

feelings of motivational and volitional sphere. It is STEAM education that is a means of learning that provides the formation of vital competencies in the context of interpersonal relationships. This article presents an example of the use of elements of STEAM education, aimed at developing the ecological culture of the individual, from their own work experience.

Keywords: *STEAM-education, STEM-competencies, ecological education, practical-research projects.*

УДК 373.5:53; 004.9

Іванов С. А.

**РОЗВ'ЯЗАННЯ ЗАДАЧ З ФІЗИКИ
МЕТОДАМИ КОМП'ЮТЕРНОГО МОДЕЛЮВАННЯ
(VPython, GlowScript, Trinket)**

Основна увага статті приділяється пізнавальному потенціалу комп'ютерного моделювання фізичних явищ та розв'язання фізичних задач у контексті упровадження STEM-освіти. Показано можливості програмного середовища 3D-моделювання GlowScript у реалізації платформи Trinket.io. Наведено приклади розв'язання задач з різних розділів шкільного курсу фізики.

Ключові слова: *STEM-освіта, комп'ютерне моделювання, 3D-моделювання, алгоритмічне мислення, програмне середовище, VPython, GlowScript.*

Постановка проблеми. Кінцевою метою навчання фізики є розуміння реального фізичного світу, однак сучасні шкільні підручники з цього предмету пропонують академічний підхід до науки. Зокрема в них перебільшують спрощення – тіла літають без опору повітря, рухаються без тертя або без опору води, маятники коливаються лише за малих кутів, орбіти завжди кругові, а в будь-який момент часу рухається не більше двох частинок тощо. Крім того,

вчитель у своїй викладацькій роботі обмежується лише завданням навчити учнів правильно писати умови задачі, зрозуміти яку формулу потрібно використати, підставити вихідні значення та отримати результат. Такий підхід виправданий у разі початкового пізнання основних законів природи. Але реальний світ набагато складніший і набагато цікавіший, тому слід виходити з того, що фізика – не тільки теоретична наука, але й практична, тобто експериментальна. У зв'язку з цим варто зазначити, що однією з найважливіших складових компетентності учня як результату його фізичної освіти є інформаційно-комунікаційна компетентність, в першу чергу, вміння будувати інформаційні моделі фізичних явищ та законів, а також моделі розв'язання фізичних задач [6].

На наш погляд, за критерієм витратності для навчальних закладів можна виділити три напрями практичної діяльності, де учні можуть набути дослідницьких компетентностей за допомогою інформаційно-комунікаційних технологій:

- комп'ютерне моделювання (практично без витрат);
- цифрові лабораторії на Ардуїно-подібних платформах (маловитратні);
- розвинуті цифрові лабораторії (високовитратні), наприклад, National Instrument, Vernier, Einstein, LabDisc, Nova Link, Архімед тощо).

Аналіз останніх досліджень і публікацій. У цій статті основну увагу зосереджено на першому напрямі, тобто комп'ютерному моделюванні. Під терміном «модель» будемо розуміти «...деякий об'єкт-замінник об'єкта-оригіналу, що забезпечує вивчення деяких істотних, з погляду дослідника, властивостей оригіналу», а під терміном «моделювання» – заміщення одного об'єкта іншим із метою здобуття інформації про найважливіші властивості об'єкта-оригіналу за допомогою об'єкта-моделі [3, 5, 7]. Множину видів моделей можна звести до двох – матеріальних та інформаційних. Матеріальні моделі відтворюють фізичні, геометричні та інші властивості об'єктів моделювання (макети будь-яких споруджень, технічних пристроїв, модель

молекули ДНК тощо). Серед типів інформаційних моделей найбільш пізнавальний потенціал мають комп'ютерні моделі, реалізовані за допомогою певного програмного середовища. Зауважимо, що до комп'ютерного моделювання не слід відносити використання комп'ютерів при будь-якому дослідженні або вирішенні проблеми, наприклад, застосування табличного процесора Excel [8].

Застосування комп'ютерного моделювання при дослідженні фізичних явищ має певну історію. Однією з найбільш цитованих робіт у цьому напрямі є книга «Компьютерное моделирование в физике», в якій автори чітко визначилися з метою, яка полягає у «...спробі впровадити принципи комп'ютерного мислення у вивченні фізики» [4]. Іншими словами, для вивчення фізики найбільш важливим є створення умов, в яких учні мають можливість навчить комп'ютер моделювати фізичні системи, тобто писати відповідні програмні коди. Не заперечуючи важливість комп'ютерного моделювання фізичних досліджень, варто звернути увагу на пізнавальний потенціал комп'ютерного моделювання для розв'язання задач у рамках шкільного курсу фізики.

Виклад основного матеріалу. Сьогодні існує достатня кількість програмних засобів, які використовуються для комп'ютерного моделювання, серед яких особливе місце займає GlowScript – хмарне середовище, яке дозволяє здійснювати 3D-моделювання фізичних явищ безпосередньо на веб-сайті [9]. В основі GlowScript покладено візуальну мову програмування Visual Python (або VPython), використання якої вже має певний досвід у моделюванні фізичних явищ [10]. VPython це візуальне розширення мови програмування Python спільно з 3D-графічним модулем під назвою «Visual», яку заснував Девід Шерер у 2000 році. VPython надає користувачам можливості створювати тривимірні віртуальні моделі за допомогою обмеженого набору об'єктів, які показуються на холсті (canvas) (Рис. 32):










arrow 	box 	cone 	cylinder 	Ellipsoid 
helix 	text «Дано»	pyramid 	Ring 	Spher e 

Рис. 32. Набор об'єктів VPython

Програмне моделювання матеріальних тіл дозволяє користувачам більше концентруватися, по-перше, на фізичній сутності задачі, по-друге, – на обчислювальному аспекті програмного коду. Саме ці особливості мови VPython зумовило її вибір для комп'ютерного моделювання задач шкільного курсу фізики.

Об'єкти в мові VPython як представники тих чи інших класів можуть бути створені з певним набором атрибутів. Наприклад, простий об'єкт *tube* класу *cylinder* створюється за допомогою вказівки атрибутів, таких як координати положення, вісь напрямку, прозорість, видимість, колір тощо.

До цього переліку можна додавати нові атрибути, що є дуже корисним для застосування VPython для розв'язання фізичних задач. Наприклад, до атрибутів об'єкту *spring* класу *helix* (пружина) має сенс додати атрибути *spring.coils=20* (число витків) та *spring.thickness=0.1* (товщина дроту). До об'єкту класу *ball* можна вказати атрибут *trail*, який підключає слід руху об'єкту у межах екрану. А до об'єкту, який представляє, наприклад, об'єкт *грузик* у вигляді сфери (*sphere*) або паралелепіпеду (*box*) за ім'ям *load*, слід вказати атрибут його маси: *load.m=100*. Програмні засоби VPython дозволяють створювати об'єкти контролю процесу: кнопки (*button*), повзунок (*slider*), перемикач (*toggle*), меню (*menu*) тощо. Крім того, в арсеналі VPython є практично всі операції с мишою та клавіатурою. Нарешті, підключаючи ті чи інші бібліотеки можна відображати результати моделювання або розв'язання фізичних задач на графіках. Більше детально можливості мови VPython надані на порталі

VPython IDLE [10]. Важливою особливістю VPython є можливість візуалізації векторних величин, такі як швидкість та прискорення, тобто анімація фізичного процесу, що забезпечує можливість учням бачити рух об'єктів, про які йдеться у фізичній задачі, насамперед, рух під впливом сил, зокрема, земного тяжіння. Наприклад, об'єкт *cart* класу *box*, пофарбований зеленим кольором, рухається зі швидкістю *v*. Об'єкт *cart* описується у такому вигляді:

```
cart = box(pos=vector(-0.5,0,0), size=(0.1, 0.04, 0.06), color=color.green)
```

Анімація створюється шляхом запровадження циклу *while*, з параметром часу *t*, який змінюється від 0 до певного значення з кроком, що має невелике значення, наприклад, $\Delta t=0.01$. Тіло циклу містить інструкції комп'ютеру як змінити позицію об'єктів знову і знову, імітуючи рух об'єктів по екрану.

Ці інструкції (вирази) вказують комп'ютерові як розрахувати рівнодіючу силу на об'єкти, як розрахувати новий імпульс кожного об'єкта, використовуючи рівнодіючу силу та принцип імпульсу, як знайти нове положення об'єктів, використовуючи момент імпульсу (Рис. 33).

До початку циклу *while* об'єкту *cart* треба додати атрибут масу (*m*) і початковий імпульс (*p*): *cart.m = 0.80* та *cart.p = cart.m*vector(0.5,0,0)*. У тілі циклу *while* моделюється рух об'єкту *cart*, поточне положення якого визначається згідно рівняння:

$$\vec{r}_f = \vec{r}_i + \vec{V}_{avg} * \Delta t \quad \text{або} \quad \vec{r}_f = \vec{r}_i + (\vec{p} / m) * \Delta t, \quad (1)$$

де вектор \vec{r}_i - положення центру об'єкту; *m* – маса, \vec{p} – момент імпульсу.

Програмне це виглядає таким чином (*deltat* – це Δt):

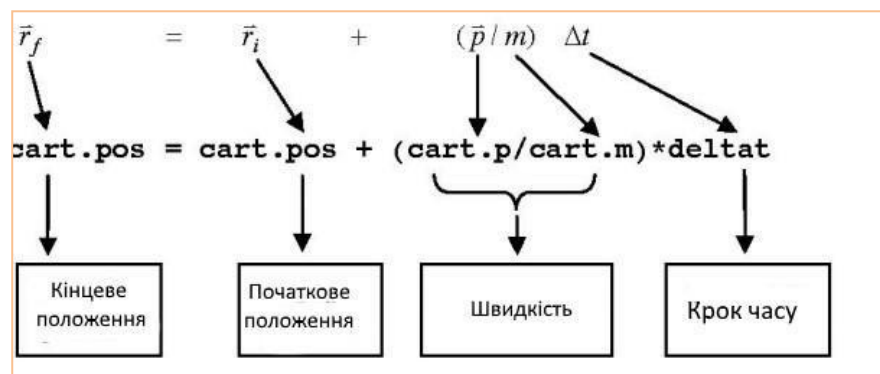


Рис. 33. Розрахунок нового положення об'єкта *cart*

У циклі обов'язковим є оператор *rate(speed)*, де параметр *speed* може дорівнювати 1000. Цей оператор примушує стільки раз здійснювати цей цикл за 1 секунду, саме у цю коротеньку паузу об'єкти створюються та відображаються на екрані. За рахунок цього складається враження плавного руху об'єктів. Підсумовуючи, цей фрагмент програми виглядає приблизно так:

```
t = 0
deltat = 0.001
while t < 3:
    cart.pos = cart.pos + (cart.p/cart.m)*deltat
    t = t + 1
```

Незважаючи на очевидні переваги VPython у застосуванні його для моделювання фізичних явищ, демонстрування фізичних законів та розв'язання фізичних задач він у своєму первісному варіанті не знайшов свого місця у шкільних програмах. Це стосується взагалі програмування, за останні часи виявилось, що учні, навіть ті, хто закінчив спеціальні курси з програмування, реально не в змозі запрограмувати будь-яку реальну задачу. Ці проблеми добре відомі учителям інформатики і докладно представлені в педагогічній літературі [2]. Головне тут полягає у тому, що у більшості учнів, по перше, не сформовано алгоритмічне мислення, а, по друге – він не бачить відразу результатів своєї роботи. Програмування потребує уважності і терпіння, тобто тих якостей, яких не вистачає школярам, які достатньо швидко опановують сучасні гаджети.

Поява у 2011 році середовища програмування GlowScript та у 2014 році можливість використовувати мову RapydScript для підтримки програм VPython в середовищі GlowScript, може суттєво змінити ставлення учнів до програмування.

Практично водночас було розроблено програмне середовище Trinket.io, яке полегшує кодування в GlowScript, саме тому воно в останні часи поширюється для навчання. На малюнку 34 показано робочу область Trinket.io,

яка ділиться наполовину. Код програми вводиться зліва, а на правому боці можна бачити і результати виконання цього коду. Як можна бачити на малюнку у середовищі використовується версія 2.7 мови VPython.

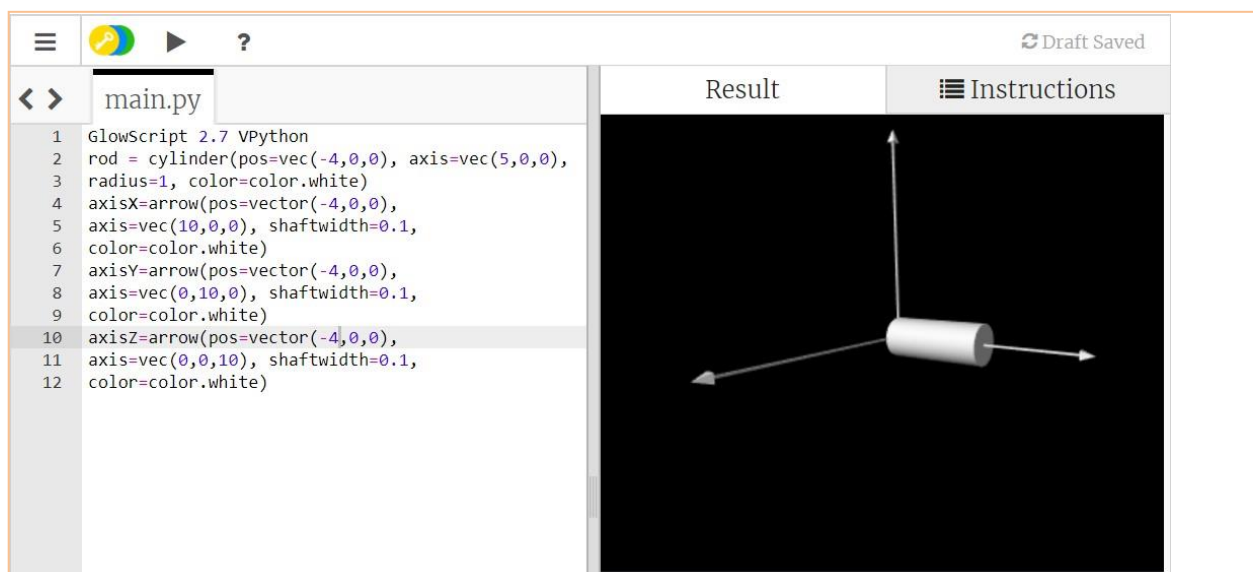


Рис. 34. Код та його результат

Застосування технології Trinket.io для моделювання фізичних явищ відкриває нові, широкі можливості, зокрема у розв'язанні шкільних задач. Очевидні переваги використання цієї технології полягають, насамперед, в одночасному вивченні як фізики, так і основ програмування. Крім того, психологічно дуже важливо, щоб учень відразу бачив би результати застосування програмного коду. Звичайно, що написання коду не є простою справою.

Тому можливий інший варіант застосування цієї технології, який полягає в розробці шаблонів моделювання основних, типових задач шкільного курсу фізики. Такий підхід дозволяє вчителю ставити перед учнями справжні дослідницькі завдання, шляхом, наприклад, побудови графічного відображення впливу тих чи інших вхідних параметрів (даних фізичної задачі) на результат. Тобто, учень, усвідомлюючи фізичний смисл задачі, може втручатися в програмний код, змінюючи ці параметри. Таким чином, учень зацікавлений аналізувати код, вивчаючи основи мови VPython в оточенні GlowScript та Trinket.io, та глибше занурюватися в мистецтво програмування.

Тепер, продемонструємо застосування технології Trinket.io для розв'язання задач шкільного курсу фізики. Почнемо з розв'язання типової задачі з кінематики кинутого вгору тіла. У шкільному курсу фізики дається проста формула зв'язку висоти кидка з початковою швидкістю. Учень просто підставляє дані у формулу та, отримуючи результат, остається задоволеним, не бачачи фізичного процесу.

У комп'ютерному моделюванні можна ускладнити завдання таким чином. М'яч, який має певну пружність, вилітає з циліндра з початковою швидкістю під певним кутом до горизонту. Опір повітря не враховується. Треба визначити скільки раз і на яку висоту підніметься м'яч. Таку задачу вже складно розв'язати за формулою. Спочатку було розроблено алгоритм розв'язання цієї задачі за допомогою мови візуального програмування DRAKON (Мал. 3) [1].

На підставі цього алгоритму було написано програмний код, що моделює фізичний процес у даній задачі, який частково відображений на рис. 35.

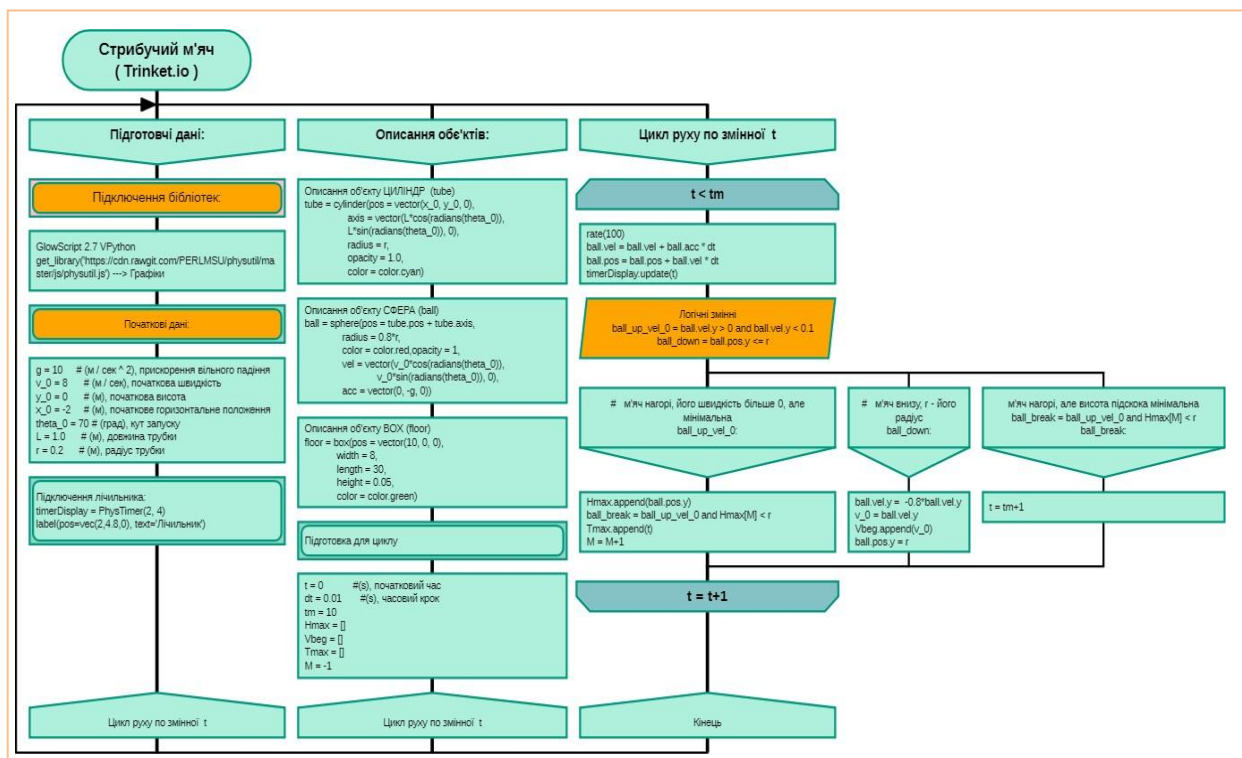


Рис. 35. Дракон-схема задачі «Стрибаючий м'яч»

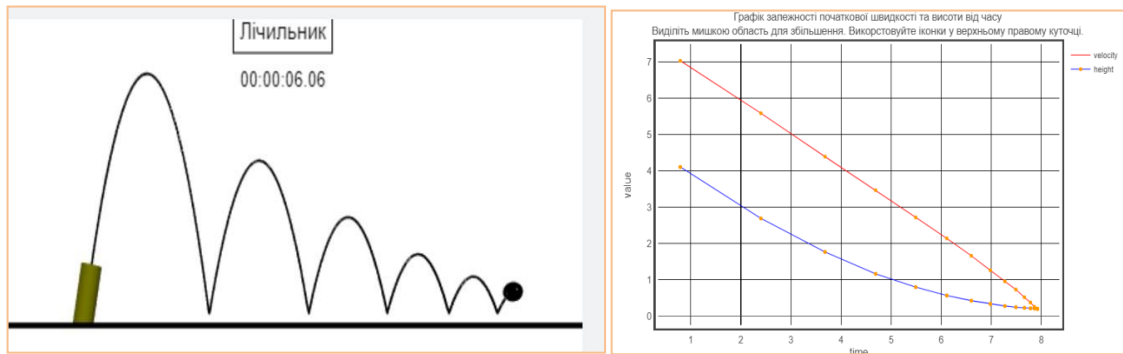


Рис. 36. Результати виконання програмного коду

Дуже цікаві задачі на тему «Основи кінематики», серед яких відома задача на рух лодки. Знову ж таки комп'ютерне моделювання дозволяє ускладнити задачу, наприклад, знайти початкову швидкість та кут руху човна (що моделюється об'єктом класу *ellipsoid*), який має потрапити в протоку на протилежному березі (Рис. 37). Можливості середовища GlowScript дозволяють слідкувати за рухом човна на екрані монітора, використовуючи об'єкт *trail* (стежка).

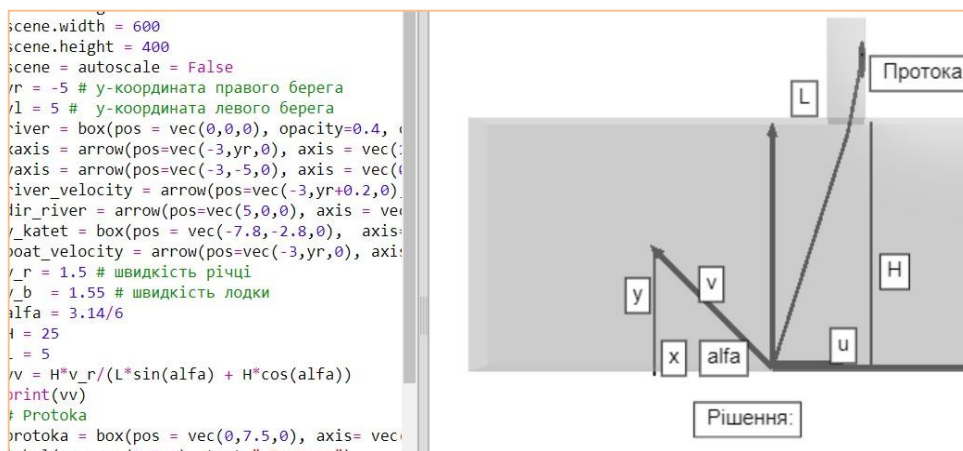


Рис. 37. Задача на рух лодки

Розвиваючи ідею про розширення постановки фізичних задач у межах шкільного курсу, розглянемо приклад такого підходу у задачі на застосування закону Гука. Комп'ютерне моделювання цього закону на прикладі пружини в середовищі GlowScript (Trinket.io) дозволяє оголосити об'єкт *spring* класу *helix*, в якому можна ввести нові атрибути, такі як радіус пружини і дроту, з якого її зроблено, та кількість витків (Рис. 38):

spring=helix(pos=point_ot,axis=sinker.pos-point_ot, radius=1.0, constant=k, thickness=thick, coils=N_coils, color=color.black)



Рис. 38. Моделювання закону Гука

Під час моделювання такої задачі деформація пружини дозволяє наочно показати її зсув і точку рівноваги сил тяжіння і пружності. Можна ще більше ускладнити задачу, поставивши дослідницькі завдання, наприклад, дослідити силу пружності в залежності від матеріалу пружини, товщину дроту, радіусу пружини, числа витків тощо. Результати такого дослідження слід представляти графічно.

Нарешті, наведемо приклад застосування програмного середовища GlowScript для моделювання задачі на закон заломлення світла. На рис. 39 показано моделювання падіння світла під різними кутами (alfa), відображаючи кути заломлення в іншому оптичному середовищі та взаємозв'язок між цими параметрами. В цьому випадку теж є місце для дослідницьких робіт, ставлячи завдання відобразити залежність кутів заломлення від кутів падіння світла, змінюючи вхідні дані, наприклад, показник заломлення середовища.

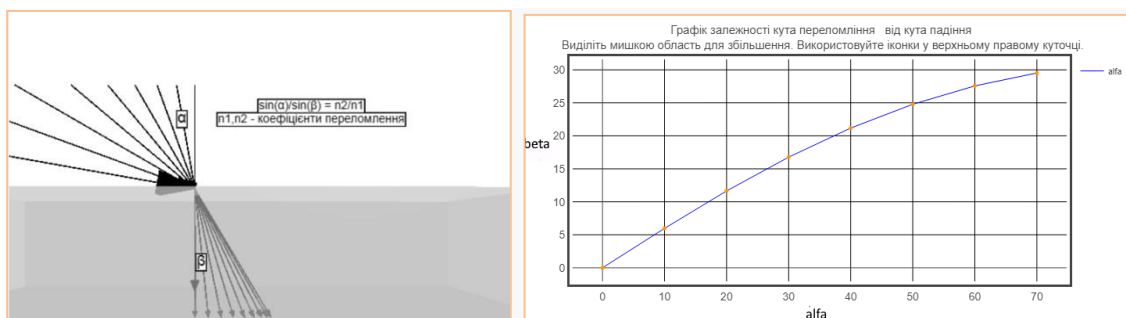


Рис. 39. Результати виконання програмного коду

Висновки. Комп'ютерне моделювання є ефективним засобом пізнання фізичного світу, яке можна упроваджувати як на уроках фізики, так і у позакласній роботі. Учителі, які застосовують комп'ютерне моделювання,

мають ставити такі цілі:

- показати учням сутність, переваги та недоліки комп'ютерного моделювання;
- навчити учнів поєднувати та порівнювати моделювання з експериментом;
- розробляти або запозичувати шаблони типових задач з фізики;
- надати учням первісні відомості про мову програмування VPython та програмні середовища GlowScript та Trinket.io;
- навчити учнів алгоритмічному мисленню;
- навчити учнів користуватися шаблонами фізичних задач та ставити дослідницькі завдання;
- навчити учнів графічно відображати результати моделювання.

Своєю чергою учні отримають початкові компетентності у:

- розумінні і формулюванні конкретних навчальних цілей;
- вмінні розв'язувати фізичні задачі шляхом комп'ютерного моделювання;
- вмінні використовувати шаблони розв'язання фізичних задач з метою проведення певних досліджень, шляхом зміни вхідних даних задачі;
- написанні простіших алгоритмів та програмних кодів, що моделюють фізичні задачі;
- у візуальному відображенні результатів дослідження;
- у вмінні здійснювати нові дослідницькі завдання.

Список використаних джерел

1. Визуальный язык ДРАКОН [Електронний ресурс] / https://drakon.su/drakon_editor_web
2. Ворон Ю. Чому онлайн-курси не навчають вас програмувати? [Електронний ресурс] / <https://dou.ua/lenta/articles/online-course-is-not-enough/>

3. Гетманова Е.Е. Моделирование физических процессов в VPython, ФинАрт, Харьков, – 2004.

4. Гулд, Х. Компьютерное моделирование в физике (в 2-х томах) / Х. Гулд, Я. Тобочник. М.: «Мир», – 1990. – 350 с.

5. Малютин, В.М. Компьютерное моделирование физических явлений: Учебное пособие / В.М. Малютин, ЕА. Складорова. – Томск: Изд-во ТПУ, – 2004. – 156 с.

6. Мерзликін О.В. Засоби інформаційно-комунікаційних технологій підтримки навчальних досліджень у профільному навчанні фізики // Інформаційні технології і засоби навчання, 2015, – Том 48, – №4. – С. 58–86.

7. Шарапов О.Д., Дербенцев В.Д., Семьонов Д.Є. Економічна кібернетика Навч. посібник. – К.: КНЕУ, 2004. – 231 с.

8. Теплицький І.О. Комп'ютерне моделювання механічних рухів у середовищі електронних таблиць / І.Теплицький, С.Семеріков // Фізика та астрономія в школі. – 2002. – №5. – С.41–46.

9. Introduction to GlowScript and VPython [Електронний ресурс] / <https://www.compadre.org/PICUP/exercises/exercise.cfm?I=130&A=gsvpythonintro>.

10. VPython Classic VPython 6.11 based on wxPython [Електронний ресурс] / <http://vpython.org/contents/docs/index.html>

Иванов Сергей. РЕШЕНИЕ ЗАДАЧ ПО ФИЗИКЕ МЕТОДАМИ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ (VPython, GlowScript, Trinket)

Основное внимание статьи уделяется познавательному потенциалу компьютерного моделирования физических явлений и решения физических задач в контексте внедрения STEM-образования. Показаны возможности программной среды 3D-моделирования GlowScript в реализации платформы Trinket.io. Приведенные примеры решения задач по различным разделам школьного курса физики.

Ключевые слова: *STEM-образование, компьютерное моделирование, алгоритмическое мышление, программная среда, 3D-моделирование, VPython, GlowScript.*

Ivanov Sergey. SOLUTION OF PHYSICAL TASKS BY METHODS OF COMPUTER MODELING (VPython, GlowScript, Trinket)

The main focus of the article is on the cognitive potential of computer modeling of physical phenomena and solving physical problems with the context of STEM-education implementation. The capabilities of the GlowScript 3D modeling software in the implementation of the Trinket.io platform are shown. The examples of solving problems in various sections of the school course of physics.

Keywords: *STEM-education, computer modeling, algorithmic thinking algorithmic thinking, software environment, 3D-modeling, VPython, GlowScript.*

УДК 378.011.3–051

Клунко Р. Ю.

**ПРОБЛЕМА ФОРМУВАННЯ ПРАВОВОЇ КОМПЕТЕНТНОСТІ
МАЙБУТНІХ ВЧИТЕЛІВ ЗАГАЛЬНООСВІТНІХ ШКІЛ УКРАЇНИ В
КОНТЕКСТІ STEM-ОСВІТИ**

Стаття присвячена розгляду проблеми формування правової компетентності майбутніх вчителів в умовах STEM-освіти. Обґрунтовано необхідність компетентнісного підходу до підготовки сучасних педагогічних кадрів та важливість якісної правової підготовки вчителів будь-якого профілю у контексті STEM-освіти. Автор висвітлює основні компоненти правової компетентності майбутніх вчителів. А також розкриває основні напрями її реалізації в умовах вищого педагогічного закладу. У статті досліджується ефективність формування правової компетентності майбутніх учителів, що можлива при обов'язковому дотриманні створених у процесі професійної підготовки студентів педагогічних умов.

Ключові слова: *правова компетентність, правова компетенція.*