

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет комп'ютерних наук, фізики та математики**

Кафедра фізики та методики її навчання

**ВИКОРИСТАННЯ ВІРТУАЛЬНИХ СИМУЛЯЦІЙ ПРИ ВИКЛАДАННІ
РОЗДІЛУ «ФІЗИКА АТОМА ТА АТОМНОГО ЯДРА» В ГІМНАЗІЯХ**

Кваліфікаційна робота (проект)

на здобуття ступеня вищої освіти «бакалавр»

Виконав (ла): студент 4 курсу, групи 15-411
Спеціальності 014 Середня освіта (Фізика)
Освітньо-професійна програма
Середня освіта (Фізика)
Югай Марк Едуардович

Керівник
кандидатка педагогічних наук, доцентка
Єрмакова-Черченко Н.О.

Рецензент
докторка педагогічних наук, доцентка
Валько Н.В.

Херсон – 2021

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. Використання симуляцій фізичних процесів як невід’ємна необхідність для процесу викладання фізики	5
1.1. Актуальність використання в освітньому процесі інтерактивних засобів та симуляцій.....	5
1.2. Аналіз сучасних програмних засобів та сервісів симуляцій	9
РОЗДІЛ 2. Розробка програмного забезпечення для симуляцій ядерних реакцій	Ошибка! Закладка не определена.
2.1. Аналіз програми навчання і постановка цілей для програмного забезпечення.	14
2.2. Методичні рекомендації використання програмного NP на уроках фізики..	20
ВИСНОВКИ	24
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	26
ДОДАТКИ.....	29
Додаток А.....	28
Додаток Б.....	29

ВСТУП

Зміни у світі та суспільстві вимагають розробки нової стратегії організації освітнього процесу у закладах освіти усіх рівнів. Оскільки, одним із пріоритетів сучасної системи освіти є навчання школярів використовувати набуті знання у практичній діяльності та працювати з різними видами інформації (аналіз, синтез, перекодування), то викладання усіх дисциплін вимагає залучення сучасних інформаційних технологій.

Паперові підручники та методичні посібники звичайно нікуди не зникають і все ще використовуються, але загальна «діджиталізація» суспільства впливає на школярів і ефективно викладати новий навчальний матеріал без залучення інтерактивних засобів стає звичним для вчителя.

Фізика як одна із наук природничого циклу має експериментальну складову і її навчання потребує залучення учнів до планування та виконання фізичного експерименту. Проведення дослідів та фізичного експерименту можна здійснювати з використанням різних інтерактивних засобів, у тому числі й симуляцій. Питання використання в освітньому процесі інтерактивних симуляцій знайшло відображення у роботах багатьох науковців, серед яких О. Іваницький, В. Заболотний, О. Кузьменко, О. Головка, Н. Мислицька, Н. Корніжевський та інші. Не применшуючи здобутки науковців у цьому напрямі, вважаємо доцільним продовжити вивчати це питання.

У зв'язку з цим, **метою дослідження** є теоретичне обґрунтування необхідності симуляцій фізичних процесів під час викладання фізики та розробка власного програмного забезпечення для симуляції фізичних процесів (на прикладі розділу «Фізика атома та атомного ядра. Фізичні основи атомної енергетики»).

Досягнення поставленої мети вимагає виконання наступних **завдань**:

- зробити аналіз науково-методичної літератури з проблеми дослідження, з'ясувати зміст основних проблем дослідження («фізична симуляція», «сучасні засоби навчання фізики»), визначити основні шляхи

використання власного програмного забезпечення для занять з розділу «Фізика атома та атомного ядра. Фізичні основи атомної енергетики» у 9 класі;

– розробити програму симуляцій ядерних реакцій;

– узагальнити та проаналізувати результати теоретичного дослідження.

Об’єкт дослідження – освітній процес з фізики у закладах загальної середньої освіти.

Предмет дослідження – методика використання фізичних симуляцій у процесі викладання фізики на прикладі власного програмного забезпечення.

Методи дослідження: теоретичні (аналіз та синтез науково-методичної літератури, інших сервісів симуляцій); емпіричні (збір інформації про використання симуляцій в школах, розробка власного софту).

Практичне значення дослідження полягає у тому, що розроблене програмне забезпечення може використовуватись вчителями для поглиблення знань школярів з розділу «Фізика атома та атомного ядра. Фізичні основи атомної енергетики».

Апробація результатів дослідження. Результати дослідження були обговорені на секційному засіданні Всеукраїнської студентської інтернет-конференції «Інноваційні технології навчання природничо-математичних дисциплін у закладах середньої і вищої освіти» (22-23 квітня 2021 року).

Публікації. За результатами дослідження опубліковані тези доповіді «Методологія розробки програмного засобу Nuclear Power» [29].

РОЗДІЛ 1.

ВИКОРИСТАННЯ СИМУЛЯЦІЙ ФІЗИЧНИХ ПРОЦЕСІВ ЯК НЕВІД'ЄМНА НЕОБХІДНІСТЬ ДЛЯ ПРОЦЕСУ ВИКЛАДАННЯ ФІЗИКИ

1.1. Актуальність використання в освітньому процесі інтерактивних засобів та симуляцій.

В поточній програмі навчання фізики [1], яка затверджена МОН України, багато уваги приділяється інтерактивному навчанню та експериментальній складовій. Ознайомлення школярів із методами та прийомами наукового дослідження сприяє розвитку у них цілеспрямованості та самостійності при плануванні та виконанні експериментальних завдань, а також глибокому засвоєнню нових знань.

Сучасні тенденції в освіті потребують вдосконалення методів вивчення фізики, при цьому особлива увага надається використанню симуляцій фізичних процесів. Це зумовлено тим, що: фізика беззаперечно є експериментальною наукою; система різних видів експериментальної діяльності сприяють не тільки глибокому засвоєнню учнями програмного матеріалу, а і їх особистій реалізації як дослідників; використання симуляцій дозволяє продемонструвати та виконати такі досліди, які не можливо провести в реальних умовах, особливо при викладанні розділу «Ядерна фізика».

Підсумовуючи, можна стверджувати, що проблема використання в освітньому процесі фізичних симуляцій актуальна і перспективна для розвитку. Питання методики на техніки використання фізичних симуляцій та моделювань як невід'ємної частини процесу викладання фізики досліджувалось багатьма вченими та організаціями, серед яких О. Буров, С. Гайдучок, М. Жалдак, Ю. Жук, С. Величко, Ю. Науоменко, О. Слободяник.

Аналіз методичних джерел та онлайн-проектів свідчить, що в освітньому процесі моделювання і симуляція фізичних процесів є:

- ✓ методом їх дослідження, що забезпечує науковість і цілісність шкільного курсу фізики;
- ✓ найефективніший засіб наочного сприйняття природних явищ, є джерелом цікавих для учнів емпіричних фактів;
- ✓ необхідна умова для формування концепту знань учня, що є базою для створення ідей і руху науки в подальшому;
- ✓ засіб візуального сприйняття теоретичних висновків та гіпотез, що забезпечує інтерпретацію теоретичних знань школярів до експериментальної сфери діяльності (зв'язок теорії з практикою);
- ✓ засобом імплементації дослідницьких умінь та навичок учнів.

На думку О. Усової для придання знанням практичної значущості, необхідно навчитися використовувати їх на практиці, як наприклад для розв'язання теоретичних або експериментальних задач, виконання лабораторних робіт [3], тому важливим є мотивація та стимулювання роботи з моделями та симуляціями.

У своїй праці Ю. Єчкало акцентує увагу на тому, що першочергово комп'ютерні моделі слугують для розвитку інтелектуальних здібностей учня та стимулюють його для подальшого навчання, а вже на другому місці є фахове навчання розробляти програмне забезпечення власноруч [4].

В залежності від діяльності учнів роботу з моделюванням фізичних процесів можна поділити на два види: опрацювання та апробація вже «готових» комп'ютерних моделей процесів та явищ, розробка своїх елементарних моделей на основі набутих теоретичних знань. Внаслідок недостатньої бази підготовки учнів і складності розроблення своїх моделей, акцент робиться на вже готове програмне забезпечення, де учню залишається лише вписати вхідні дані для симуляції і отримати результат, який вже опрацювати самостійно.

В багатьох країнах засоби начального моделювання розробляються педагогами, зокрема широко відомі тематичні пакети: Micro Cap, Design Lab, Electronics Workbench для дослідження процесів в електричних колах. Проте,

їх використання можливе за умови покупки ліцензії на їх користування. Вартість ліцензії на використання зазначених програмних продуктів пересічному вчителю сплатити за власний кошт складно, а державне фінансування в даний момент не підтримується.

У зв'язку з цим, актуальними стають безкоштовні готові віртуальні моделі, розроблені педагогами різних країн, з високим рівнем функціоналу та інтерактивності.

Софт для моделювання має різний рівень глибини розробки, деякі мають демонстративний характер, без можливості будь-яких змін, інші надають можливість створювати проекти зі спеціальним розширенням, які можна передавати і відкривати на інших носіях за допомогою розробленої програми.

Моделювання і симуляції хоч і вживаються як синонімічні слова, але предметно відрізняються. Аналіз наукової літератури засвідчив, що:

✓ моделювання (фізика) – це візуалізація спрощеного концепту фізичного явища, який розглядається за допомогою деякого наукового інструмента(в даному випадку – комп'ютера) [2].

✓ симуляція (фізика) – процес розробки моделі реальної або уявної системи і проведення експериментів з нею [2].

З точки зору методики навчання, моделювання фізичних явищ є необхідною при розв'язанні деяких задач, як наприклад:

- візуалізації пояснень учителя. Досвід роботи з учнями показує, що матеріал засвоюється набагато краще при наявності демонстрації, підкріпленої текстовим або голосовим поясненням;

- моделювання цікавих фізичних явищ показує широту можливостей сучасного суспільства і розвитку інформаційних технологій, що підвищує мотивацію учнів до навчання, створює умови до вивчення учнями фізики, щоб мати можливість займатись і розробляти такі ж «круті» моделі;

- моделювання і симуляції дають можливість з високою точністю продемонструвати явища, які в реальному житті відтворити неможливо або дуже дорого, тим самим приєднати учнів до світу великої науки.

Незважаючи на всі переваги комп'ютерного моделювання, необхідно чітко пояснювати учням, що комп'ютерна модель – це не підміна реального фізичного явища, а всього лише спрощення і симуляція реальності за допомогою математичного апарату і розрахунків на комп'ютері з подальшою графічною обробкою чи без неї. Важливість цього моменту підкреслює О. Слободяник в своїй статті [2].

Узагальнюючи вищенаведене можна стверджувати, що використання інтерактивних симуляцій фізичних процесів є актуальним питанням у галузі освіти і потребує подальшого вивчення.

1.2. Аналіз сучасних програмних засобів та сервісів симуляцій.

Незважаючи на те, що інтерактивні програмні засоби є новим та сучасним напрямом, вже існує велика кількість програм і сервісів симуляцій на просторах інтернету.

Один із найкращих сервісів, які рекомендують провідні педагоги є PhET Interactive Simulations від University of Colorado Boulder [5]. Цей сайт створений в 2004 році науковцями з Університету Колорадо (США). На сайті запропоновані безкоштовні моделювання з різних галузей природничих наук, є українська версія, функціонал повністю адаптований до державних стандартів і навчальних програм з фізики.

На сайті розміщені поради та методичні рекомендації щодо використання симуляцій для глибшого пояснення нового програмного матеріалу, індивідуальних та групових позакласних занять тощо. Основною перевагою даного сервісу (у порівнянні з іншими) є можливість впливати і виставляти параметри симуляцій самостійно, що дає учням простір до їх власної творчості і глибшого пізнання фізичних явищ (рис. 1).

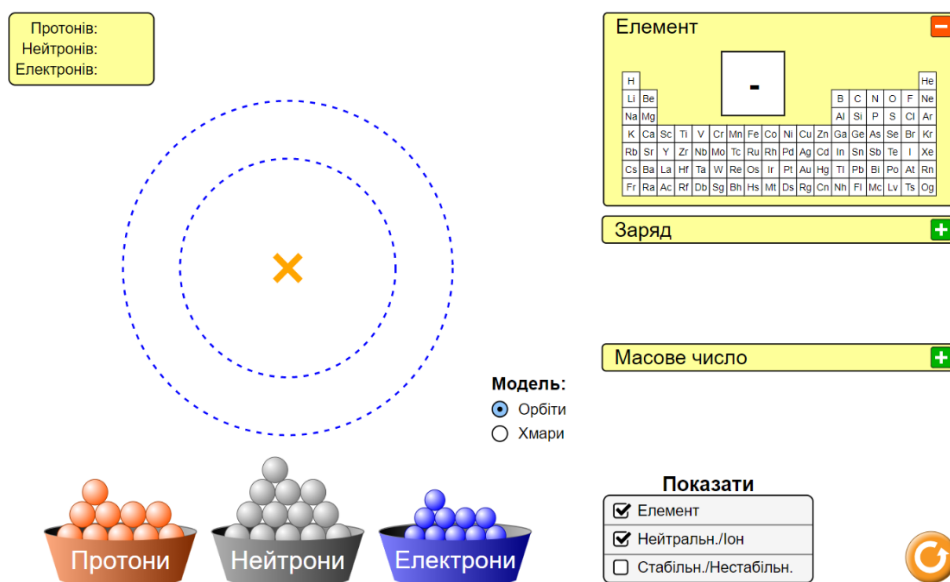


Рисунок 1.1 - Моделювання атома за допомогою сервісу PhET Interactive Simulations.

Розберемо функціонал сервісу на одній із доступних моделей (рис.1). Бачимо, що учень може брати з кошиків елементарні частинки і формувати

атом переміщуючи їх до схеми. Наприклад, додаючи до ядра 5 протонів і 2 нейтрона, а до орбіталей 7 електронів ми отримуємо природну відповідь, що маємо справу в ізотопом бору, який ще іонізований до стану с зарядом -2 (рис.1.2).

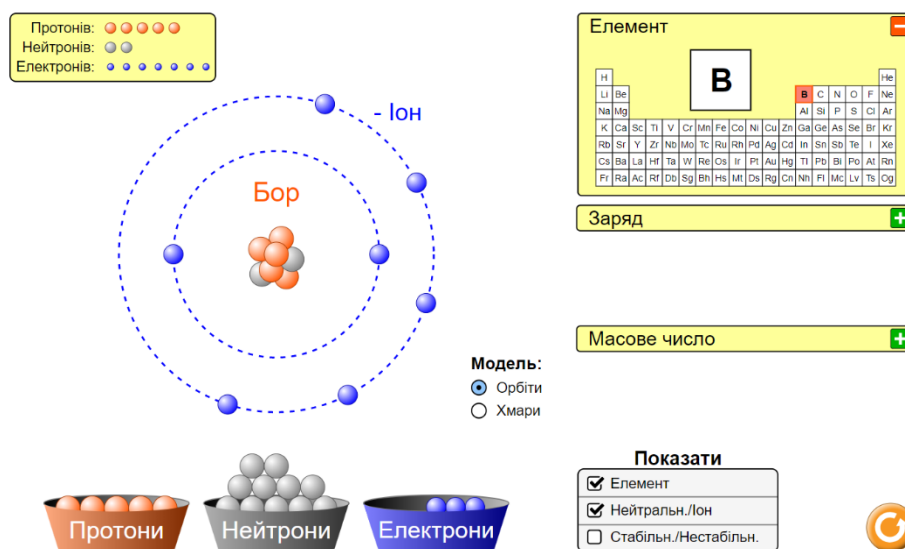


Рисунок 1.2 - Моделювання атома бору за допомогою сервісу PhET Interactive Simulations.

Звичайно, в цієї комп'ютерної моделі є недолік, вона не виявляє неіснуючі ізотопи або іони, але це не є настільки критичним, адже використовувати цю симуляцію слід скоріше для ознайомлення учнів і стимулювання їх до більш сумлінного вивчення теми.

Наприклад, модель явища дифузії (рис.1.3) дає можливість учню спостерігати та провести аналіз руху частинок у мікросвіті за допомогою лише комп'ютера з доступом до мережі Інтернет, а для проведення реального експерименту з вивчення явища дифузії та спостереження за частинками необхідно серед обладнання мати якісний мікроскоп.

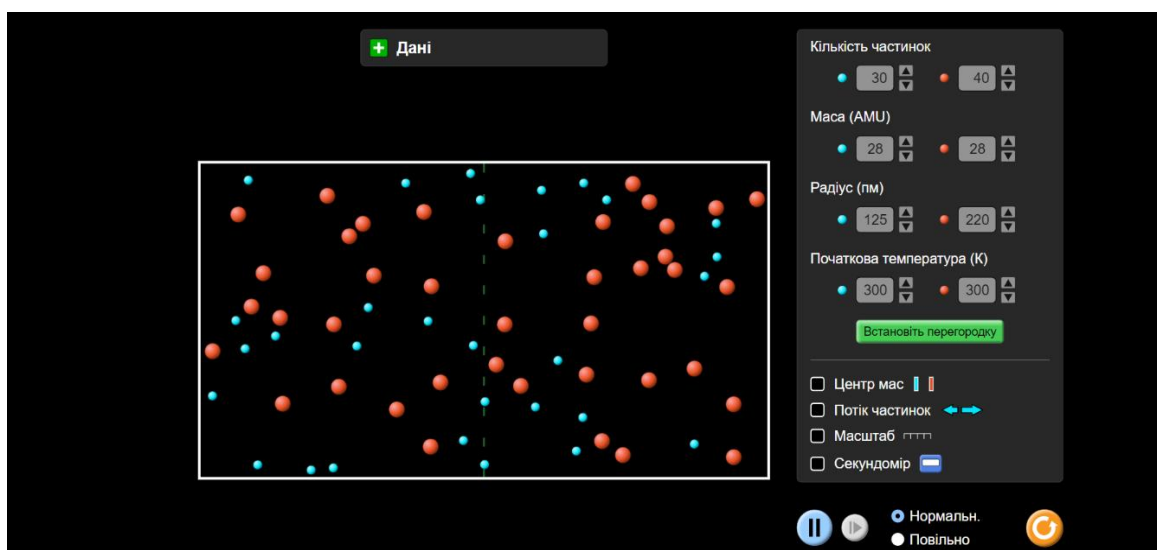


Рисунок 1.3 - Симуляція дифузії за допомогою сервісу PhET Interactive Simulations.

Непоганий сервіс симуляцій для роботи зі школярами розроблений чеськими педагогами [6] (рис.1.4). Безперечною перевагою даного сервісу є україномовна версія сайту, але моделі мають більш казуальну візуалізацію, для учнів старшого шкільного віку симуляції можуть здатися зовсім «дитячими», що може відштовхнути і відбити мотивацію до подальшого розвитку в напрямі фізичної науки.



Рисунок 1.4 - Моделі фізичних процесів на сайті vascak.cz.

Незважаючи на це, за допомогою цих моделей можливе проведення лабораторних робіт за відсутності обладнання або неможливості реального експерименту, адже функціонал сервісу дає можливість введення деяких числових даних і отримання фізично змістовних, правдивих результатів.

З методичної точки зору використання інтерактивних засобів з можливістю взаємодії з учнями є великою перевагою перед звичайними заняттями з підручниками, але зазвичай часу, який відводиться на проведення уроку ледь вистачає для ознайомлення з основами деяких тем, а на пояснення і допомогу школярам в роботі з комп'ютерними моделями часу немає зовсім. У таких випадках бажання розбавити «сухе навчання» цікавою демонстрацією можна втілити за допомогою сервісів з готовими моделями фізичних явищ. Наприклад, таким сервісом є Math, Physics and Engineering Applets [7] (рис.1.5)

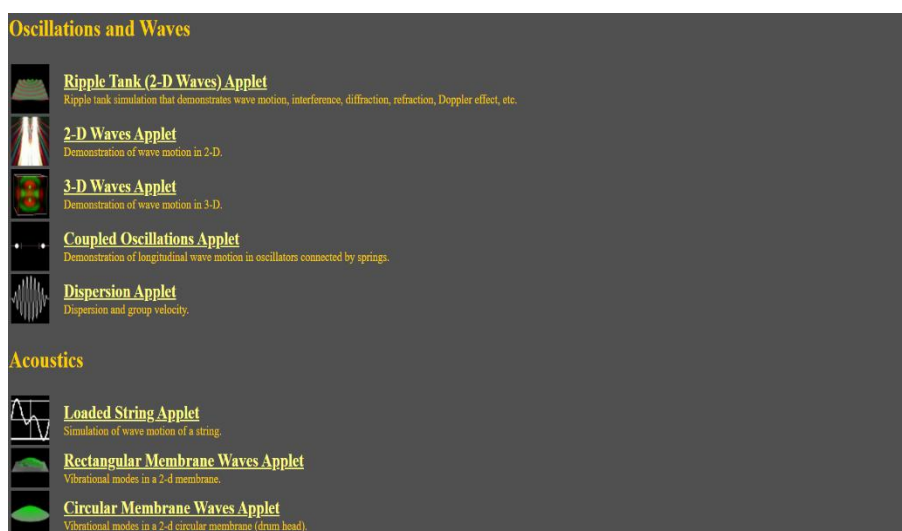


Рисунок 1.5 - Math, Physics and Engineering Applets

Готові демонстрації з цього сайту можна додати до методичного матеріалу заняття, чим покращити сприйняття учнів деяких тем.

Підсумовуючи огляд вже існуючих сервісів симуляцій фізичних процесів можна виділити їх переваги та недоліки. Величезним плюсом є взагалі наявність таких сервісів у безкоштовному доступі, адже розробка якісних і фізично підкріплених симуляцій коштує багато зусиль і часу. Готові інструменти для роботи з симуляціями фізичних явищ безумовно цікаві учням і, в умовах діджиталізації світу, є необхідним інструментом розвитку фізичної шкільної освіти.

РОЗДІЛ 2.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИМУЛЯЦІЇ ЯДЕРНИХ РЕАКЦІЙ

2.1. Аналіз навчальної програми з фізики і постановка цілей для програмного забезпечення.

Аналіз навчальної програми з фізики засвідчив, що на вивчення розділу «Фізика атомна та атомного ядра. Основи атомної енергетики» у 9 класі виділяється всього 12 годин, незважаючи на всю її важливість і актуальність в сучасному світі. При цьому спектр питань, які повинен висвітлити вчитель широкий: від сучасної моделі атома, з висвітленням історії розвитку фізики атома, до екологічних проблем атомної енергетики та термоядерних реакцій. При цьому вчитель повинен провести на уроках 4 основні демонстрації: 1. Модель досліду Резерфорда; 2. Модель атома. Модель ядра атома; 3. Принцип дії лічильника іонізації; 4. Дозиметри (за наявності).

Нажаль, при відсутності фінансування та моральної застарілості наявного у шкільних кабінетах фізики обладнання (інколи й відсутності обладнання), вчитель не може провести демонстрації зазначені у навчальній програмі на належному рівні. У цьому випадку доцільним є використання інтерактивних засобів та симуляцій фізичних процесів.

Проведений аналіз програмних продуктів, які є у вільному доступі, засвідчив наявні у них переваги та недоліки. Це стимулювало до розробки власного інтерактивного продукту, яке може бути використане вчителем при викладанні розділу «Фізика атомна та атомного ядра. Основи атомної енергетики».

Метою розробки власного програмного забезпечення для симуляції ядерних реакцій стала відсутність безкоштовного, зрозумілого для школярів софту, який може використовувати кожний. Безумовно, в інтернеті маємо декілька моделей, наприклад із вказаних у попередньому розділі сервісах, але

вони виглядають по-дитячому, що в основному відштовхує учнів від користування ними. Учням старшого шкільного віку хочеться почувати себе дорослими, тому і оформлення, адаптація програми повинна бути в науковому стилі, але не дуже складною для освоєння вчителями та учнями.

Перед початком роботи, було розроблено наступний план дій:

- за допомогою програми МОН виділити основні теми, до яких повинні бути представлені теми в симуляції;
- розробити віконний додаток;
- розробити парсер даних для отримання інформації з інтернету про фізичні властивості елементів і переріз різних ядерних реакцій тощо;
- розробити мінімалістичний стильний дизайн;
- створити елементарні графічні анімації для візуалізації ядерних реакцій;
- поєднати все в готову програму з доступним інтерфейсом для користувача.

Основним джерелом чисельних даних було вибрано сайт NIST (Nuclear Institute of Standards and Technology).

Очікуваний функціонал:

- детальна кількісна інформація про будь-який хімічний елемент за запитом;
- усі базові та відомі реакції розпаду, синтезу;
- деякі дерева розпаду для елементів уранової групи;
- яскраві та зрозумілі анімації ядерних реакцій;
- можливість розрахунку енергетичного виходу реакцій.

Першочергово відбулась розробка парсера, для швидкого сканування інформації з вказаних сайтів. Після обробки парсером сайту, вся інформація була об'єднана в локальні бази даних, для того, щоб зробити можливим користування в режимі поза мережею. Далі була розроблена розрахункова програма (Додаток Б), для того, щоб пов'язати дані з бази з користувачем, дати можливість за допомогою зрозумілого інтерфейсу отримувати інформацію,

розраховувати деякі числові величини (дефект мас, вихід реакції для різних вхідних даних).

Перевести сухий код на мові програмування Python в віконний додаток на базі операційної системи Windows вдалось за рахунок кросплатформової графічної події, орієнтованої графічної бібліотеки на основі Tk (розповсюджена GNU). Як результат отримали програму, яку було названо NP (Nuclear Power). Головне меню зображено на рис. 2.1.



Рисунок 2.1 – Інтерфейс авторської програми Nuclear Power.

Розглянемо кожен із розділів функціоналу детальніше.

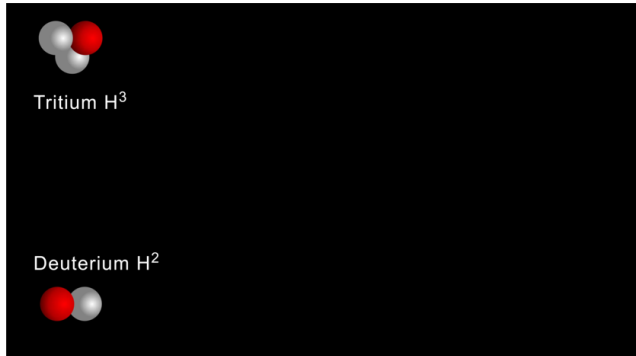
Базові ядерні реакції. Назва цього розділу говорить сама за себе, в ньому учень може ознайомитись з базовими ядерними реакціями, а також з існуючими реакціями для кожного елементу періодичної системи. Після переходу до цього розділу, учню доступний функціонал який складається з наступних пунктів:

- Альфа-розпад;
- Бета-розпад;
- Гамма-розпад;
- Термоядерний синтез;

- Ланцюгова ядерна реакція;
- Реакції для окремих елементів (рис.2.2).

NP

Виберіть реакцію, яка Вас зацікавила.



Tritium H^3

Deuterium H^2

Основні термоядерні реакції:

$$H_1^3 + H_1^2 = He_2^3(0.82 MeV) + n(2.45 MeV)$$

$$H_1^3 + H_1^3 = He_4^2(3.5 MeV) + n(14.1 MeV)$$

$$H_1^2 + He_2^3 = He_4^2(3.6 MeV) + p(14.7 MeV)$$

Функціонал

Альфа-розпад

Бета-розпад

Гамма-розпад

Термоядерний синтез

Ланцюгова ядерна реакція

Реакції для окремих елементів

Повернутися до головного меню

Рисунок 2.2 – Розділ «Базові ядерні реакції».

NP

Виберіть реакцію, яка Вас зацікавила.

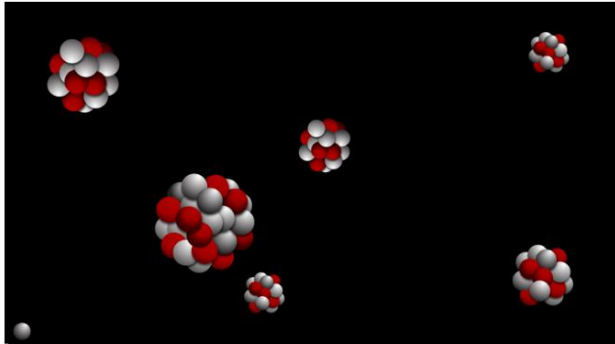


Схема контрольованої ланцюгової реакції

Функціонал

Альфа-розпад

Бета-розпад

Гамма-розпад

Термоядерний синтез

Ланцюгова ядерна реакція

Реакції для окремих елементів

Повернутися до головного меню

Рисунок 2.3 – Підрозділ «Ланцюгова ядерна реакція».

На рис 2.2 зображено підрозділ «Термоядерний синтез», використавши який школяр може подивитися симуляцію ядерної реакції вмонтовану в програму у вигляді відеопрогравача, а також ознайомитися з основними термоядерними реакціями та їх енергетичними виходами.

В розділі «Розрахунок характеристик реакцій» учень має можливість ввести до спеціального вікна реакцію, знайти її енергетичний вихід, дізнатися її ймовірність відносно інших реакцій для заданого хімічного елемента.

Одним із найцікавіших є розділ «Інформація про елементи періодичної системи» (рис.2.4). В цьому розділі у спеціальному вікні вводиться назва хімічного елемента українською мовою і отримується досить детальна інформація про конкретний хімічний елемент. Це дуже зручно використовувати в умовах уроку, оскільки не потребує часу для пошуку в інтернеті, основна інформація про хімічні елементи об'єднана у межах однієї програми.

Розділ дерева розпаду – це інформація для допитливих, яка виходить за межі шкільної програми. Учень має можливість подивитися еволюцію нестабільного хімічного елемента до стабільних, оцінити ймовірність перетворення в той чи інший елемент та розрахувати приблизний час еволюції для ядра хімічного елемента.

The screenshot shows a window titled "NP" with the following content:

- Гелій** (Helium)
- atomic number: 2, atomic weight: 4.003
- symbol: He
- electron configuration: 1s²
- name: helium
- crystal structure: Hexagonal
- physical state at 20 °C (68 °F): Gas
- Classification: Noble gases

Below the data is a descriptive paragraph in Ukrainian:

Гелій - хімічний елемент з атомним номером 2, який належить до 18-ї групи (за старою класифікацією — головної підгрупи 8-ї групи), 1-го періоду періодичної системи хімічних елементів, та є першим представником благородних газів. Також, гелій (He) — проста речовина, яку утворює хімічний елемент гелій — (за нормальних умов) дуже легкий, хімічноінертний (найінертніший) одноатомний газ без запаху, кольору та смаку. Також гелій є другою за легкістю (після водню) хімічною речовиною. Його температура кипіння — найнижча серед усіх відомих речовин.

On the right side of the window, there is a red instruction: "Введіть у вікно назву хімічного елемента" (Enter the name of the chemical element in the window). Below it is a text input field containing "Гелій". At the bottom right, there is an orange button labeled "Повернутися до головного меню" (Return to the main menu).

Рисунок 2.4 – Розділ «Інформація про елементи періодичної системи».

Узагальнюючи вищенаведене, можна стверджувати, що розроблений власний продукт NP має інтуїтивний та зручний інтерфейс, а також простий у користуванні. Методичні рекомендації використання програмного засобу наведені у п.п. 2.2.

2.2. Методичні рекомендації використання програмного NP на уроках фізики.

Аналіз навчальної програми та підручників з фізики засвідчив складність для розуміння учнями теоретичного матеріалу розділу «Фізика атома та атомного ядра. Фізичні основи атомної енергетики». Розроблене програмне забезпечення покликане осучаснити виклад навчального матеріалу, надати можливість учням «побачити» атом та його ядро зсередини. У зв'язку з цим, у межах дослідження нами був розроблений конспект уроку з теми «Ядерні реакції» з використанням розробленого програмного продукту NP. За основу були обрані та адаптовані взяли два конспекти уроку з порталу «На Урок» [9, 10].

Тема: Ядерні реакції.

Мета: вивчити поняття про ядерні реакції, розглянути їх практичне використання; за допомогою програми NP розглянути основні типи реакцій, елементи, які приймають в них участь; навчитись розв'язувати задачі, розвинути уважність, спостережливість, допитливість.

Обладнання: шкільний підручник, програма «Nuclear Power», збірник задач.

Хід заняття

1. Організаційний етап.

2. Опитування з попередньої теми: «Радіоактивність. Радіаційне випромінювання»

✓ Що таке ядро? Який його склад?

- ✓ Де можемо знайти інформацію про атомні ядра?
- ✓ Які числа використовуємо для характеристики ядер?
- ✓ Як визначити кількість протонів? Нейтронів?
- ✓ Що таке енергія зв'язку?
- ✓ Що таке ядерні сили?

3. Етап вивчення нового матеріалу.

Ядерні реакції – це процес перетворення атомних ядер внаслідок їх взаємодії з елементарними частинками або між собою. Ядерні реакції відбуваються переважно при бомбардуванні ядер речовини потоком прискорених частинок.

Ядерні реакції записують за загальною схемою:

$A+a \rightarrow V+b$, або $A(a,b)V$, де A – початкове ядро, a – частинка, яка ініціює ядерну реакцію, V – нове ядро, b – частинка, що виникає при реакції

Під час ядерної реакції безумовно виконуються закон збереження імпульсу, моменту імпульсу та енергії, електричного заряду, нуклонного числа.

Енергетичним виходом ядерної реакції називають різницю між енергією спокою частинок до реакції і після реакції.

Залежно від енергетичного виходу реакцію можна поділити на ендотермічну – коли енергію поглинається та екзотермічну – енергія виділяється.

4. Приклади реакцій (одночасно запускається програма NP).

- Альфа-розпад
- Бета – розпад
- Гамма-розпад
- Термоядерний синтез
- Ланцюгова реакція
- Цікаві реакції для окремих елементів

Під час пояснень застосовуємо симуляції за допомогою програми NP (рис.2.5, 2.6).

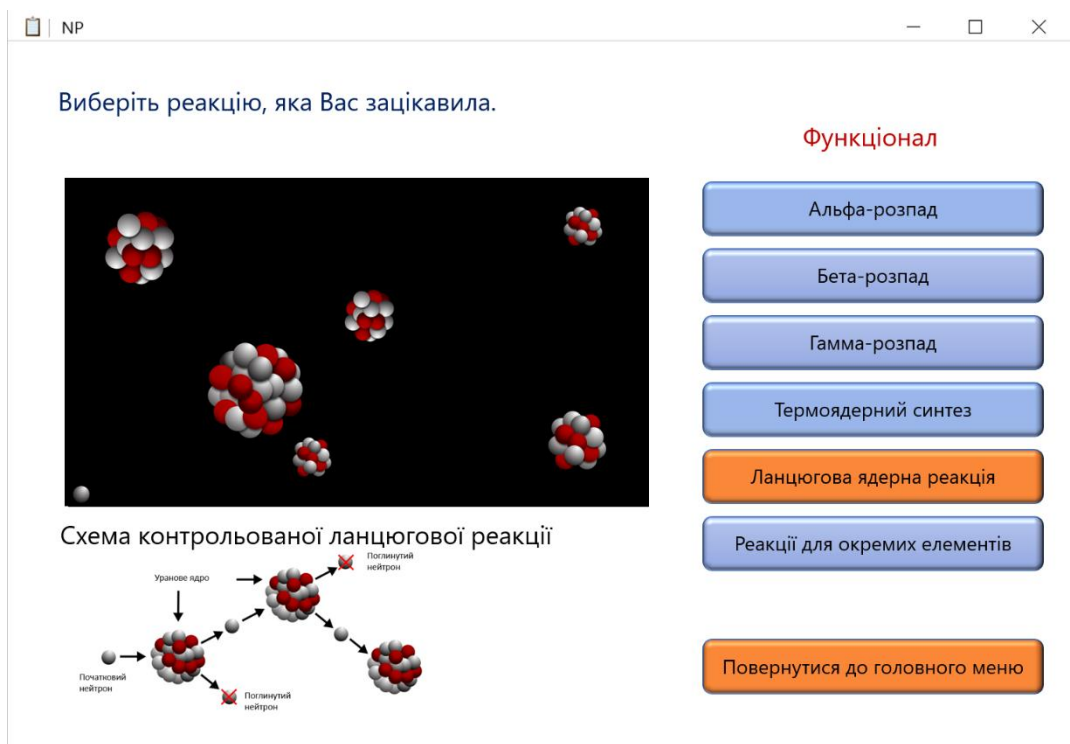


Рисунок 2.5 - Ланцюгова ядерна реакція.

Рисунок 2.5 - Реакції термоядерного синтезу.

5. Закріплення знань учнів:

1. Якщо енергія спокою частинок більше до реакції, ніж після, якого типу ця реакція? (Екзотермічна).

2. Задача. Використавши реакцію ${}^{17}_7\text{N} + {}^4_2\text{He} \rightarrow {}^{17}_8\text{O} + {}^1_1\text{H}$: Обчислити енергетичний вихід цієї реакції, з'ясувати є реакція екзотермічною або ендотермічною? Перевірити результат за допомогою програми «Nuclear Power».

6. Підбиваємо підсумки заняття, оголошення домашнього завдання.

Нажаль, перевірити ефективність розробленого програмного забезпечення в умовах освітнього процесу не вдалось у цьому навчальному році. Проте, нами була організована бесіда з учнями 9 класу (у кількості 6 осіб), у ході якої їм був представлений програмний засіб «Nuclear Power (NP)». Під час роботи із програмою, учні швидко орієнтувалися, проявили зацікавленість до вивчення розділу «Фізика атома та атомного ядра. Фізичні основи атомної енергетики».

Використання програми «Nuclear Power (NP)» рекомендується під час вивчення теми «Фізика атома та атомного ядра. Фізичні основи атомної енергетики» як допоміжний інтерактивний матеріал. Функціонал NP дозволяє отримувати знання щодо процесів ядерних реакцій, дізнаватися коротку інформацію про різні хімічні елементи, розраховувати основні параметри ядерних реакцій, дивитися цікаві симуляції. Мінімалістичний зрозумілий дизайн допомагає швидко адаптуватися учням і користуватися програмою самостійно як електронним посібником.

Для покращення засвоювання теми, рекомендується розповсюджувати її серед учнів, давати домашні завдання на ознайомлення с різними хімічними елементами, пропонувати отримати кількісні характеристики різних ядерних реакцій.

ВИСНОВКИ

1. Використання моделей і симуляцій є невід'ємною, а в умовах загального технологічного буму, стає обов'язковою частиною освітнього процесу з фізики. Під час шкільних занять вчитель намагається зацікавити учнів, осучаснити освітній процес, поглибити знання учнів шляхом залучення їх до виконання фізичних демонстрацій та дослідів. Але матеріальне обладнання шкільних кабінетів не завжди дає можливість це зробити. Проте, сьогодні існує багато сервісів і програм для моделювання різних фізичних явищ, які легко можуть замінити реальні демонстрації, дати учням простір для роздумів, а ще й можливість побути учасниками і творцями віртуального експерименту. Використання віртуальних симуляцій фізичних процесів є доцільним не тільки при поясненні вчителем нового навчального матеріалу, а й під час розв'язування фізичних задач, виконання демонстрацій та фізичного експерименту, виконання учнями домашніх досліджень.

2. Аналіз навчальної програми з фізики та існуючих сервісів засвідчили відсутність доступних програм симуляцій з деяких тем та методичних рекомендацій щодо їх використання у процесі навчання фізики, зокрема при вивченні структури атома та атомного ядра.

З метою створення актуального і конкурентоспроможного програмного забезпечення, були розглянуті та проаналізовані переваги та недоліки вже існуючих проектів інших педагогів, організацій, інтернет-сервісів. Основним критерієм вважалась привабливість сервісу симуляцій для учнів та широкий функціонал, який мав забезпечити не тільки демонстраційний сегмент, а й можливість отримання числових результатів, цікавої і точної актуальної інформації.

Результатом роботи стала авторська програма з назвою «Nuclear Power» (скорочено NP). Використання програмного засобу дозволяє: побачити модель та симуляції базових ядерних реакцій; розраховувати та моделювати основні параметри ядерних реакцій; отримати повну та вичерпну інформацію про

хімічні елементи; спостерігати та кількісно оцінювати еволюцію нестабільних ядер за допомогою дерев розпаду.

У ході дослідження розроблені методичні рекомендації використання програми «Nuclear Power», зокрема розроблено конспект уроку з теми «Ядерні реакції».

Узагальнюючи отримані результати можна стверджувати, що розроблена програма симуляцій може бути використана вчителями у професійній діяльності під час викладання розділу «Фізика атома та атомного ядра. Фізичні основи атомної енергетики», шкільних гуртках, тематичних виставках. Перспективою подальших досліджень є удосконалення розробленого програмного продукту та впровадження в освітній процес закладу загальної середньої освіти.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Навчальна програма з фізики 7-9 клас. URL: <https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/navchalni-programi/navchalni-programi-5-9-klas> (дата звернення 11.01.2021)
2. Слободяник О.В. Комп'ютерні моделі у дослідницькій діяльності учнів з фізики. *Фізико-математична освіта*. Суми, 2018. Випуск 4(18). С. 149-153.
3. Усова А.В., Вологодская З.А. Самостоятельная работа учащихся по физике в средней школе. М.: Просвещение, 1981, 181 с.
4. Єчкало Ю.В. Засоби навчання комп'ютерного моделювання в курсі фізики. *Вісник Дніпропетровського університету імені Альфреда Нобеля. Серія: Педагогіка і психологія*. Дніпро, 2015. № 1, С. 7–12.
5. Phet. URL: <https://phet.colorado.edu/uk/> (дата звернення 11.01.2021)
6. Фізика в школі. URL: vascak.cz (дата звернення 05.01.2021)
7. Educational applets. URL: <https://www.falstad.com/mathphysics.html> (дата звернення 11.02.2021)
8. Physical Measurement Laboratory. URL: <https://www.nist.gov/pml/> (дата звернення 12.02.2021)
9. Освітній портал «На урок». Конспект уроку «Ядерні реакції». URL: <https://naurok.com.ua/urok-yaderni-reakci-177396.html> (дата звернення 11.01.2021)
10. Освітній портал «На урок». Конспект уроку «Ядерні реакції». URL: <https://naurok.com.ua/urok-yaderni-reakci-132697.html> (дата звернення 11.01.2021)
11. Калапуша Л.Р. Моделювання у вивченні фізики. К.: Рад. шк., 1982. С. 43-78.
12. Жалдак М.І. Проблеми інформатизації навчального процесу в середніх і вищих навчальних закладах. *Комп'ютер у школі та сім'ї*. Київ, 2013. № 3. С. 8-15.
13. Мадзігон В.М., Лапінський В.В. Сучасне навчальне середовище і електронна педагогіка. *Комп'ютер у школі та сім'ї*. Київ, 2010. № 3. С. 3-6.

14.Шут М. І., Сергієнко В.П. Теоретико-методичні особливості використання сучасних комп'ютерно орієнтованих засобів навчання загальної фізики. URL: <https://journal.iitta.gov.ua/index.php/itlt/article/view/298> (дата звернення 11.01.2021).

15.Слободяник О.В. Домашні експериментальні завдання як засіб активізації самостійної пізнавальної діяльності студентів. *Наукові записки. Серія: Проблеми методики фізико-математичної і технологічної освіти*. Кіровоград, 2011. № 1. С.108-113.

16.. Морзе Н. В. Підготовка педагогічних кадрів до використання комп'ютерних телекомунікацій. *Комп'ютерно-орієнтовані системи навчання*. Київ, 2003. Вип. 6.

17.Василинчук А. В. Поєднання фізичного й віртуального експерименту під час вивчення дифракції світла. *Фізика та астрономія в школі*. Київ, 2005. №5. С. 36-39.

18.Данилюк Р. Використання комп'ютерних моделей у шкільному курсі фізики. *Фізика*. Київ, 2004. №30. С. 1-2

19.Волинський В.П., Головка Н.І. Використання відеоінформації як засобу навчання. *Педагогіка і психологія*. Київ, 1995. №3. С. 71-76.

20..Матрос Д.Ш. Внедрение информационных и коммуникационных технологий в школу. *Информатика и образование*. Москва, 2000, №8. С. 9-10.

21.Національна доктрина розвитку освіти України у XXI ст. URL: <https://zakon.rada.gov.ua/laws/show/347/2002#Text> (дата звернення 21.01.2021).

22.Галузяк В.М., Сметанський М.І., Шахов В.І. Педагогіка: навчальний посібник. Вінниця: РВВ ВАТ «Вінницька обласна друкарня», 2001. 206 с.

23.Сільвейстр А. Актуалізація пізнавальної діяльності учнів на уроках з застосуванням НІТН. URL: <http://enpuir.npu.edu.ua/bitstream/123456789/15171/1/Silveystr.pdf> (дата звернення 21.02.2021).

24.Чашко Л.Б. Аудіовізуальна навчальна інформація на уроках. *Педагогіка і психологія*. Київ, 1999. №1. С. 51-57ю

25.Дорошенко Ю., Лапінський В., Мальований Ю. Педагогічні програмні засоби: сучасний стан і можливості. *Гуцульська школа.* – 2000. №. 1-2. С. 6-10.

26.Желюк О.М. Комп'ютерна техніка в навчальному курсі фізики: метод.рекомендації. Рівне: РДПУ, 1994. 112 с.

27.. Желюк О. Засоби НІТ у навчальному фізичному експерименті. Фізика. Київ, 2001. №9.

28.Мельник Ю.С. Комп'ютерне моделювання в процесі розв'язування фізичних задач. *Комп'ютер у школі та сім'ї.* Київ, 2015. №7. С. 18-25.

29.Югай М.Е. Методологія розробки програмного засобу Nuclear Power. *Актуальні проблеми природничо-математичних дисциплін у закладах освіти:* зб. тез Всеукраїнської студентської інтернет-конф., м. Херсон, 22-23 квітня. 2021 р. Херсон, 2020. С. 53-55.

ДОДАТКИ

Додаток А

КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНЬСЬКОГО
ДЕРЖАВНОГО УНІВЕРСИТЕТУ

Я, Югай Марк Едуардович, учасник(ця) освітнього процесу Херсонського державного університету, **УСВІДОМЛЮЮ**, що академічна доброчесність – це фундаментальна етична цінність усієї академічної спільноти світу.

ЗАЯВЛЯЮ, що у своїй освітній і науковій діяльності **ЗОБОВ'ЯЗУЮСЯ**:

- дотримуватися:
 - вимог законодавства України та внутрішніх нормативних документів університету, зокрема Статуту Університету;
 - принципів та правил академічної доброчесності;
 - нульової толерантності до академічного плагіату;
 - моральних норм та правил етичної поведінки;
 - толерантного ставлення до інших;
 - дотримуватися високого рівня культури спілкування;
- надавати згоду на:
 - безпосередню перевірку курсових, кваліфікаційних робіт тощо на ознаки наявності академічного плагіату за допомогою спеціалізованих програмних продуктів;
 - оброблення, збереження й розміщення кваліфікаційних робіт у відкритому доступі в інституційному репозитарії;
 - використання робіт для перевірки на ознаки наявності академічного плагіату в інших роботах виключно з метою виявлення можливих ознак академічного плагіату;
- самостійно виконувати навчальні завдання, завдання поточного й підсумкового контролю результатів навчання;
 - надавати достовірну інформацію щодо результатів власної навчальної (наукової, творчої) діяльності, використаних методик досліджень та джерел інформації;
 - не використовувати результати досліджень інших авторів без використання покликань на їхню роботу;
 - своєю діяльністю сприяти збереженню та примноженню традицій університету, формуванню його позитивного іміджу;
 - не чинити правопорушень і не сприяти їхньому скоєнню іншими особами;
 - підтримувати атмосферу довіри, взаємної відповідальності та співпраці в освітньому середовищі;
 - поважати честь, гідність та особисту недоторканність особи, незважаючи на її стать, вік, матеріальний стан, соціальне становище, расову належність, релігійні й політичні переконання;
 - не дискримінувати людей на підставі академічного статусу, а також за національною, расовою, статевою чи іншою належністю;
 - відповідально ставитися до своїх обов'язків, вчасно та сумлінно виконувати необхідні навчальні та науково-дослідницькі завдання;
 - запобігати виникненню у своїй діяльності конфлікту інтересів, зокрема не використовувати службових і родинних зв'язків з метою отримання нечесної переваги в навчальній, науковій і трудовій діяльності;
 - не брати участі в будь-якій діяльності, пов'язаній із обманом, нечесністю, списуванням, фабрикацією;
 - не підроблювати документи;
 - не поширювати неправдиву та компрометуючу інформацію про інших здобувачів вищої освіти, викладачів і співробітників;
 - не отримувати і не пропонувати винагород за несправедливе отримання будь-яких переваг або здійснення впливу на зміну отриманої академічної оцінки;
 - не залучувати й не проявляти агресії та насильства проти інших, сексуальні домагання;
 - не завдавати шкоди матеріальним цінностям, матеріально-технічній базі університету та особистій власності інших студентів та/або працівників;
 - не використовувати без дозволу ректорату (деканату) символіки університету в заходах, не пов'язаних з діяльністю університету;
 - не здійснювати і не заохочувати будь-яких спроб, спрямованих на те, щоб за допомогою нечесних і негідних методів досягти власних корисних цілей;
 - не завдавати загрози власному здоров'ю або безпеці іншим студентам та/або працівникам.

УСВІДОМЛЮЮ, що відповідно до чинного законодавства у разі недотримання Кодексу академічної доброчесності буду нести академічну та/або інші види відповідальності й до мене можуть бути застосовані заходи дисциплінарного характеру за порушення принципів академічної доброчесності.

05.04.2021
(дата)


(підпис)

Югай М.Е.
(ім'я, прізвище)

Додаток Б

```

from future import print_function
import sys
import os

sys.path.insert(0, './Scripts/')
import re
import subprocess
import shutil
import math
import parseENSDF as ensdf
import getNaturalIsotopes as gni
import getAbundance as isoabund

class constants:
    N_A = 6.0221409e+23
    MeV_to_keV = 1.e3
    mb_to_cm2 = 1.e-27
    year_to_s = 31536000
    min_bin = 0 # keV
    max_bin = 20000 # keV
    delta_bin = 100 # keV
    run_talys = False
    run_alphas = True
    print_alphas = False
    download_data = False
    force_recalculation = False
    ofile = sys.stdout

class material:
    def __init__(self, e, a, f):
        self.ele = str(e)
        self.A = float(a)
        self.frac = float(f)

def isoDir(ele, A):
    return './Data/Isotopes/' + ele.capitalize() + '/' + ele.capitalize() +
str(int(A)) + '/'

def parseIsotope(iso):
    ele = ''
    A = 0
    for i in iso:
        if i.isalpha():
            ele += i
        if i.isdigit():
            A = A * 10 + int(i)
    return [ele, A]

def generateAlphaFileName(ele, A):
    outdir = './AlphaLists/'
    fName = outdir + ele.capitalize() + str(A) + 'Alphas.dat'
    return fName

def generateAlphaList(ele, A):
    print('generateAlphaList(', ele, A, ')', file=constants.ofile)
    ensdf.main(['parseENSDF', ele, A])

```

```

def loadAlphaList(fname):
    f = open(fname)
    tokens = map(lambda line: line.split(), f.readlines())
    alpha_list = []
    for words in tokens:
        if words[0][0] == '#' or len(words) < 2:
            continue

        alpha = []
        for word in words:
            alpha.append(float(word))
        alpha_list.append(alpha)
    f.close()
    return alpha_list

def getAlphaList(ele, A):
    fname = generateAlphaFileName(ele, A)
    return loadAlphaList(fname)

def getAlphaListIfExists(ele, A):
    fName = generateAlphaFileName(ele, A)
    tries = 3
    while not os.path.isfile(fName):
        if tries < 0:
            print('Cannot generate alpha list for ele = ', ele, ' and A = ',
A, file=constants.ofile)
            return 0
        print('generating alpha file ', fName, file=constants.ofile)
        generateAlphaList(ele, A)
        tries -= 1
    return getAlphaList(ele, A)

def loadChainAlphaList(fname):
    f = open(fname)
    tokens = map(lambda line: line.split(), f.readlines())
    alpha_list = []
    for line in tokens:
        if len(line) < 2 or line[0][0] == '#':
            continue
        iso = line[0]
        br = float(line[1])
        [ele, A] = parseisotope(iso)

        aList_forIso = getAlphaListIfExists(ele, A)
        if constants.print_alphas:
            print(iso, file=constants.ofile)
            print('\t', aList_forIso, file=constants.ofile)
        for [ene, intensity] in aList_forIso:
            alpha_list.append([ene, intensity * br / 100])
    return alpha_list

def readTargetMaterial(fname):
    f = open(fname)
    mat_comp = []
    tokens = map(lambda line: line.split(), f.readlines())
    for line in tokens:
        if len(line) < 3:
            continue
        if line[0][0] == '#':

```

```

        continue
    ele = line[0].lower()
    A = int(line[1])
    frac = float(line[2])

    if A == 0:
        natIso_list = gni.findIsotopes(ele).split()
        for A_i in natIso_list:
            abund = float(isoabund.findAbundance(str(A_i) +
ele.capitalize()))
            mater = material(ele, A_i, frac * abund / 100.)
            mat_comp.append(mater)
    else:
        mater = material(ele, A, frac)
        mat_comp.append(mater)

# Нормальний розмір
norm = 0
for mat in mat_comp:
    norm += mat.frac
for mat in mat_comp:
    mat.frac /= norm
return mat_comp

def calcStoppingPower(e_alpha_MeV, mat_comp):
    # Зупиняюча сила для вказаних енергій
    e_alpha = e_alpha_MeV
    sp_total = 0
    mat_comp_reduced = {}
    for mat in mat_comp:
        if mat.ele in mat_comp_reduced:
            mat_comp_reduced[mat.ele] += mat.frac
        else:
            mat_comp_reduced[mat.ele] = mat.frac

    for mat in mat_comp_reduced:
        spDir = './Data/StoppingPowers/'
        spFile = spDir + mat.lower() + '.dat'
        spf = open(spFile)

        tokens = map(lambda line: line.split(), spf.readlines())
        first = True
        sp_found = False
        e_curr = 0
        e_last = 0
        sp_curr = 0
        sp_last = 0
        sp_alpha = 0
        for line in tokens:
            if line[0][0] == '#':
                continue
            e_curr = float(line[0])
            if str(line[1]) == 'keV':
                e_curr /= 1000
            elif str(line[1]) == 'MeV':
                e_curr *= 1
            sp_curr = float(line[3]) + float(line[2])

        # Енергія альфа-частинок
        if e_curr > e_alpha and first:
            first = False
            sp_found = True
            sp_alpha = sp_curr

```

```

        break
    if e_curr > e_alpha:
        first = False
        sp_alpha = (sp_curr - sp_last) * (e_alpha - e_last) / (e_curr
- e_last) + sp_last
        sp_found = True
        break
    first = False
    sp_last = sp_curr
    e_last = e_curr
    if not sp_found:
        sp_alpha = sp_last
    sp_total += sp_alpha * mat_comp_reduced[mat] / 100
    return sp_total

def runTALYS(e_a, ele, A):
    iso = str(ele) + str(int(A))
    inpdire = isoDir(ele, A) + 'TalysInputs/'
    outdire = isoDir(ele, A) + 'TalysOut/'
    nspecre = isoDir(ele, A) + 'NSpectra/'
    if not os.path.exists(inpdire):
        os.makedirs(inpdire)
    if not os.path.exists(outdire):
        os.makedirs(outdire)
    if not os.path.exists(nspecre):
        os.makedirs(nspecre)

    command = "\nprojectile a\njectiles p n g\nelement " + ele + "\nmass " +
str(int(A)) + "\nenergy " + str(
        e_a) + "\npreequilibrium y\nngiantresonance y\nmultipreeq
y\noutspectra y\noutlevels y\noutgamdis y\nfilespectrum n\nelwidth 0.2\n"

    inp_fname = inpdire + "inputE" + str(e_a)
    inp_f = open(inp_fname, 'w')
    inp_f.write(command)
    inp_f.close()
    out_fname = outdire + "outputE" + str(e_a)

    bashcmd = 'talys < ' + inp_fname + ' > ' + out_fname
    print('Running TALYS:\t ', bashcmd, file=constants.ofile)
    runscript_fname = "./runscript_temp.sh"
    runscript_f = open(runscript_fname, "w")
    runscript_f.write("#!/usr/bin/bash\n\n" + bashcmd)
    runscript_f.close()
    process = subprocess.call(bashcmd, shell=True)
    ls = os.listdir("./")
    moved_file = False
    for f in ls:
        if "nspec" in f:
            if os.path.exists(nspecre + f):
                os.remove(nspecre + f)
                shutil.move(f, nspecre)
                moved_file = True
    if not moved_file:
        fname = nspecre + 'nspec{0:0>7.3f}.tot'.format(e_a)
        blank_f = open(fname, 'w')
        blank_f.write("EMPTY")
        blank_f.close()

def getMatTerm(mat, mat_comp):
    A = mat.A
    conc = mat.frac / 100.

```

```

mat_term = (constants.N_A * conc) / A
return mat_term

def getIsotopeDifferentialNSpec(e_a, ele, A):
    target = ele + str(int(A))
    path = isoDir(ele, A) + 'NSpectra/'
    if not os.path.exists(path):
        os.makedirs(path)
    fname = path + 'nspec{0:0>7.3f}.tot'.format(int(100 * e_a) / 100.)
    outpath = isoDir(ele, A) + 'TalysOut'
    if constants.force_recalculation:
        print('Forcibly running TALYS for', int(100 * e_a) / 100., 'alpha
on', target, file=constants.ofile)
        print('Outpath', outpath, file=constants.ofile)
        runTALYS(int(100 * e_a) / 100., ele, A)

    if not os.path.exists(fname):
        if constants.run_talys:
            while not os.path.exists(fname):
                print('Running TALYS for', int(100 * e_a) / 100., 'alpha on',
target, file=constants.ofile)
                print('Outpath', outpath, file=constants.ofile)
                runTALYS(int(100 * e_a) / 100., ele, A)
            else:
                print("Warning, no (alpha,n) data found for E_a =", e_a, "MeV on
target", target,
                    "...skipping. Consider running with the -d or -t options",
file=constants.ofile)
                return {}

    if not os.path.exists(outpath):
        return {}

    f = open(fname)
    spec = {}
    tokens = map(lambda line: line.split(), f.readlines())
    for line in tokens:
        if len(line) < 1 or line[0] == 'EMPTY':
            break
        if line[0][0] == '#':
            continue
        # line[0] = E-out
        # line[1] = Total
        # line[2] = Direct
        # line[3] = Pre-equil
        # line[4] = Mult. preeq
        # line[5] = Compound
        # line[6] = Pre-eg ratio
        # convert from mb/MeV to cm^2/MeV
        energy = int(float(line[0]) * constants.MeV_to_keV)
        sigma = float(line[1]) * constants.mb_to_cm2 / constants.MeV_to_keV
        spec[energy] = sigma
    return spec

def rebin(histo, step, minbin, maxbin):
    nbins = (maxbin - minbin) / step
    newhisto = {}
    normhisto = {}
    for i in sorted(histo):
        index = sorted(histo).index(i)
        delta = sorted(histo)[0]
        if index > 0:

```



```

        delta = sorted(histo)[index] - sorted(histo)[index - 1]
    if i < minbin:
        print('Underflow: ', i, ' (minbin = ', minbin, ')',
file=constants.ofile)
        if -1 in newhisto:
            newhisto[-1] += histo[i] * delta
            normhisto[-1] += delta
        else:
            newhisto[-1] = histo[i] * delta
            normhisto[-1] = delta
        # ...or the overflow bin if too high
    if i > maxbin:
        print('Overflow: ', histo[i], ' (maxbin = ', maxbin, ')',
file=constants.ofile)
        overflowbin = int(nbins + 10 * step)
        if overflowbin in newhisto:
            newhisto[overflowbin] += histo[i] * delta
            normhisto[overflowbin] += delta
        else:
            newhisto[overflowbin] = histo[i] * delta
            normhisto[overflowbin] = delta
        newbin = int(minbin + int((i - minbin) / step) * step)
        if newbin in newhisto:
            newhisto[newbin] += histo[i] * delta
            normhisto[newbin] += delta
        else:
            newhisto[newbin] = histo[i] * delta
            normhisto[newbin] = delta
    for i in newhisto:
        if normhisto[i] > 0:
            newhisto[i] /= normhisto[i]

    return newhisto

def integrate(histo):
    integral = 0
    for i in sorted(histo):
        delta = sorted(histo)[0]
        index = sorted(histo).index(i)
        if index > 0:
            delta = sorted(histo)[index] - sorted(histo)[index - 1]

        integral += histo[i] * delta
    return integral

def readTotalNXsect(e_a, ele, A):
    fname = isoDir(ele, A) + 'TalysOut/outputE' + str(int(100 * e_a) / 100.)
    if not os.path.exists(fname):
        print("Could not find file ", fname, file=constants.ofile)
        return 0
    f = open(fname)
    lines = map(lambda line: line.split(), f.readlines())
    xsect_line = 0
    for line in lines:
        if line == ['2.', 'Binary', 'non-elastic', 'cross', 'sections',
'(non-exclusive)']:
            break
        else:
            xsect_line += 1

    xsect_line += 3
    if len(lines) < xsect_line:

```

```

        return 0
    if lines[xsect_line][0] != 'neutron':
        return 0
    sigma = float(lines[xsect_line][2])
    sigma *= constants.mb_to_cm2
    return sigma

def condense_alpha_list(alpha_list, alpha_step_size):
    alpha_ene_cdf = []
    max_alpha = max(alpha_list)
    e_a_max = int(max_alpha[0] * 100 + 0.5) / 100.
    alpha_ene_cdf.append([e_a_max, max_alpha[1]])
    e_a = e_a_max
    while e_a > 0:
        cum_int = 0
        for alpha in alpha_list:
            this_e_a = int(alpha[0] * 100 + 0.5) / 100.
            if this_e_a >= e_a:
                cum_int += alpha[1]
            alpha_ene_cdf.append([e_a, cum_int])
            e_a -= alpha_step_size
    return alpha_ene_cdf

def run_alpha(alpha_list, mat_comp, e_alpha_step):
    spec_tot = {}
    xsects = {}
    total_xsect = 0
    counter = 0
    alpha_ene_cdf = condense_alpha_list(alpha_list, e_alpha_step)
    for [e_a, intensity] in alpha_ene_cdf:
        counter += 1
        if counter % (int(len(alpha_ene_cdf) / 100)) == 0:
            sys.stdout.write('\r')
            sys.stdout.write(
                "[%100s] %d%%" % ('=' * int(counter * 100 /
len(alpha_ene_cdf)), 100 * counter / len(alpha_ene_cdf)))
            sys.stdout.flush()

        stopping_power = 0
        if stopping_power == 0:
            stopping_power = calcStoppingPower(e_a, mat_comp)
        for mat in mat_comp:
            mat_term = getMatTerm(mat, mat_comp)
            spec_raw = getIsotopeDifferentialNSpec(e_a, mat.ele, mat.A)
            spec = rebin(spec_raw, constants.delta_bin, constants.min_bin,
constants.max_bin)
            delta_ea = e_alpha_step
            if e_a - e_alpha_step < 0:
                delta_ea = e_a
            prefactors = (intensity / 100.) * mat_term * delta_ea /
stopping_power
            xsect = prefactors * readTotalNXsect(e_a, mat.ele, mat.A)
            total_xsect += xsect
            matname = str(mat.ele) + str(mat.A)
            if matname in xsects:
                xsects[matname] += xsect
            else:
                xsects[matname] = xsect
            for e in spec:
                val = prefactors * spec[e]
                if e in spec_tot:

```

```

        spec_tot[e] += val
    else:
        spec_tot[e] = val

    sys.stdout.write('\r')
    sys.stdout.write(
        "[%-100s] %d%%" % ('=' * int((counter * 100) / len(alpha_ene_cdf)),
100 * (counter + 1) / len(alpha_ene_cdf))
    sys.stdout.flush()
    newspec = spec_tot
    print('', file=constants.ofile)
    print('# Total neutron yield = ', total_xsect, ' n/decay',
file=constants.ofile)
    for x in sorted(xsects):
        print('\t', x, xsects[x], file=constants.ofile)
    print('# Integral of spectrum = ', integrate(newspec), " n/decay",
file=constants.ofile)
    for e in sorted(newspec):
        print(e, newspec[e], file=constants.ofile)

def help_message():
    print(
        "Usage: You must specify an alpha list or decay chain file and a
target material file.\nYou may also specify a step size to for integrating
the alphas as they slow down in MeV; the default value is 0.01 MeV\n\t-l
[alpha list file name]\n\t-c [decay chain file name]\n\t-m [material
composition file name]\n\t-s [alpha step size in MeV]\n\t-t (to run TALYS for
reactions not in libraries)\n\t-d (download isotopic data for isotopes
missing from database)\n\t-o [output file name]",
        file=sys.stdout)

def main():
    alpha_list = []
    mat_comp = []
    alpha_step_size = 0.01
    for arg in sys.argv:
        if arg == '-l':
            alphas_list_file = sys.argv[sys.argv.index(arg) + 1]
            print('load alpha list', alphas_list_file, file=sys.stdout)
            alpha_list = loadAlphaList(alphas_list_file)
        if arg == '-c':
            chain_file = sys.argv[sys.argv.index(arg) + 1]
            print('load alpha chain', chain_file, file=sys.stdout)
            alpha_list = loadChainAlphaList(chain_file)
        if arg == '-m':
            mat_file = sys.argv[sys.argv.index(arg) + 1]
            print('load target material', mat_file, file=sys.stdout)
            mat_comp = readTargetMaterial(mat_file)
        if arg == '-s':
            alpha_step_size = float(sys.argv[sys.argv.index(arg) + 1])
            print('step size', alpha_step_size, file=sys.stdout)
        if arg == '-h':
            help_message()
            return 0
        if arg == '-t':
            constants.run_talys = True
        if arg == '-d':
            constants.download_data = True
        if arg == '--print-alphas':
            constants.print_alphas = True
        if arg == '--print-alphas-only':
            print('Only printing alphas', file=sys.stdout)

```

```

        constants.print_alphas = True
        constants.run_alphas = False
    if arg == '--force-recalculation':
        constants.force_recalculation = True
    if arg == '-o':
        ofile = str(sys.argv[sys.argv.index(arg) + 1])
        print('Printing output to', ofile, file=sys.stdout)
        constants.ofile = open(ofile, 'w')
        # sys.stdout = open(ofile, 'w')

    if len(alpha_list) == 0 or len(mat_comp) == 0:
        if len(alpha_list) == 0: print('No alpha list or chain specified',
file=sys.stdout)
        if len(mat_comp) == 0: print('No target material specified',
file=sys.stdout)
        print('', file=sys.stdout)
        help_message()
        return 0

    if constants.print_alphas:
        print('Alpha List: ', file=sys.stdout)
        print(max(alpha_list), file=sys.stdout)
        condense_alpha_list(alpha_list, alpha_step_size)
        for alph in alpha_list:
            print(alph[0], '&', alph[1], '\\\\', file=sys.stdout)

    if constants.download_data:
        for mat in mat_comp:
            ele = mat.ele
            if not os.path.exists('./Data/Isotopes/' + ele.capitalize()):
                print('\tDownloading data for', ele, file=sys.stdout)
                bashcmd = './Scripts/download_element.sh ' + ele
                process = subprocess.call(bashcmd, shell=True)

    if constants.run_alphas:
        print('Running alphas:', file=sys.stdout)
        run_alpha(alpha_list, mat_comp, alpha_step_size)

if __name__ == '__main__':
    main()

```