він перейде до розділу «Карта додатку», а не до розділу - логічного попередника. Наприклад, при переході до «Теорії» й натисканні кнопки повернення користувач перейде знову до розділу «Карта додатку», а не до «Головного меню».

#### Теорія

У цьому розділі можна переглянути теоретичний матеріал, на основі якого ведуться розрахунки у програмі. Перелік доступних файлів можна змінити, додавши чи видаливши файли з папки «Theory», що знаходиться у папці «Sources», яка повинна знаходитись у папці з .exe файлом програми. Додавати можна файли, що мають розширення .htm або .html.

#### Тести

Цей розділ складається з трьох частин різного рівня складності, які дозволяють користувачу перевірити свої знання. У кожній частині користувач може перевірити власні відповіді, переглянути правильні відповіді чи оновити умови, що дозволить спробувати виконати завдання знов.

Перша частина тестів містить чотири запитання за такими темами як пружне зіткнення, імпульс та закон збереження імпульсу. Варіанти відповідей не змінюються, проте їх порядок різний при кожному новому відкритті вікна «Тести».

Друга частина складається з двох задач середнього рівня складності.



Рис. 2.2 Тест. Друга частина.

Починаючи з цього етапу варіанти відповідей та параметри задач генеруються випадково. Крім того, користувач може власноруч вводити значення вхідних параметрів задач. Користувачу пропонується знехтувати дробовою частиною відповіді, оскільки точності розрахунків користувача і програми можуть відрізнятись причина – різні кількості знаків, які використовуються при обчисленнях тригонометричних функцій, квадратного кореня).

Третя частина має найбільший рівень складності. Користувачу пропонується вирішити задачу, причому на її вирішенні базується моделювання. Правильна відповідь розраховується за допомогою алгоритму, який використовуються при моделюванні. Тому можна наочно побачити його роботу у вигляді числової візуалізації. Усі параметри, окрім мас куль, можуть бути дробовими. Оскільки програма виконує багато розрахунків, точність яких може суттєво відрізнятись від точності розрахунків користувача, пропонується обрати відповідь, найбільш близьку до отриманої користувачем.

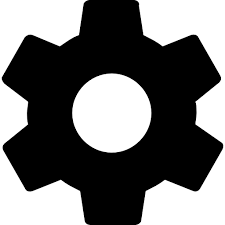
#### Меню моделювання

Тут представлено ситуації для моделювання, завантажені з файлу.



Рис. 2.3 Меню моделювання

Користувачу доступні такі дії.

* Зміна налаштувань, перехід до яких здійснюється натисканням картинки .
* Обрання потрібних для моделювання елементів. Кількість обраних елементів повинна бути від одного до шести включно. Вибір здійснюється кліком на елементі, який після цього виділений синьою рамкою. Повторний клік прибирає виділення. Обрати більше від шести елементів неможливо. Якщо не обрано жодного елементу, кнопки «Редагувати», «Видалити» та «Моделювати» стають недоступними.
* Редагування елементу шляхом подвійного кліку на ньому або за допомогою натискання кнопки «Редагувати». В останньому випадку для редагування буде відкрито останній обраний елемент.
* Створення нового елементу шляхом натискання кнопки «Створити» на панелі меню.
* Видалення останнього обраного елементу за допомогою натискання кнопки «Видалити».
* Збереження внесених змін до файлів. Налаштування зберігаються за шляхом «Sources\Settings.dat» відносно .exe файлу програми, а змінені дані заміщують зміст файлу, з якого були зчитані.
* Перехід до розділу «Моделювання».
* Повернення до попереднього розділу.

Якщо налаштування чи дані не збережені за допомогою кнопки «Зберегти», при спробі закрити додаток користувачу буде запропоновано зробити це.

#### Створення/Редагування елементу

Елементом називається сукупність параметрів, що описують ситуацію для моделювання. Кожен елемент є окремим об’єктом під час роботи програми і рядком таблиці при збереженні даних у файл.

Вікно «Створення/Редагування елементу» містить усі доступні для зміни користувачем параметри моделювання і використовується або для редагування створених раніше ситуацій для моделювання (Рис. 2.4), або для створення нових (Рис. 2.5). У випадку використання цього вікна для створення елементів, його вміст ініціалізується на основі першого елементу з «Меню моделювання» а кнопка «Застосувати» замінюється кнопкою «Додати». Якщо «Меню моделювання» не містить жодного елементу, ініціалізація відбувається на основі даних за замовченням (Рис. 2.5).

Додатне значення поля «Швидкість» означає рук кулі зліва направо. Від’ємне значення цього поля змінює напрям руху кулі на протилежний. Інші поля груп параметрів куль мають зміст, що пояснюється написами біля них.

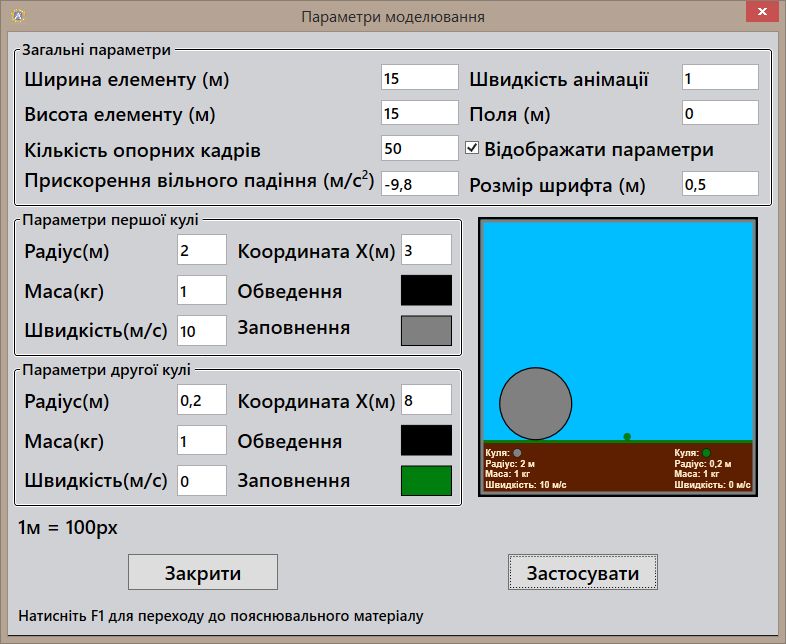


Рис. 2.4 Редагування елементу

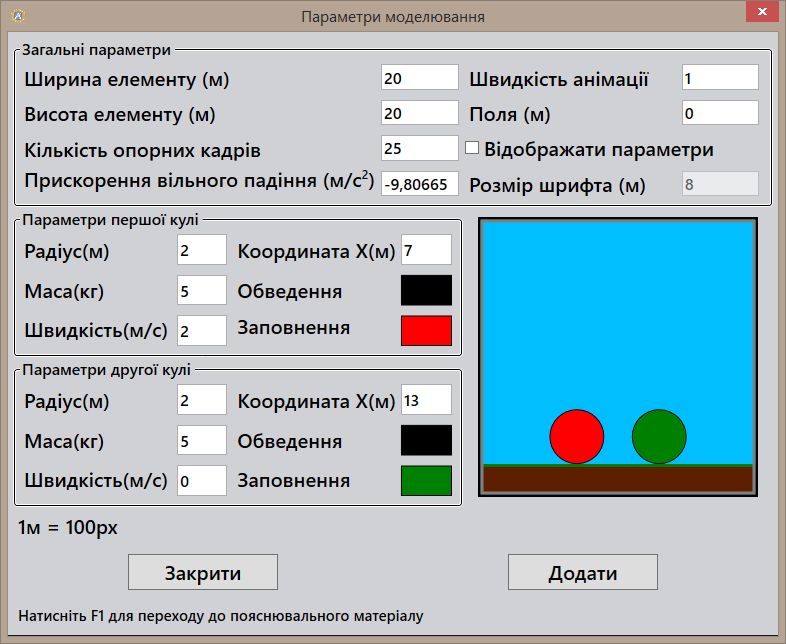


Рис. 2.5 Створення елементу за замовчанням

Розглянемо детальніше деякі поля групи «Загальні параметри».

*Кількість опорних кадрів* визначає кількість опорних точок, для яких буде розраховано позиції куль протягом секунди. Чим більше значення цього параметра, тим точніше змодельована ситуація. Не рекомендовано встановлювати значення набагато менше від максимального модуля швидкості одної з куль. Це єдине поле цього розділу, що допускає введення тільки цілих значень.

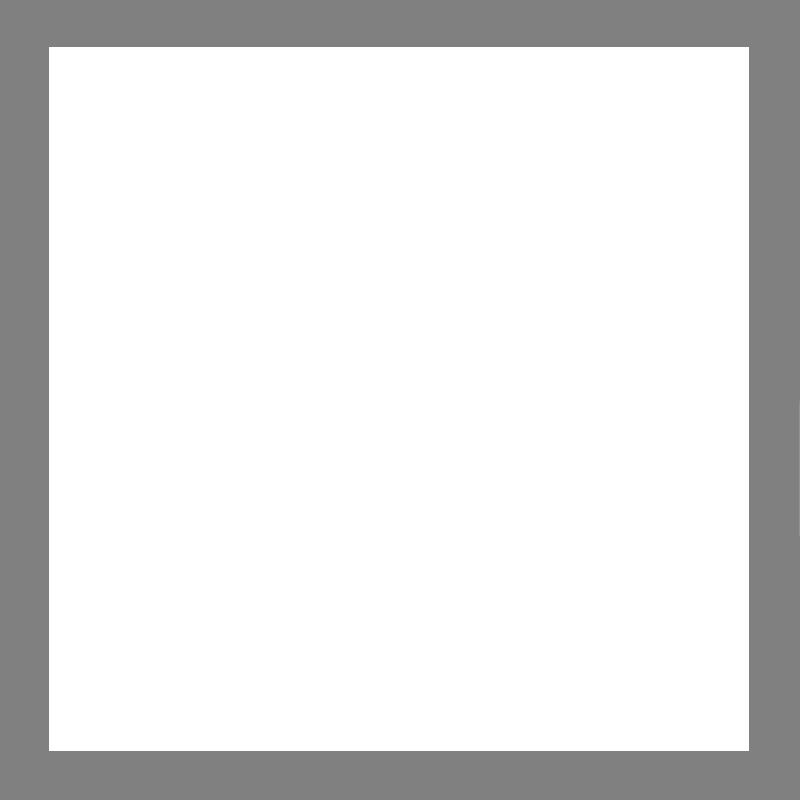
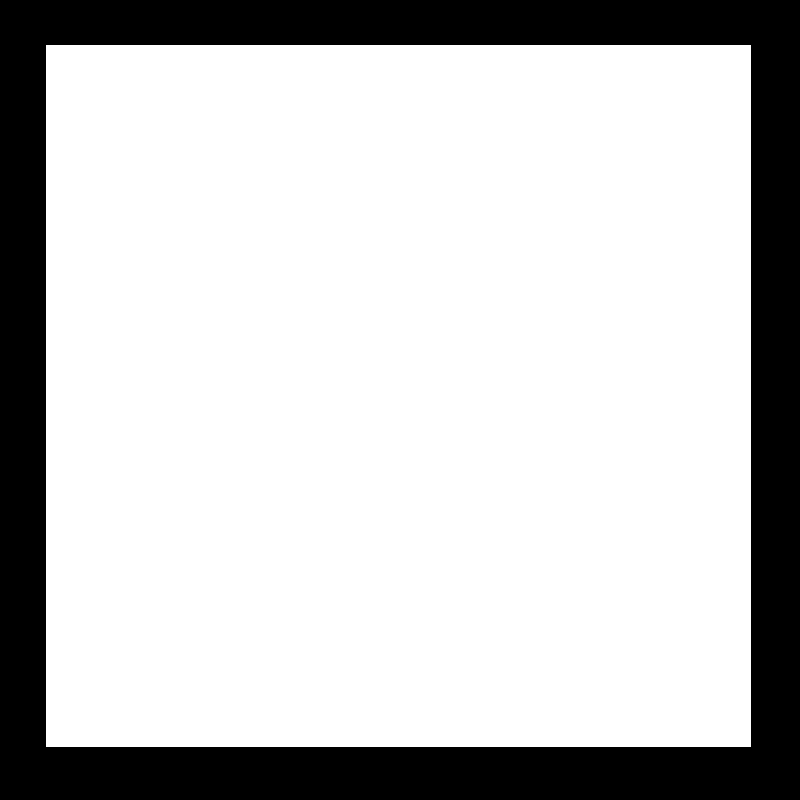
*Поля* – ті межі, при доторканні до яких однією з куль припиняється процес анімації. Додатне значення – відступ зовні від вікна відображення модельованої ситуації, від’ємне – відступ до середини вікна елементу.

Якщо до якого-небудь поля введено некоректне значення (наприклад, значення містило букви), то застосованим залишається останнє коректне введене значення.

Введена крапка замінюється на кому (кома вважається символом десяткового розділення). Введення від’ємних значень можливе тільки у поля «Прискорення вільного падіння», «Поля» та «Швидкість».

Якщо параметр «Відображати параметри» обрано, то у елементі відображуються написи, що містять основні параметри куль. Частина елементу, що являє собою землю, приймає такий розмір, щоб умістити ці написи. Якщо даний параметр не обрано, земля займає 1/10 від розміру елемента.

Змінюючи параметри, що мають миттєвий вплив на візуалізацію, користувач одразу бачить їх у правій нижній частині поточного вікна - області візуалізації, яка складається з зони для моделювання та двох рамок навколо:



* Чорна рамка демонструє кордони області поточної ситуації, що задаються параметрами «Ширина елементу» та «Висота елементу»
* Сіра рамка демонструє встановлені навколо елементу поля, що задаються відповідним параметром.

При застосуванні змін відбувається перевірка на коректність заданої ситуації, перевіряються такі умови.

* Прискорення вільного падіння повинно бути від’ємним.
* Кулі не повинні перетинатись.
* Жодна з куль не повинна виходити за поля поточного елементу.

При закритті вікна (хрестиком, кнопкою «Закрити» або клавішею «Esc»), усі внесені зміни втрачаються.

#### Налаштування

Цей розділ містить деякі параметри додатку. Розглянемо детально зміст кожного з них.

*Розташування даних* - містить поточний шлях до файлу з даними. При виборі іншого файлу програма перевіряє, чи збережені зміни у поточних даних. Якщо зміни є, пропонується користувачу зберегти їх. Далі програма очищує поточні дані та завантажує нові елементи з файлу.

*Мінімальний врахований час*. Моделювання у програмі здійснюється за опорними кадрами. Тому цілком імовірно, що за час між сусідніми кадрами кулька може здійснити кілька стрибків. Взаємодія між кулями за цей період часу не враховується. Отже, на вектор швидкості, яку має куля в кінці проміжку часу між кадрами, впливає тільки та частина стрибка, яку куля не встигла здійснити. Якщо час повного стрибка кулі менший від зазначеного в даному полі, то вважається, що куля здійснила тільки деяку кількість повних стрибків, інакше розраховується вплив частково виконаного стрибка на вектор швидкості кулі. Значного впливу на розрахунки цей параметр не має, проте чим він менший, тим точніше виконується моделювання.

*Максимальна кількість кадрів* – якщо у процесі моделювання кількість опорних кадрів перевищує задане тут значення, моделювання завершується. Допомагає уникнути переповнення пам’яті.

Наступні параметри описують поведінку елементів у розділі «Моделювання».

*Мінімальний масштаб* – параметр, що обмежує максимальне зменшення елементу при його масштабуванні. Має належати проміжку .

*Крок масштабування* – поточний масштаб елементу помножується (ділиться) на це значення при скролінгу. Має належати проміжку .

*Поля перенесення –* елемент не можна повністю перемістити за межі цих полів. Він має належати проміжку .

*Швидкість анімації* – набувається спільною анімацією в розділі «Моделювання» при першому зіткненні куль, якщо її поточна швидкість була більша від значення цього поля. Має належати проміжку .

У нижній частині вікна розташовані три кнопки: «Закрити» – закриває вікно і якщо є не застосовані налаштування, пропонує їх застосувати; «Стандартні» - присвоює всім полям значення за замовченням; «Застосувати» - застосовує зміни , але не зберігає їх до файлу. Закриття вікна відбувається після натискання кнопок «Закрити», «Застосувати» або клавіші «Esc».

#### Моделювання

У розділі моделюються ситуації, обрані користувачем у «Меню моделювання».

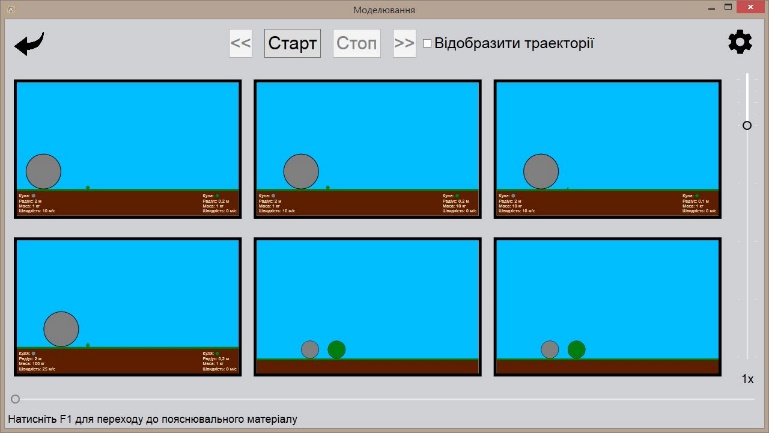
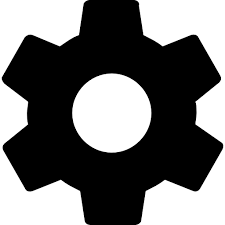


Рис. 2.6 Моделювання

Для кожного елементу виділяється максимально можливий простір і виконується його масштабування.

Доступна зміна налаштувань, перехід до яких здійснюється натисканням картинки .

Чотири кнопки верхньої панелі керують анімацією: повертають до початку, запускають/зупиняють, ставлять на паузу/поновлюють та переводять до кінця відповідно. CheckBox, розташований поруч із кнопками, дозволяє обрати, чи будуть відображені траєкторії руху куль (їх центрів) при відтворенні анімації. Якщо куля під час руху змінює напрям швидкості на протилежний, то колір її траєкторії змінюється з чорного на синій.

Трекбар знизу від області моделювання відображує прогрес загальної анімації (час якої дорівнює часу найдовшої з анімацій) та дозволяє його змінювати.

Трекбар праворуч від областей моделювання дозволяє змінювати швидкість загальної анімацій (для всіх елементів), не змінюючи при цьому відносну (в одному з елементів), що задавалась у налаштуваннях кожного елементу.

У момент першого зіткнення куль в одному з моделюючих елементів відбувається перевірка: чи перевищує поточна швидкість загальної анімації задану в полі налаштувань «Швидкість анімації». Якщо перевищує, то швидкість загальної анімації приймає значення, встановлене у цьому полі.

Елементи можна масштабувати та пересувати згідно з заданими для цього параметрами у розділі «Налаштування» у межах простору, відведеного для кожного з них. Можна відкрити вікно редагування елементу подвійним кліком на ньому.

### Структура файлів

Файли повинні знаходитись у папці «Sources», розташованій в одному каталозі з файлом .exe.

Програма використовує культуру uk-UA, тому в якості символ десяткового розділення використовується символ коми (‘,’) в усіх його частинах.

Структура файлів, використовуваних програмою, повинна бути суворо дотримана.

#### Файл налаштувань

Файл налаштувань повинен мати ім’я «Settings.dat» та наступну структуру:

MinimalCalculateTime: дробовеЧисло

MaxFramesCount: цілеЧисло

MinScale: дробовеЧисло

TranslateFields: дробовеЧисло

ScrollStep: дробовеЧисло

AverageSpeedRatio: дробовеЧисло

DataBasePath: шлях.

цілеЧисло – будь-яке ціле зі значенням від -21 474 836 до 21 474 836.

дробовеЧисло – будь-яке дробове зі значенням від до і має до 16 цифр у дробовій частині.

Порядок перечислення параметрів неважливий. Між іменем та значенням параметру мають бути один або більше символів ‘ ‘ або ‘:’. Назви параметрів не можуть бути змінені. Зменшувати кількість параметрів не можна. Усе, що записано після цих даних у файлі, не вплине на роботу програми.

#### Файл даних

Файл даних повинен мати розширення .xml та ім’я і розташування згідно з указаним у налаштуваннях, або, якщо немає відповідного файлу за вказаним шляхом, мати ім’я «Elements.xml» та бути розташованим у папці «Sources». Цей файл повинен мати структуру згідно з мовою розмітки xml та складатись з однієї таблиці, що містить стовпці у наступному порядку.

Таблиця 2.1

**Стовбці XML таблиці файлу «Elements.xml»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Стовбець | Тип/ одиниця виміру | № | Стовбець | Тип/ одиниця виміру |
| 1 | Ширина елементу | м | 11 | Колір заповнення першої кулі | ARGB |
| 2 | Висота елементу | м | 12 | Колір контуру першої кулі | ARGB |
| 3 | Кількість опорних кадрів на секунду | fps | 13 | Координата за віссю X першої кулі | м |
| 4 | Швидкість анімації | X | 14 | Швидкість першої кулі | м/c |
| 5 | Поля | м | 15 | Радіус другої кулі | м |
| 6 | Розмір шрифту короткої інформації про кулю | м | 16 | Маса другої кулі | кг |
| 7 | Чи відображується інформація | bool | 17 | Колір заповнення другої кулі | ARGB |
| 8 | Прискорення вільного падіння | м/с2 | 18 | Колір контуру другої кулі | ARGB |
| 9 | Радіус першої кулі | м | 19 | Координата за віссю X другої кулі | м |
| 10 | Маса першої кулі | кг | 20 | Швидкість другої кулі | м/c |

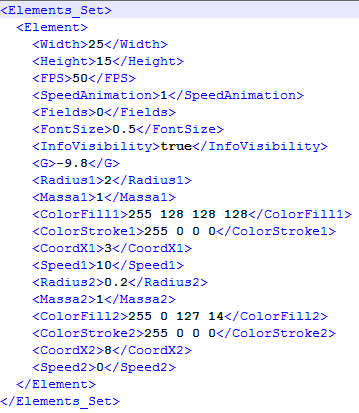


Рисунок 2.7 Приклад файлу «Elements.xml»

### Параметри за умовчанням

#### Налаштування

Якщо не вдалось зчитати налаштування або в разі застосування стандартних параметрів у розділі «Налаштування» використовуються наступні дані.

Таблиця 2.2

**Налаштування за замовчанням**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр | Значення | Тип/ одиниця виміру | Параметр | Значення | Тип/ одиниця виміру |
| Мінімальний врахований час | 0,0001 | с | Крок масштабування | 1,2 | X |
| Максимальна кількість кадрів | 1000 | шт. | Швидкість анімації | 0,5 | X |
| Мінімальний масштаб | 0,1 | X | Шлях до файлу з даними | \Sources\ Elements.xml | UNC |
| Поля перенесення | 15 | px | - | - | - |

#### Дані

Оскільки вікно створення елементів необхідно ініціалізувати даними, але не завжди існує хоча б один створений елемент, необхідно створювати елемент за замовченням із наступними значеннями параметрів.

Таблиця 2.3

**Параметри елементу за замовчанням**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Параметр | Значення | Тип/ одиниця виміру | Параметр | Значення | Тип/ одиниця виміру |
| Ширина елементу | 20 | м | Колір заповнення першої кулі | 255 255 0 0 | ARGB |
| Висота елементу | 20 | м | Колір контуру першої кулі | 255 0 0 0 | ARGB |
| Кількість опорних кадрів на секунду | 25 | fps | Координата за віссю X першої кулі | 7 | м |
| Швидкість анімації | 1 | X | Швидкість першої кулі | 2 | м/c |
| Поля | 0 | м | Радіус другої кулі | 2 | м |
| Розмір шрифту | 1 | м | Маса другої кулі | 5 | кг |
| Чи відображується інформація | false | bool | Колір заповнення другої кулі | 255 0 255 0 | ARGB |
| Прискорення вільного падіння | -9.80665 | м/с2 | Колір контуру другої кулі | 255 0 0 0 | ARGB |
| Радіус першої кулі | 2 | м | Координата за віссю X другої кулі | 135 | м |
| Маса першої кулі | 5 | кг | Швидкість другої кулі | 0 | м/c |

### Програмні та апаратні вимоги

Апаратні вимоги.

* Частота процесора 1.5GHz або вище
* 1024 MB RAM
* 20 MB вільного місця на жорсткому диску
* Роздільна здатність монітору 1280 x 720 або вище

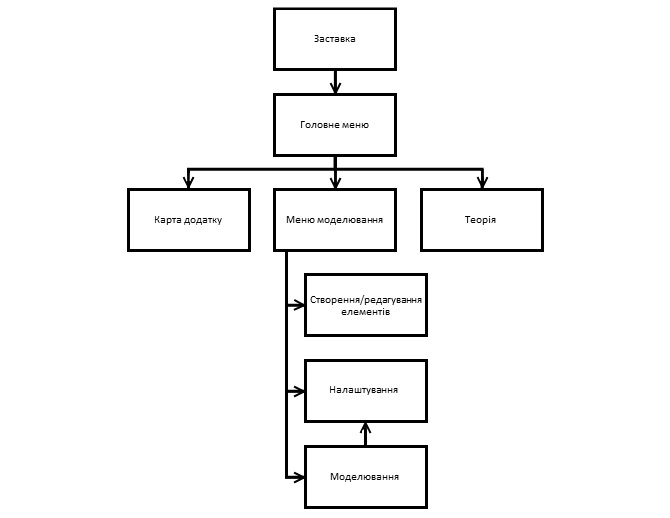
Програмні вимоги.

* .Net Framework 4.6.1

Підтримуючі операційні системи.

* Windows 10
* Windows 8, Windows 8.1
* Windows 7 with SP1

### Структурна схема програми

На схемі представлено основні переходи у додатку, без урахування можливих переходів з розділу «Карта додатку»:

## Використані програмні продукти

* Microsoft Visual Studio 2017 Enterprise
* Microsoft Blend для Visual Studio 2017
* Microsoft Word 2016
* GeoGebra 5
* paint.net 4.0
* Notepad ++

## Комплектація програми

Для встановлення програми необхідно скопіювати папку «Modeling» на жорсткий диск. ЇЇ зміст такий.

1. Modeling.exe – виконуваний файл програми;
2. Папка «Sources». Її зміст наступний.
   * + 1. Settings.dat – файл налаштувань програми;
       2. Elements.xml – файл із базою даних;
       3. Папка Explanations, що містить наступні фали з поясненнями до відповідних розділів.

ElementCreater.html;

MenuMain.html;

MenuModeling.html;

Modeling.html;

* + - 1. Папка Theory. Її зміст може бути змінений, проте передбачається наявність наступних файлів з теоретичним матеріалом.

«1 Основні положення.htm»;

«2 Умовні позначення.htm»;

«3 Рух кулі, кинутої під кутом до горизонту.htm»;

«4 Застосування розрахунків у програмі.htm»;

«5 Зіткнення.htm»;

Кожний файл із розширенням .htm потребує наявності в своєму каталозі папки, що містить необхідні файли для його функціонування.

# ВИСНОВКИ

Під час виконання роботи я зіткнувся з такими проблемами як проектування класів програми та їх взаємодії, розробка алгоритму для моделювання і перевірка коректності його роботи.

Моделювання повністю жорсткого та пружного зіткнення куль на площині дозволило проаналізувати наступні ситуації.

*Кулі сильно відмінні за радіусом та масою.* Якщо більша куля значно важча за меншу, то при зіткненні остання відскакує, набувши значної швидкості, і досить слабко впливає на зміну швидкості руху більшої кулі. Якщо ж важчою є менша за радіусом куля, то при зіткненні вона майже не переміщується, а велика куля може перелетіти через неї, якщо володіє достатньою швидкістю, або відскакує у напрямку протилежному до початкового напрямку своєї швидкості.

Якщо *кулі сильно відмінні за радіусом і мають приблизно однакові маси*, то переліт більшої кулі через меншу - менш вірогідний порівняно з попередньо розглянутою ситуацією, але залишається можливим.

Моделювання ситуації, коли *кулі відмінні за масою і мають однакові радіуси* дозволилоперевірити коректність розрахунків, оскільки цей випадок є добре відомим.

Отже, дана робота може стати добрим практичним посібником при вивченні абсолютно пружного жорсткого зіткнення а також законів збереження імпульсу й енергії для учнів старшого шкільного віку. Також можливе використання роботи як основи при дослідженні зіткнень у реальних умовах, які можна вважати пружними, наприклад зіткнення колеса автомобіля з каменем.

# ДЖЕРЕЛА

**Книги:**

1. Засєкіна Т. М. Фізика: підручник для 10 кл. загальноосвіт. навч. закл. (профільн. Рівень)/ Т.М. Засєкіна, М.В. Головко. -К.: Педагогічна думка, 2010. – 304 с., іл., табл.
2. Фленов М. Е. Библия C#.- 2-е изд*.,* перераб. и доп. – СПб.: БХВ-Петербург, 2011. – 560с.: ил.;

**Інтернет джерела:**

1. *1.21. Упругие и неупругие соударения*. (б.д.). Отримано з Открытый Колледж. Физика.: http://www.physics.ru
2. *Анимации в WPF*. (б.д.). Отримано з Хабрахабр: habrahabr.ru
3. *Графика и анимация WPF*. (б.д.). Отримано з professorweb: professorweb.ru
4. *Объектно-ориентированное программирование (C# и Visual Basic)*. (б.д.). Получено из Microsoft Developer Network: msdn.microsoft.com
5. *Привязка, стили и команды WPF*. (б.д.). Отримано з professorweb: professorweb.ru

# ДОДАТОК А

В об’єкті кулі наявні такі властивості:

private Point Coord;

public double Speed;

public double SpeedAngle;

public double Radius;

private Element Element;

Об’єкти типу Element містять властивості:

public double G;

public double Ground

Коди геттерів та сеттерів цих властивостей об’ємні і розглядити їх потрібно у контексті відповідних класів, тому вони не наводяться.

Визначення позиції кулі через заданий інтервал часу виконується за допомогою наступного методу об’єкту кулі:

public void moveForce(double deltaTime){

double x = Coord.X; double y = Coord.Y;

double speedY = Speed Sin(SpeedAngle);//Проекція швидкості на вісь **Y**

double speedX = Speed Cos(SpeedAngle); //Проекція швидкості на вісь **X**

x += speedX deltaTime;

//Наступний код виконується тільки якщо куля у польоті або швидкість напрямлена вгору

if (y + Radius < Element.GroundY || speedY > 0){

//Дискримінант рівняння ( D завжди більше або дорівнює

double D = speedY speedY - 2 Element.G (Element.GroundY - y - Radius);

//(-b-sqrt(D))/2a - час польоту до першого зіткнення з землею

double time = (-speedY - Sqrt(D)) / Element.G;

if (time >= deltaTime)//якщо за час deltaTime, куля не впаде на горизонтальну поверхню, з якої злетіла

{

y -= speedY deltaTime + Element.G deltaTime deltaTime / 2;//h=v0t+at^2/2

speedY += Element.G deltaTime;

} else//якщо відбудеться одне чи більше падінь кулі на горизонтальну поверхню за час deltaTime

{

//швидкість кулі в момент падіння (уся енергія перейшла в кінетичну).

speedY += Element.G time;

speedY = Abs(speedY);//швидкість повинна мати додатне значення

deltaTime -= time;//час руху кулі, для якого слід розрахувати її переміщення

time = 2 -speedY / Element.G;//час, за який куля описує дугу

//якщо інтервал стрибку дуже малий, наступні дії не мають сенсу.

if (time > MinimalCalculateTime){

while (deltaTime > time)//визначаємо, скільки ще лишилося часу після виконання усіх стрибків.

deltaTime -= time;

y = (Element.GroundY - Radius) - speedY deltaTime - Element.G deltaTime deltaTime / 2;

speedY += Element.G deltaTime;

}

}

}

if(y + Radius > Element.GroundY)//контролюємо, щоб куля не «провалювалась» під поверхню.

y = Element.GroundY - Radius;

//новий стан об’єкту

Coord = new Point(x, y);

Speed = Sqrt(Pow(speedX, 2) + Pow(speedY, 2));

SpeedAngle = Atan2(speedY, speedX);

}