

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «Комплекс інтерактивних тренажерів для вивчення технологій анімації в 3ds Max»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ.м-01 Дуванов Семен Сергійович

**Кваліфікаційну роботу
захищено на засіданні ЕК
з оцінкою**

«__» грудня 2021 р.

Науковий керівник

(підпис)

к.т.н., доц., Баранова І.В.

Голова комісії

(підпис)

Шифрін Д.М.

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2021

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. кафедри ІТ

_____ В. В. Шендрик
«__» _____ 2021 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Дуванов Семен Сергійович
(прізвище, ім'я, по батькові)

1 Тема проекту Комплекс інтерактивних тренажерів для вивчення технологій анімації в 3ds Max

затверджена наказом по університету від «29» жовтня 2021 р. № 0787-VI

2 Термін здачі студентом закінченого проекту «__» _____ грудня _____ 2021 р.

3 Вхідні дані до проекту технічне завдання на розробку комплексу інтерактивних тренажерів

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) аналіз предметної області, постановка задачі та методи дослідження, проектування комплексу веб-тренажерів, розробка комплексу інтерактивних веб-тренажерів

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) актуальність, постановка задачі, огляд аналогів, функціональні вимоги до тренажерів, засоби реалізації, технології реалізації, структурно-функціональне моделювання, діаграма варіантів використання, практична реалізація, висновки

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)Завдання прийняв до виконання _____
(підпис)**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів випускної проекту	Термін виконання етапів проекту	Примітка
1	Аналіз досліджень та аналогів	02.11.2021 - 03.11.2021	
2	Вибір технологій реалізації	03.11.2021 - 04.11.2021	
3	Планування робіт з ІТ-проекту	04.11.2021 - 06.11.2021	
4	Структурно-функціональне моделювання	07.11.2021 - 09.11.2021	
5	Створення макетів інтерфейсів	10.11.2021 - 12.11.2021	
6	Верстання	12.11.2021 - 16.11.2021	
7	Додавання розділів із теорією	16.11.2021 - 20.11.2021	
8	Створення завдань	20.11.2021 - 24.11.2021	
9	Впровадження різних варіантів локалізації інтерфейсів	24.11.2021 - 27.11.2021	
10	Виправлення багів	27.11.2021 - 30.11.2021	
11	Виправлення недоліків розмітки	27.11.2021 - 30.11.2021	
12	Розміщення веб-додатків на хостингу дистанційного курсу	30.11.2021 - 01.12.2021	
13	Завершення оформлення документації	30.11.2021 - 04.12.2021	

Магістрант _____
(підпис)

Дуванов С.С.

Керівник роботи _____
(підпис)

к.т.н., доц. Баранова І.В.

РЕФЕРАТ

Тема кваліфікаційної роботи магістра «Комплекс інтерактивних тренажерів для вивчення технологій анімації в 3ds Max».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 34 найменувань, додатків. Загальний обсяг роботи – 116 сторінок, у тому числі 50 сторінок основного тексту, 4 сторінки списку використаних джерел, 65 сторінок додатків.

Кваліфікаційну роботу магістра присвячено розробці серії інтерактивних веб-тренажерів для вивчення основ створення анімації в програмі 3Ds Max.

В роботі проведено аналіз предметної області та існуючих аналогів, розглянуто їх недоліки та переваги. Обрано засоби та технології реалізації проекту. Виконано моделювання роботи серії веб-тренажерів з використанням методологій IDEF та UML, проведено планування робіт.

Розроблено дизайн інтерфейсів, створено теоретичні блоки та блоки із завданнями для кожного веб-тренажера із серії.

Результатом виконання проекту є розроблена серія веб-тренажерів для вивчення основ створення тривимірної анімації в програмі 3Ds Max для дистанційного курсу дисципліни «Технології анімації та візуалізації» університету СумДУ за темами: «Кінематика пов'язаних об'єктів», «Анімація об'єктів Bones», «Анімація за ключовими кадрами».

Ключові слова: дистанційний курс, тренажер, комплекс, веб-додаток, анімація, 3ds Max.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Дослідження актуальності проблеми.....	8
1.2 Аналіз публікації з предметної області	9
1.3 Аналіз програмних продуктів – аналогів	10
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ.....	16
2.1 Мета та задачі	16
2.2 Вибір засобів реалізації	17
3 ПРОЕКТУВАННЯ КОМПЛЕКСУ ВЕБ-ТРЕНАЖЕРІВ	19
3.1 Структурно-функціональне моделювання	19
3.2 UML моделювання комплексу веб-тренажерів	22
4 РОЗРОБКА КОМПЛЕКСУ ІНТЕРАКТИВНИХ ТРЕНАЖЕРІВ.....	26
4.1 Структура веб-додатку	26
4.2 Розробка веб-тренажерів	28
4.3 Результат розробки комплексу веб-тренажерів	39
ВИСНОВКИ.....	46
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	47
Додаток А.....	51
Додаток Б	58
Додаток В.....	96

ВСТУП

Наразі ринок ігрової індустрії стрімко зростає та розвивається, що зумовлює появу більшої кількості ігрових студій та зростання попиту на фахівців у сфері розробки комп'ютерних ігор. Анімація персонажів та навколишнього середовища є невід'ємною часткою процесу розробки ігор. Веб-тренажери із наявною теорією можуть стати в пригоді на початку вивчення основ тривимірної анімації в 3ds Max.

Оскільки 3ds Max сьогодні є однією з найбільш затребуваних програм для тривимірного моделювання та створення анімації, веб-тренажер, що містить компактно поданий теоретичний матеріал та практичні тестові завдання, може допомогти на початку навчання, але тільки на початку, оскільки він далі все одно не зможе зрівнятися із практикою безпосередньо у самій програмі 3ds Max. Також він буде корисним при роботі в програмі 3Ds Max паралельно як довідник.

Об'єктом дослідження дипломної роботи магістра є технології розробки електронних засобів підтримки навчання.

Предметом дослідження є веб-тренажери для закріплення практичних навичок з навчання анімації в 3ds Max.

Метою дипломної роботи є створення серії схожих веб-додатків для вивчення основ створення тривимірної анімації в програмі 3ds Max, що надають компактний теоретичний матеріал із великою кількістю ілюстрацій та тестові завдання для закріплення теоретичного матеріалу або перевірки якості засвоєння матеріалу.

Основними задачами досягнення мети кваліфікаційної роботи є:

- проведення аналізу предметної області та огляд існуючих додатків-аналогів;
- розробка структури веб-тренажерів;
- розробка дизайну веб-інтерфейсів серії веб-тренажерів;
- наповнення веб-тренажерів контентом: теоретичним матеріалом та завданнями.

Практична цінність роботи полягає в допомозі у вивченні азів (теорії) створення тривимірної анімації в 3ds Max. Також додатки можуть виконувати роль довідників при проведенні практичних занять у програмі 3ds Max.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження актуальності проблеми

Останні три роки показують тенденцію збільшення попиту на комп'ютері ігри. Ця тенденція підкріплюється статистикою об'ємів продажів щорічно. За прогнозами попит ще більше зростатиме, тому сферу розробки комп'ютерних ігор можна вважати перспективною [1–4]. Збільшення попиту можна також частково обґрунтувати сучасним епідеміологічним станом у світі – люди вимушені більше часу проводити вдома, що іноді спонукає деяких витратити свій вільний час, перебуваючи у віртуальному світі.

Розробка комп'ютерних ігор охоплює велику кількість професійних напрямків: гейм-дизайнери (розробляють основну концепцію відеоігор, тісно взаємодіють із усіма іншими учасниками команди, є одними із головних рушіїв всього процесу розробки), програмісти (відповідають за технічні аспекти процесу розробки, програмно реалізують механіки гри), художники (створюють концепти, вигляд персонажів, оточення, кінцевим продуктом чого є зображення), аніматори, звукорежисери (створюють саундтреки, музикальний супровід, займаються саунд-дизайном), сценаристи, тестувальники та інші. Список можна ще продовжувати, але основні напрямки було вказано [5].

Аніматор – є одним із найважчих напрямків, оскільки досвідчений аніматор повинен бути спостережливим, відчувати тонкощі взаємодії людей з реальним світом, з предметами, і вміти це переносити якомога точніше у віртуальний світ.

Чим більш природньо ігрові персонажі рухатимуться, взаємодіятимуть із оточенням, тим більш цікавий досвід отримають користувачі, тим більше буде оцінка гри у відгуках, тим більш буде попит на гру, але це лише один із факторів успіху.

Зараз існує така технологія як *motion capture*, яка може значно облегшити роботу аніматора, але вона потребує спеціального обладнання, приміщення та

наявність акторів, тому вона може бути застосована лише в середніх та великих проектах. Але в цьому випадку вирішальне слово все одно буде за аніматором. Сучасний 3d аніматор повинен володіти великою кількістю специфічних навичок, він є затребуваним на ринку праці та доволі оплачуваним [6].

Оскільки річ йде про 3d-анімацію, для цього знадобиться програма для тривимірного моделювання та створення анімації. Їх існує велика кількість, але найпопулярнішими серед них є: 3ds Max, Maya, Blender, Houdini, Cinema4D [7, 8]. Оскільки 3ds Max стабільно займає стійку позицію на ринку програм даного напрямку – доцільність її вивчення не підпадає під сумнів.

Майже вся інформація з вивчення основ створення анімації представлена у вигляді сайтів із довідниковою інформацією або у вигляді навчальних відео, тому створення серії веб-додатків із компактно зібраним теоретичним матеріалом та інтерактивними завданнями можна вважати актуальним.

Інтерактивні завдання можуть допомогти засвоїти теоретичну інформацію без наявності 3ds Max на комп'ютері, але найкориснішим варіантом буде практика у самій програмі із паралельним використанням довідникової інформації із веб-ресурсів.

Також серія веб-додатків може служити для перевірки якості засвоєння матеріалу. Доцільність розробки саме веб-додатків можна обґрунтувати відсутністю необхідності його встановлювати, чого не можна сказати у випадку десктопного чи мобільного додатків .

1.2 Аналіз публікації з предметної області

Веб-тренажери можуть бути потужним інструментом у навчанні. Одна із сфер, де веб-тренажери можуть дуже знадобитися, – це медицина. Оскільки медицина має безліч напрямків – від анатомії до фармакології, поле для вибору теми веб-тренажеру є досить широким. Однак, створення якісного веб-тренажеру вимагає занурення

розробника в цільову предметну область та потребує багато часу, тому багато напрямків досі залишаються не дослідженими [9].

Аналогічний підхід застосовується і в інших тематиках, наприклад, термоядерна реакція в реакторі із області фізики. Просте читання того, як саме все це відбувається, може виявитися не дуже зрозумілим із самого початку. Відеоролик-демонстрація може допомогти із розумінням, але для повного розуміння слід відтворити послідовність реакцій самотужки, з чим може допомогти веб-симулятор або веб-тренажер [10].

Слід звернути увагу, що веб-симулятор або веб-тренажер є лише одним із засобів допомоги у навчанні. Існує безліч корисних android-додатків, десктопних додатків, але вони всі потребують завантаження встановлення та займають місце.

Багато сервісів реалізовано в декількох варіантах одночасно. Епідеміологічний стан у світі спровокував перехід великої кількості вищих навчальних закладів на дистанційне навчання [11].

До переваг дистанційного навчання можна віднести: легка доступність до учбових матеріалів, спрощення процесу здачі лабораторних, збереження часу та грошей за відсутності необхідності приїзду в заклад.

До недоліків можна віднести: відсутність живого спілкування [12]. Враховуючи актуальність дистанційного навчання на сьогодні, можна зробити висновок, що веб-тренажер за рахунок доступності та своєї природи є досить непоганим інструментом внесення різноманіття в процес навчання.

1.3 Аналіз програмних продуктів – аналогів

Продукт дипломної роботи – комплекс схожих веб-додатків, тематикою яких є анімація в 3ds Max, не має аналогів, оскільки тема є доволі вузькою. Натомість існують приклади веб-тренажерів за іншими напрямками вивчення англійської мови, програмування, тощо.

Прикладом доволі відомого тренажеру для вивчення англійської є Puzzle English (рис.1.1). Він містить велику кількість завдань різного типу: від аудіювання та щоденного закріплення лексичного набору слів, до диктантів та логічного доповнення речень. Тренажер є доволі ефективним, але його недоліком є лімітований функціонал у разі безоплатного режиму доступу [13].

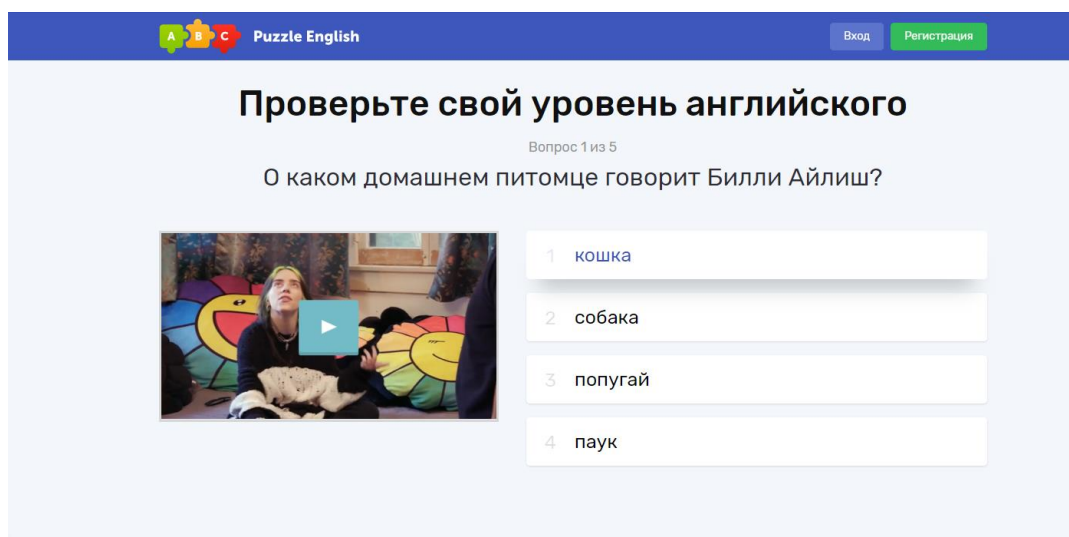


Рисунок 1.1 – Веб-тренажер для вивчення англійської Puzzle English

Також існує веб-тренажер EnglishDom [14], дуже подібний за своїм функціоналом до Puzzle English (рис.1.2).

Серед інших веб-додатків із вивчення англійської мови слід виділити Free Grammar Checker Online [15], який перевіряє граматику написаного речення англійською та підказує правильний варіант (рис.1.3).

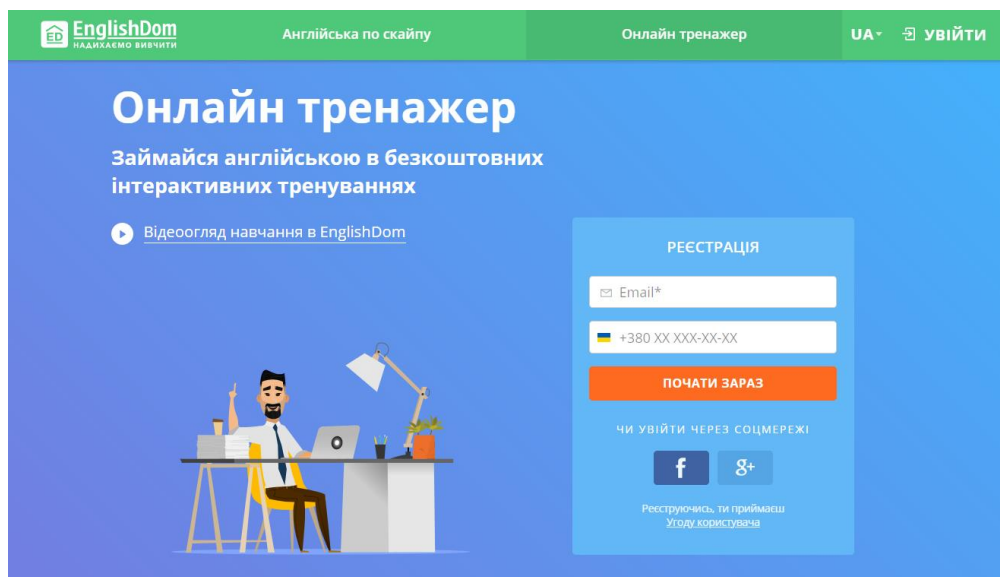


Рисунок 1.2 – Веб-тренажер для вивчення англійської EnglishDom

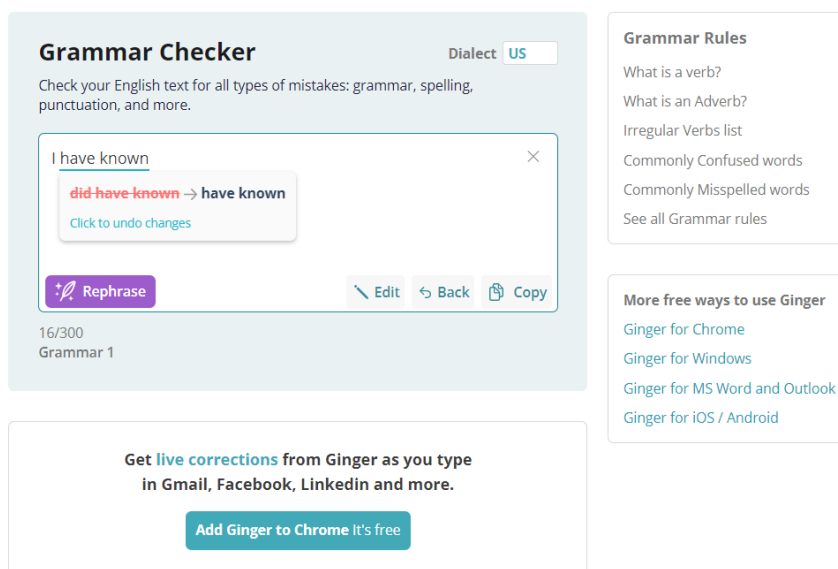


Рисунок 1.3 – Веб-тренажер Free Grammar Checker Online для перевірки правильності граматики при написанні речень

Серед тренажерів для навчання програмуванню слід приділити увагу CodeCombat [16]. Він містить декілька режимів, серед яких ігрово-орієнтований та текстово-орієнтований (рис.1.4-1.5). До переваг додатку можна віднести його безкоштовність та представлення процесу навчання у вигляді гри. Цей тренажер дозволяє зрозуміти сенс такого поняття як алгоритм та робить навчання завдяки

ігровій складовій, що інтерактивно змінюється в залежності від твоїх дій, більш цікавим. Недоліком цього додатку лише можна назвати вузький діапазон цільової аудиторії: діти та підлітки. Для дорослих, скоріш за все, такий підхід до вивчення мов програмування буде не актуальним або ж не дуже цікавим.

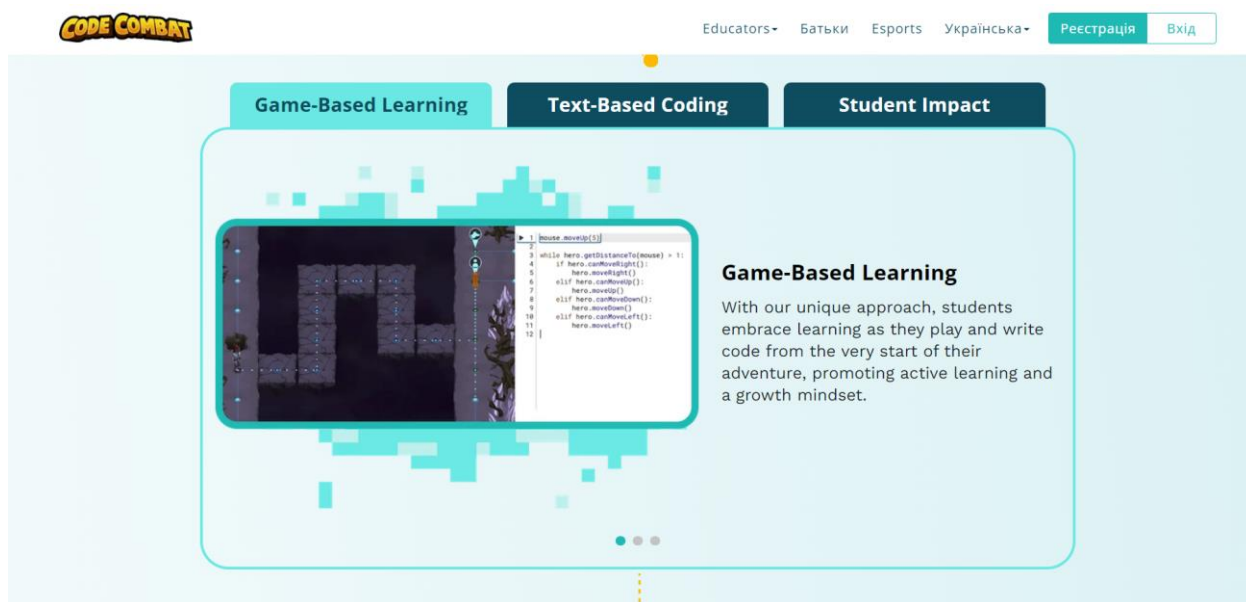


Рисунок 1.4 – Веб-тренажер CodeCombat навчання програмуванню

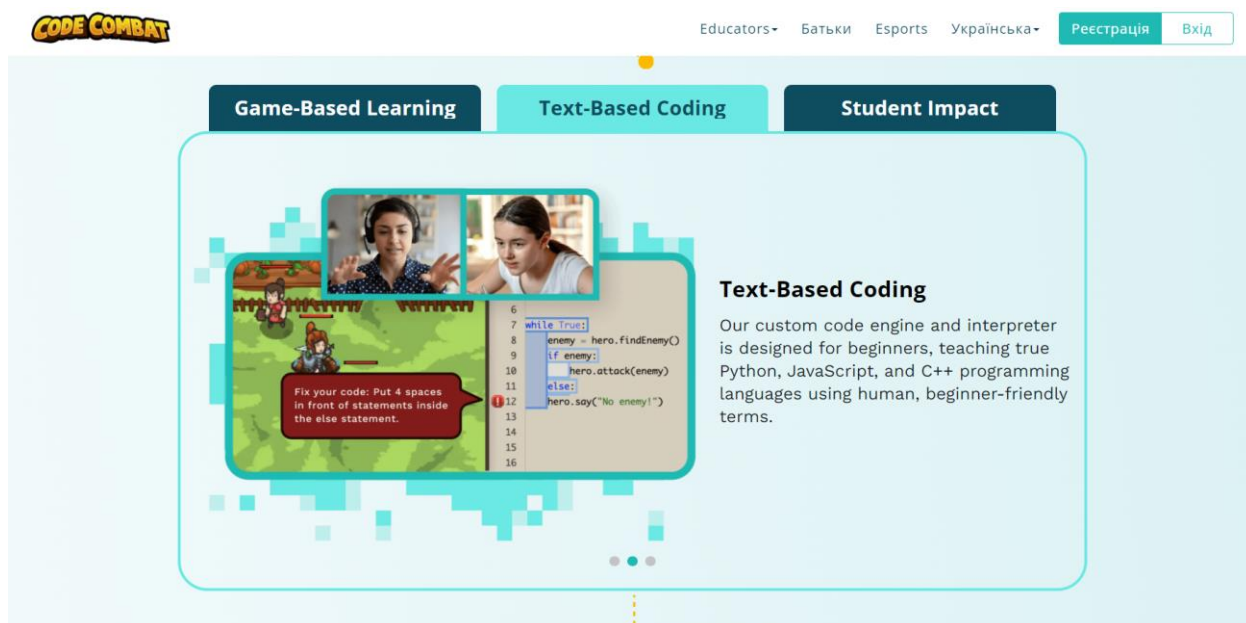



Рисунок 1.5 – Веб-тренажер CodeCombat навчання програмуванню

Існують подібні за функціоналом веб-тренажери до розроблених, що пов'язані із програмою 3ds Max [17]. Вони розроблені із використанням фреймворку для тренажерів, що використовуються і для розробки даного проєкту. Проте зазначені тренажери охоплюють тему 3d-моделювання, а не створення 3d-анімації. Нижче наведено приклад завдання одного з розроблених тренажерів (рис.1.6).

Вступ / Інструкція / Крок 1 / Крок 2 / Крок 3 / Крок 4 / Результати

Крок 1. Створення примітива Вох

Завдання: В командній панелі інтерфейсу натисніть в правильному порядку піктограми команд для створення елемента Вох. Спочатку вкажіть колір, потім параметри розміру фігури у відповідних полях.

Довжина	Шерина	Висота	Колір
50	30	25	

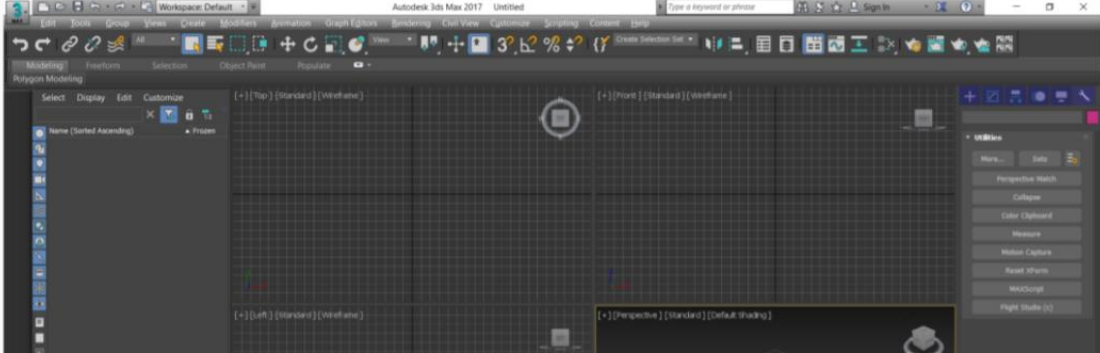


Рисунок 1.6 – Веб-тренажер для вивчення основних примітивів в 3ds Max

Розглянуті приклади можна вважати цікавими, але слід враховувати, що вони відносяться до інших тем, їх масштаб перевищує масштаб проєкту магістерської та до їх розробки залучалась велика кількість людей, або ж у випадку невеликої кількості людей, на розробку вищезгаданих тренажерів було витрачено багато часу.

На основі розглянутих прикладів було створено порівняльну таблицю (табл. 1.1).

Таблиця 1.1 – Порівняльна таблиця веб-тренажерів

Додаток Критерій	Puzzle English	English Dom	Free Grammar Checker Online	Code Combat	Веб-тренажер з моделювання примітивів у 3ds Max
Вільний доступ (відсутність реєстрації)	0	0	1	0	1
Безкоштовність	1	1	2	2	2
Охоплення цільової тематики	0	0	0	0	1
Відповідність предметній області дослідження	0	0	0	0	0
Дизайн	1	1	1	1	1
Підсумок	2	2	4	3	5

У результаті проведеного аналізу можна дійти до висновку, що розробка комплексу веб-тренажерів з анімації в 3ds Max є актуальною. Продукт дипломної роботи може бути використаний як довідник, буде корисним на початку вивчення інформації, під час практичних занять в 3ds Max, та слугуватиме інструментом перевірки якості закріплення практичних навичок або засвоєння теоретичного матеріалу.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі

Метою проекту є розробка комплексу схожих веб-тренажерів із вивчення основ створення анімації в 3ds Max для дистанційного курсу «Технології анімації та візуалізації». Кожен веб-тренажер охоплюватиме певну тему із набору: «Кінематика пов'язаних об'єктів», «Анімація об'єктів Bones», «Анімація за ключовими кадрами».

Ціллю розробки комплексу веб-додатків є підтримка процесу дистанційного навчання за рахунок надання теоретичного матеріалу, поліпшення процесу навчання завдяки інтерактивній взаємодії слухача із тренажером, допомоги під час виконання практичних завдань, закріплення отриманих знань та перевірка їх якості.

Для виконання поставленої мети потрібно вирішити наступний перелік задач:

- провести аналіз відкритих джерел за темою дослідження, розглянути схожі проекти, визначити їх переваги та недоліки, визначити методи і технології реалізації проекту;
- проаналізувати навчальний теоретичний матеріал за поданими темами тренажерів, структурувати його та подати його у вигляді теоретичного блоку для кожного додатку;
- розробити сценарії виконання завдань та самі завдання для веб-додатків;
- розробити варіанти локалізації для інтерфейсу веб-тренажерів: українська, російська, англійська;
- розробити дизайн кожного додатку;
- виконати тестування веб-додатків.

Оскільки комплекс веб-тренажерів розміщуватиметься на сервері дистанційного середовища навчання СумДУ, для їх розробки слід використовувати фреймворк тренажерів СумДУ. Це дозволить розробленим веб-додаткам

вмонтуватися в оточення, працювати безперебійно та взаємодіяти із загальною системою.

Кожен веб-тренажер складатиметься із теоретичного блоку, практичного блоку (блок із тестовими завданнями) та блоку-результатів. У ході проходження завдань користувач зможе переміщатися до теоретичного блоку та назад до поточного завдання.

Кожне завдання міститиме три спроби відповіді – це зроблено для того, щоб студент міг у разі помилки проаналізувати її, повернутися до теоретичного блоку та відповісти потім вірно.

Кожен тренажер матиме 3 варіанта локалізації інтерфейсу: українську, російську та англійську, і користувач на початку сеансу матиме змогу вибрати зручний для себе варіант інтерфейсу.

2.2 Вибір засобів реалізації

Оскільки метою дипломного проекту є розробка веб-тренажерів – створення веб-тренажерів відбувається із використанням фреймворку для тренажерів дистанційного навчання СумДУ. Фреймворк (рушій) тренажерів написано повністю на мові JavaScript, тому логіка тренажерів описується відповідною мовою. Веб-тренажер представлений односторінковим веб-додатком (контент додається та видаляється з однієї сторінки).

Середовищем розробки було обрано IDE PhpStorm, оскільки воно дає змогу взаємодіяти із директорією веб-тренажеру як з проектом, містить функціонал автодоповнення коду та інший функціонал, що набагато зручніше, аніж написання коду в текстових редакторах. IDE PhpStorm є наразі одним із найбільш популярних середовищ розробки [18].

Для написання та редагування тексту теоретичного блоку обрано текстовий редактор Adobe Brackets, оскільки в ньому зручно підготувати текст перед занесенням його в json-файли з даними локалізацій.

Для створення ілюстрацій та gif-анімацій із теоретичного та практичного блоків використовується 3ds Max як джерело графічної інформації, та програма ScreenToGif – для запису екрану відразу у формат gif-зображення.

Оскільки веб-тренажери потребують наявності логотипів, то для їх створення було використано Adobe Illustrator – найкраща та найзручніша з існуючих програм для створення та редагування векторних зображень [19].

Для підтримки розробки веб-тренажерів було використано локальний сервер OpenServer, оскільки рушій тренажерів не працює без наявності серверу.

3 ПРОЕКТУВАННЯ КОМПЛЕКСУ ВЕБ-ТРЕНАЖЕРІВ

3.1 Структурно-функціональне моделювання

Функціональну модель роботи системи описано за допомогою методології IDEF0, основними складовими якої є блоки та стрілки. Кожен блок є функцією або певним процесом, що описується дієсловом. До кожного блоку входять або виходять стрілки.

Стрілки, що входять зліва в блок, називаються вхідними даними, зверху – контролюючими елементами, знизу – механізмами (засоби, що підтримують виконання функції), що виходять праворуч – вихідними даними [20][21].

Для створення діаграм IDEF0 обрано один із інструментів моделювання бізнес процесів – CA ERwin Process Modeler [22].

Оскільки архітектура усіх веб-додатків із комплексу – ідентична, то одна IDEF0 описуватиме функціонування всіх додатків. Після створення діаграми IDEF0 «Комплекс інтерактивних тренажерів для вивчення технологій анімації в 3ds Max» було визначено наступний перелік даних:

- вхідні дані: Набір завдань, Початок тренування;
- керуючі елементи процесу: Теоретичний матеріал, Алгоритм роботи веб-додатку.
- вихідні дані: Кількість набраних балів, Час проходження тренування.
- механізми процесу: Студент, Викладач, Програмне забезпечення.

Діаграма IDEF0 представлена на рис. 3.1.



Рисунок 3.1 – Контекстна діаграма основного процесу

Далі для деталізації функціонування тренажерів із комплексу процес «Тренування та перевірка якості засвоєння матеріалу» декомпозиується на чотири процеси: «Перегляд теорії», «Виконання завдань», «Перевірка відповідей», «Виведення результатів».

Перший процес має наступний перелік даних:

- вхідні дані: Початок тренування;
- керуючі елементи процесу: Теоретичний матеріал, Алгоритм роботи веб-додатку;
- вихідні дані: Теоретичний блок переглянуто;
- механізми процесу: Студент, Викладач, Програмне забезпечення.

Другий процес має наступний перелік даних:

- вхідні дані: Теоретичний блок переглянуто, Набір завдань;
- керуючі елементи процесу: Теоретичний матеріал, Алгоритм роботи веб-додатку;
- вихідні дані: Завдання виконано;
- механізми процесу: Студент, Викладач, Програмне забезпечення.

Третій процес має наступний перелік даних:

- вхідні дані: Завдання виконано;
- керуючі елементи процесу: Алгоритм роботи веб-додатку;
- вихідні дані: Відповідь перевірено;
- механізми процесу: Студент, Викладач, Програмне забезпечення.

Четвертий процес має наступний перелік даних:

- вхідні дані: Відповідь перевірено;
- керуючі елементи процесу: Алгоритм роботи веб-додатку;
- вихідні дані: Кількість набраних балів, Час проходження тренування;
- механізми процесу: Програмне забезпечення.

Діаграму декомпозиції процесу «Тренування та перевірка якості засвоєння матеріалу» зображено на рисунку 3.2.

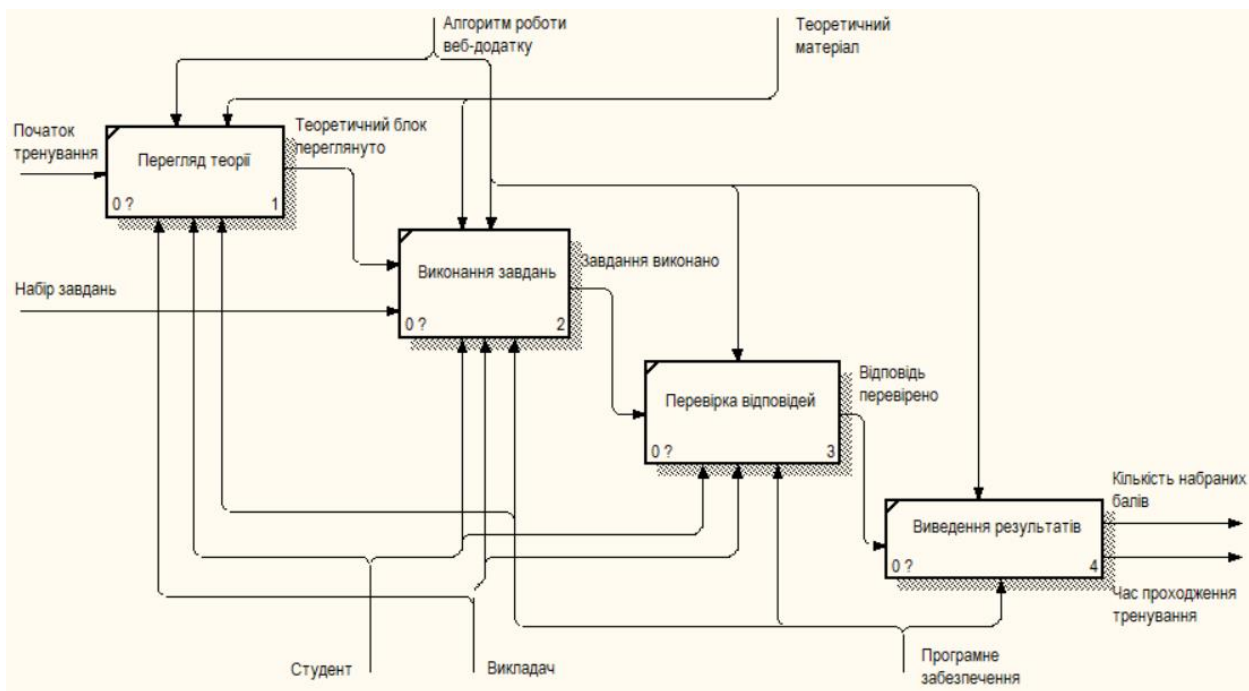


Рисунок 3.2 – Декомпозиція процесу «Тренування та перевірка якості засвоєння матеріалу»

3.2 UML моделювання комплексу веб-тренажерів

Для проектування системи використаємо мову UML – мову графічного проектування та документування систем [23, 24].

В першу чергу створимо діаграму варіантів використання, що описує загальний функціонал системи, до якого має доступ кожна група користувачів [25]. Діаграму Use Case веб-тренажерів представлено на рис.3.3.

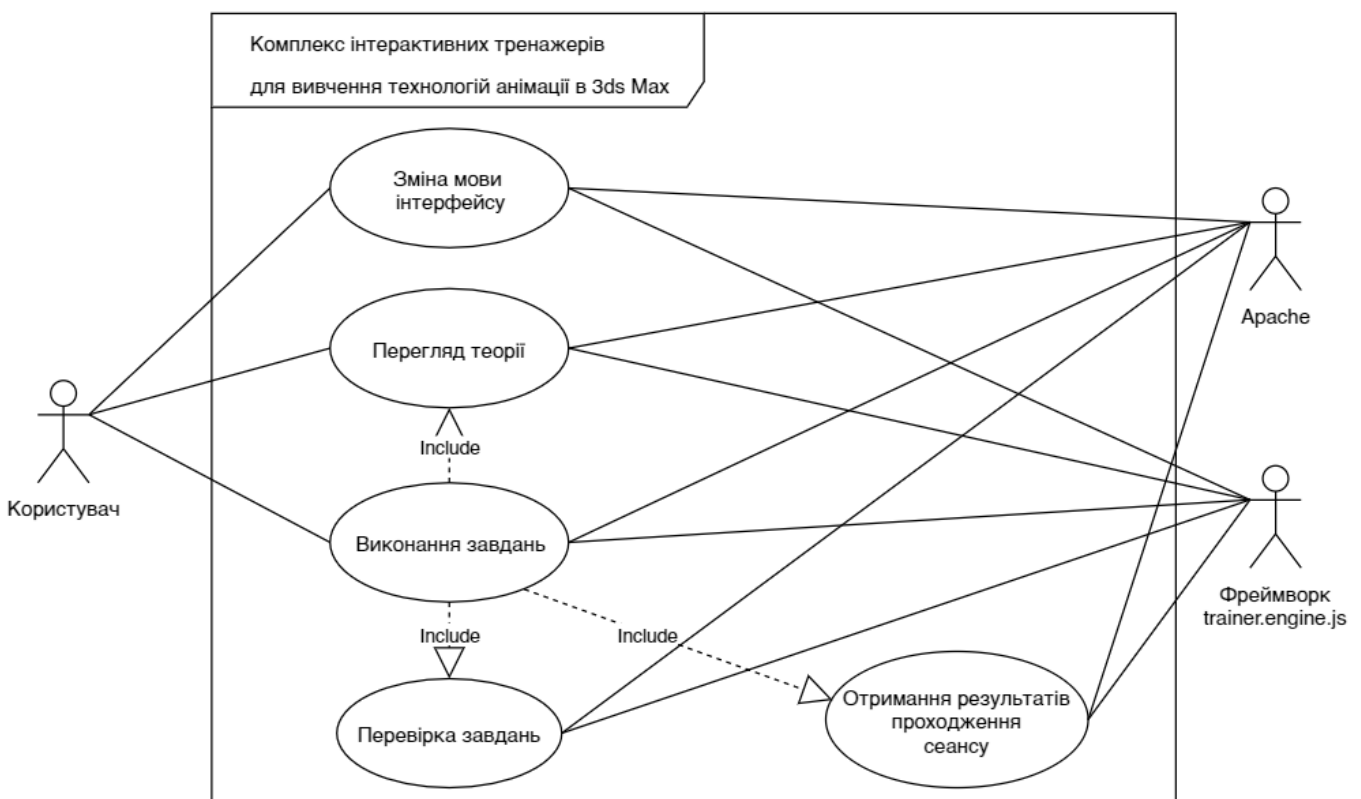


Рисунок 3.3 – Діаграма варіантів використання веб-тренажерів

Для комплексу інтерактивних тренажерів з вивчення технологій анімації в 3ds Max були виділені наступні актори:

- Користувач – користувач системи, проходить тренування та отримує його результат;
- Араче – локальний сервер, який підтримує роботу всієї системи;

- Фреймворк `trainer.engine.js` – скрипт-рушій фреймворку, на основі якого створені веб-тренажери.

Варіанти використання:

- Зміна мови інтерфейсу;
- Перегляд теорії - перегляд теоретичного блоку як перед виконанням завдань, так і паралельно;
- Виконання завдань - процес проходження тестових завдань;
- Перевірка завдань - перевірка завдання після надання відповіді;
- Отримання результатів проходження сеансу - отримання даних в кінці сеансу по часу виконання завдань та кількості правильних відповідей.

Чотири варіанти використання стали основами для створення діаграм послідовностей, що служать для уточнення описання функціоналу системи [26]. Діаграми послідовностей представлені рисунками 3.4 – 3.7.

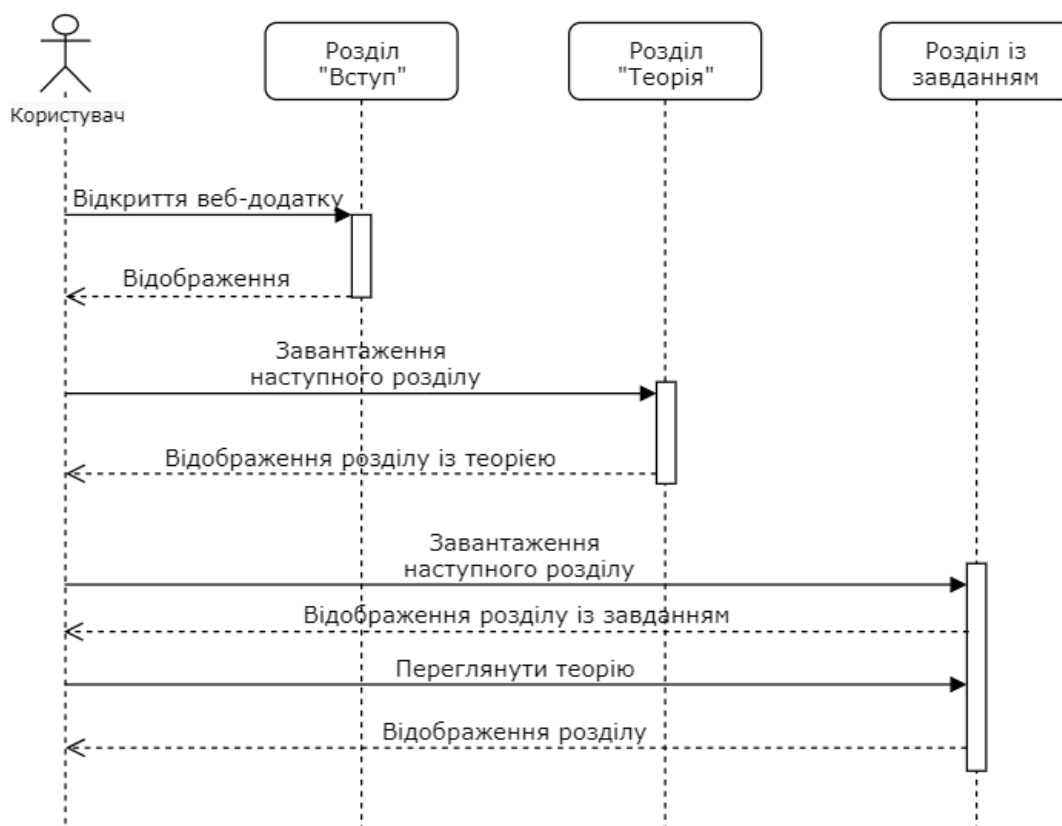


Рисунок 3.4 – Діаграма послідовності для варіанту використання «Перегляд теорії»

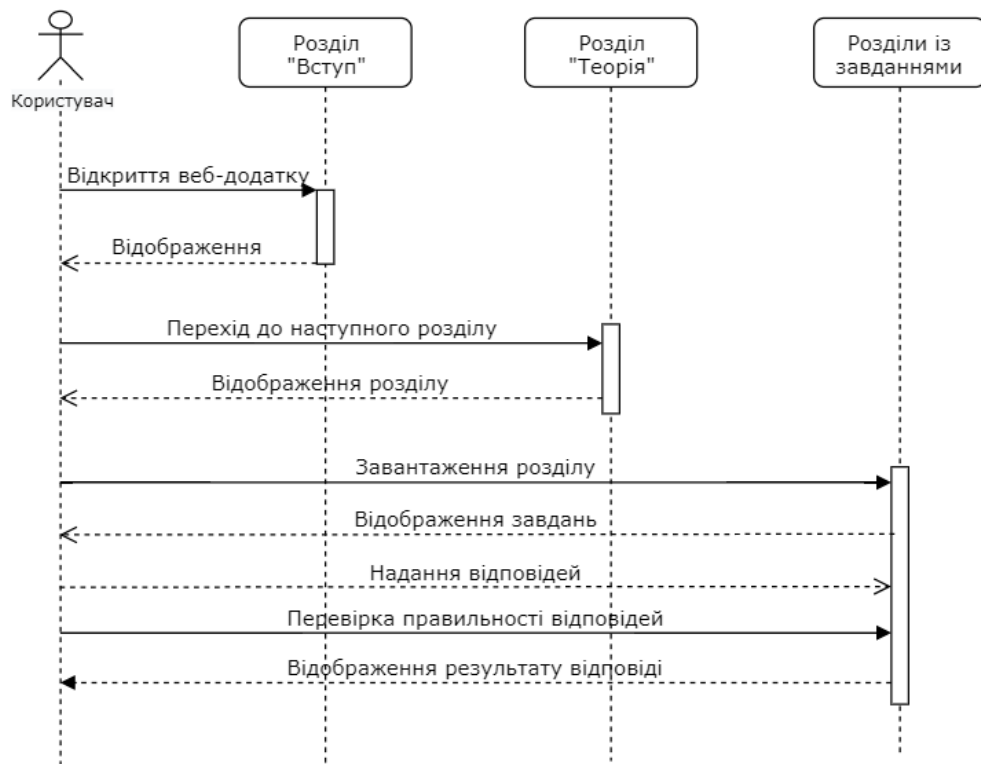


Рисунок 3.5 – Діаграма послідовності для варіанту використання «Виконання завдань»

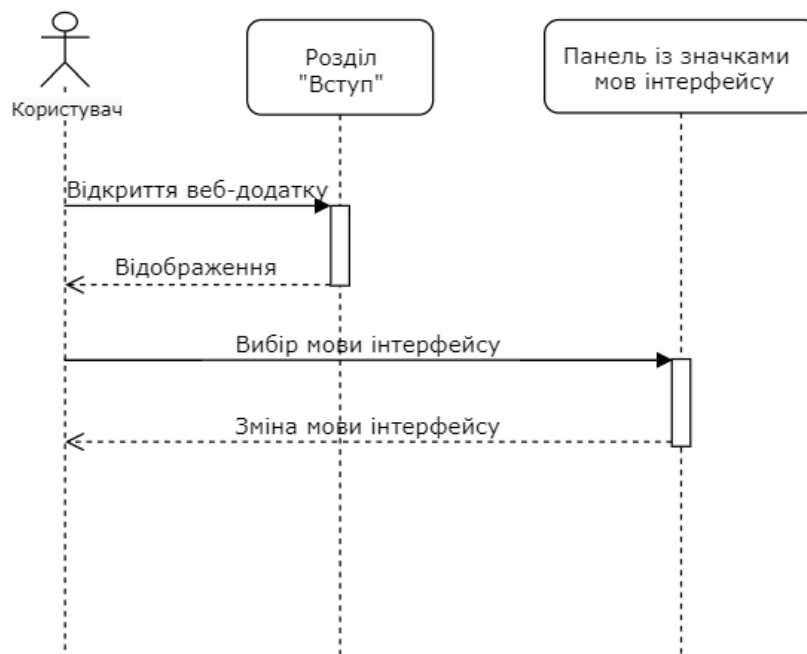


Рисунок 3.6 – Діаграма послідовності для варіанту використання «Зміна мови інтерфейсу»

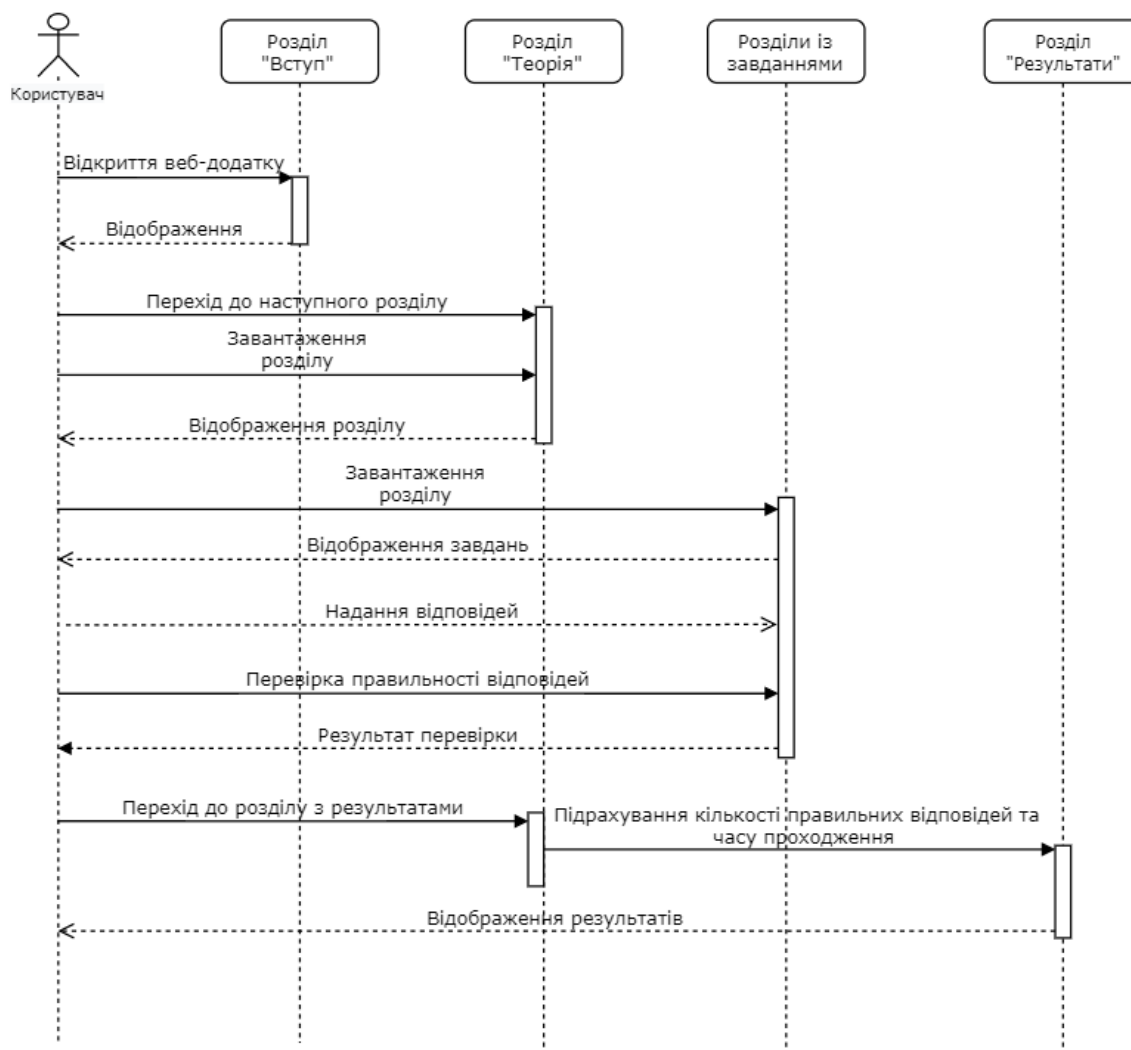


Рисунок 3.7 – Діаграма послідовності для варіанту використання «Отримання результатів проходження сеансу»

Отже, використання методологій UML та IDEF0 дозволило нам відобразити структуру веб-тренажерів. Їх перевагою є графічна нотація, яка наочно та зрозуміло відображає функціонування системи, а саме, у нашому випадку, варіанти використання системи та послідовність процесів в варіантах використання.

4 РОЗРОБКА КОМПЛЕКСУ ІНТЕРАКТИВНИХ ТРЕНАЖЕРІВ

4.1 Структура веб-додатку

Веб-тренажери дипломного проекту було створено на базі існуючого фреймворку тренажерів дистанційної освіти університету СумДУ. Уся розробка ведеться згідно обмежень цього фреймворку для успішної подальшої інтеграції веб-тренажерів в систему дистанційної освіти.

Шаблон для тренажерів містить локальну (урізану) версію серверного рушія тренажерів, функціоналу якої достатньо для розробки. Серверний рушій (повна версія) виконує запис результатів проходження завдань тренажеру студентом до бази даних. Також він надає механізм опції «Допомога викладача», що може бути використана студентом у разі виникнення будь-яких питань до завдань чи теорії. Надалі буде представлено сам шаблон для тренажерів, що надається для розробки.

Початкова сторінка шаблону містить логотип тренажеру, його назву та кнопки перемикавання мови локалізації інтерфейсу (рис. 4.1).

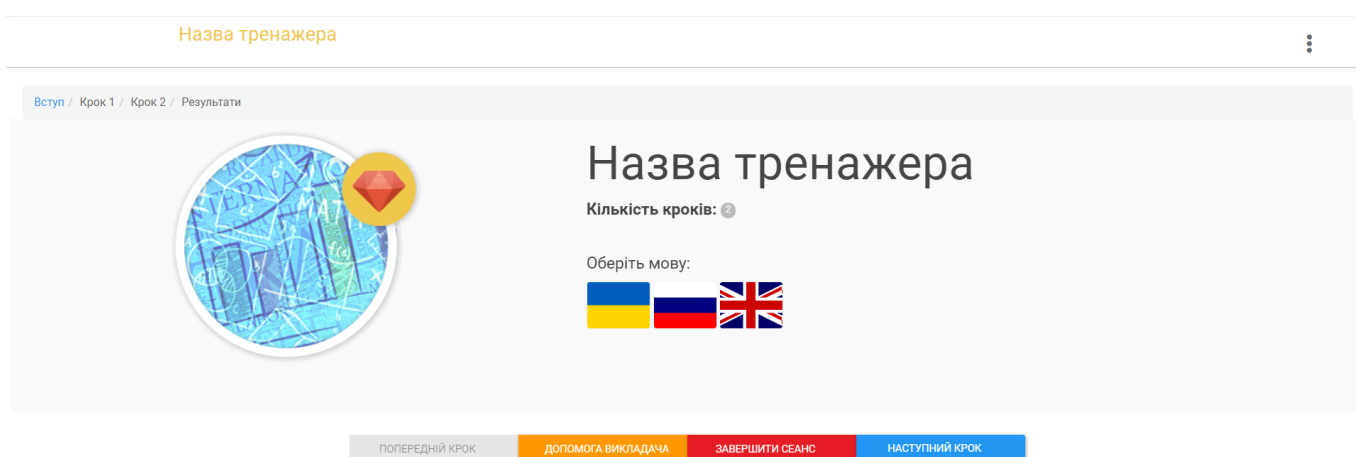


Рисунок 4.1 – Початкова сторінка шаблону веб-тренажеру

Друга сторінка шаблону містить в собі всі елементи керування, які можна використати в розробці завдань для тренажерів (рис. 4.2-4.3).

Назва тренажера

Вступ / [Крок 1](#) / Крок 2 / Результати

Перевірка елементів...

Оберіть коректний елемент зі списку:

Введіть в поле текст 'test' и 'Text' через кому:

Перетягніть відповіді 1 та 4:

Введіть числа 1 АБО 4:

plot $3x^2+2x+5$:

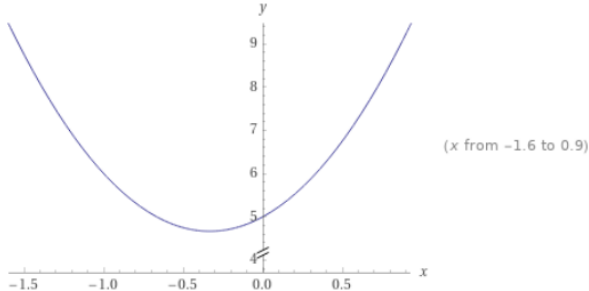


Рисунок 4.2 – Набір всіх можливих елементів для розробки завдань

Радіо №2
 Радіо №3
 Радіо №4 (правильний)
 Радіо №1

Введіть 4.5 або 4,5:

Правильна відповідь
 Неправильна відповідь

$G_{\mu\nu} = 8\pi G(T_{\mu\nu} + \rho\Lambda g_{\mu\nu})$

Рисунок 4.3 – Набір всіх можливих елементів для розробки завдань

Остання сторінка шаблону відображає результати виконання завдань (рис. 4.4).

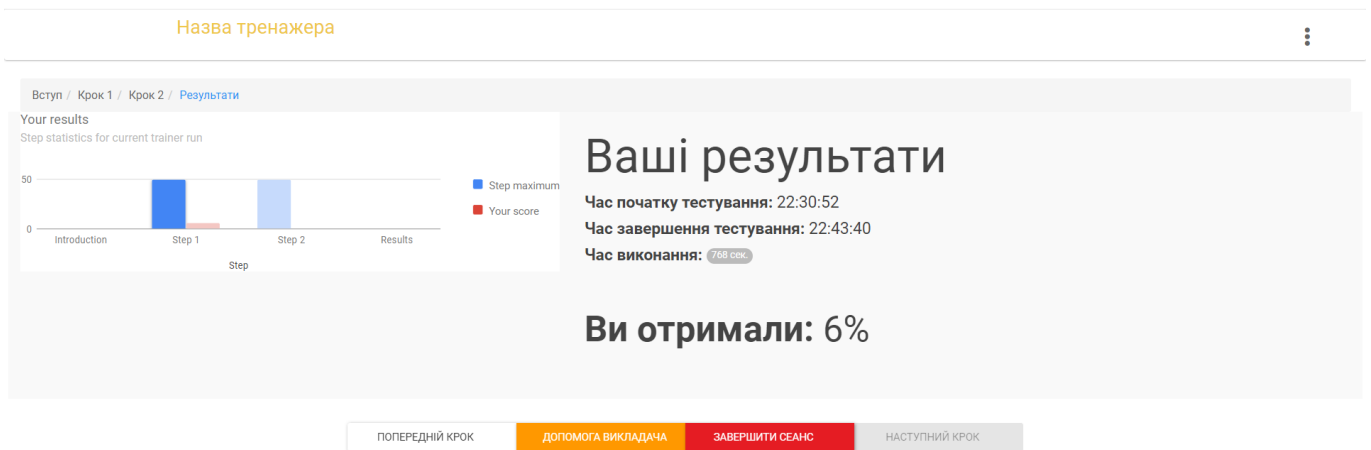


Рисунок 4.4 – Набір всіх можливих елементів для розробки завдань

Виходячи із вищезгаданого, можна умовно поділити структуру шаблону веб-тренажеру на 3 розділи: «Вступ» (надає функціонал вибору мови локалізації), «Завдання» (складається з кроків), «Результати» (відображає результати виконання завдань)

Механізми перевірки відповідей, виведення результатів, зміни мови інтерфейсу надає рушій тренажеру. Всі дані, якими оперує веб-тренажер, зберігаються локально у файлах формату json: файл конфігурації веб-тренажеру (задає варіант мови локалізації за замовчуванням при відкритті тренажеру), файл описання кроків (містить дані назви файлів кроків та максимальну кількість можливих балів за завдання), файли із текстовою інформацією для різних мов локалізації (текст підставляється замість змінних скриптом-шаблонізатором).

4.2 Розробка веб-тренажерів

Перед тим, як початку розробку веб-тренажерів, для кожного було створено логотипи в Adobe Illustrator, зображення яких відповідають тематикам тренажерів (рис. 4.5-4.7).

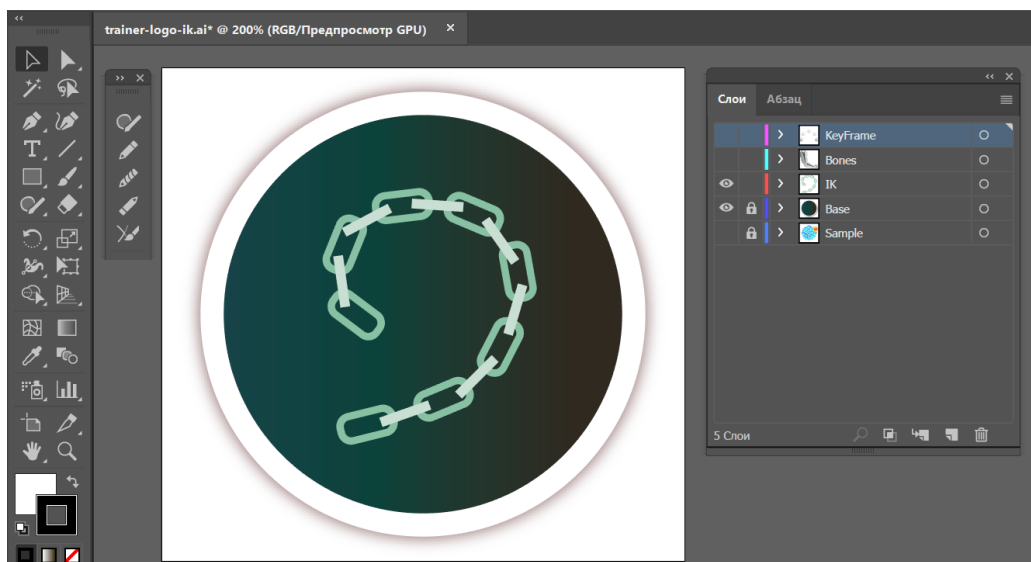


Рисунок 4.5 – Логотип тренажеру за темою «Кінематика пов'язаних об'єктів»



Рисунок 4.6 – Логотип тренажеру за темою «Анімація об'єктів Bones»

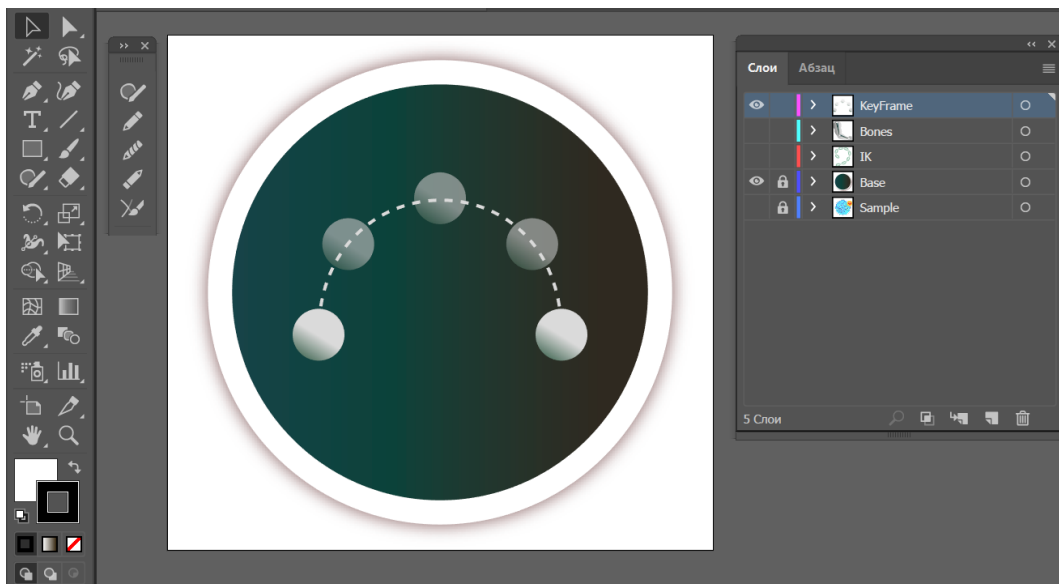


Рисунок 4.7 – Логотип тренажеру за темою «Анімація за ключовими кадрами»

Далі було створено теоретичний розділ для 1-го тренажеру («Кінематика пов'язаних об'єктів»). Бажання мати власний дизайн для веб-тренажерів спонукало використовувати власний набір стилів, тому теоретичний розділ та завдання було поміщено в округлені блоки.

Теоретичний розділ 1-го тренажеру виглядає наступним чином (рис. 4.8). Теоретичний блок містить вкладки, що перемикають контент, реалізовані із використання CSS-фреймворку Bootstrap. Інформація для теоретичного блоку збиралася, перероблялася та структурувалася із матеріалів практичних робіт дисципліни «Технології анімації та візуалізації» та мережі Інтернет.

Матеріали теоретичного розділу супроводжуються анімованими ілюстраціями, які відтворюють практичні дії в програмі 3ds Max та записані власне автором з екрану в gif-анімацію із використанням програми ScreenToGif [27] (рис. 4.9).

Ієрархія

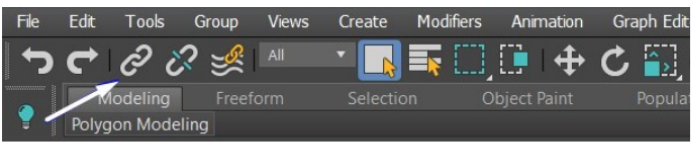
Пряма кінематика

Інверсна кінематика

Ієрархія

Ієрархія – це набір об'єктів зі зв'язками між ними за принципом «**предок-нащадок**» (інша назва: «**батьківський об'єкт – дочірній об'єкт**»). Предок контролює поведінку одного або більше нащадків. У той же час предки можуть контролюватися іншими предками, що мають більш високий рівень ієрархії. Нащадок, контрольований деяким предком, може виявитися предком для інших нащадків, що знаходяться за ієрархією нижче нього. Найвищий об'єкт ієрархії управляє всіма об'єктами ієрархії і називається **кореневим** або **Root-об'єктом**.

Ієрархія (зв'язок) об'єктів задається з використанням команди **Select and Link**, представленої на зображенні нижче



Об'єкти зв'язуються від нащадка до батька (батьком для всіх буде найперший об'єкт)

Рисунок 4.8 – Теоретичний блок 1-го тренажера

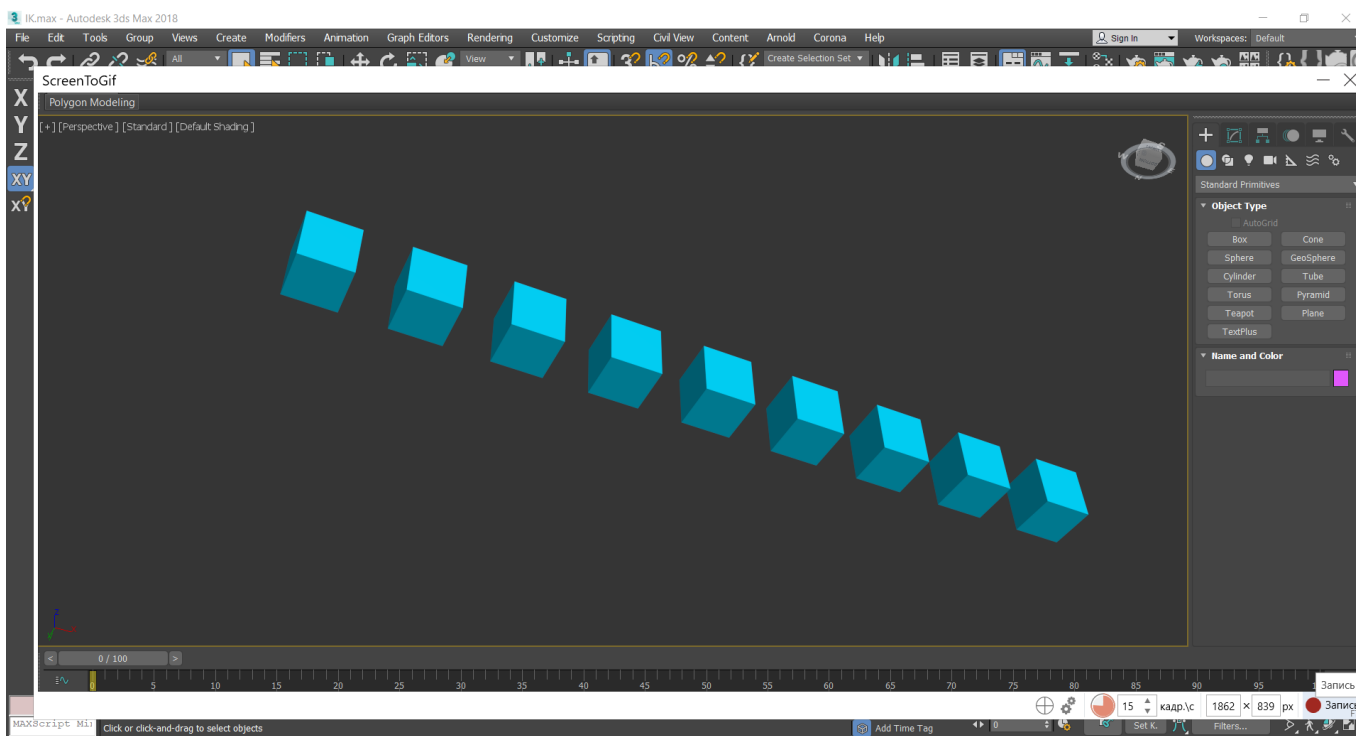


Рисунок 4.9 – Запис ілюстрацій теоретичного розділу в програмі 3ds Max та запис процесу в gif програмою ScreenToGif

Далі відбувалася розробка завдань практичної частини тренажеру. Оскільки тренажер містить лімітований набір можливих елементів вводу відповідей для завдань, було вирішено розширити функціонал додаванням механізму натискання на області зображень. Він працює наступним чином: до попереднього заготовленого зображення додається мапа (map), що містить сформовані за допомогою спеціального сайту зони (area) [28] (рис. 4.10-4.11), що повторюють форми об'єктів чи кнопок, потім зображення зі створеною мапою областей додається в блок завдання.

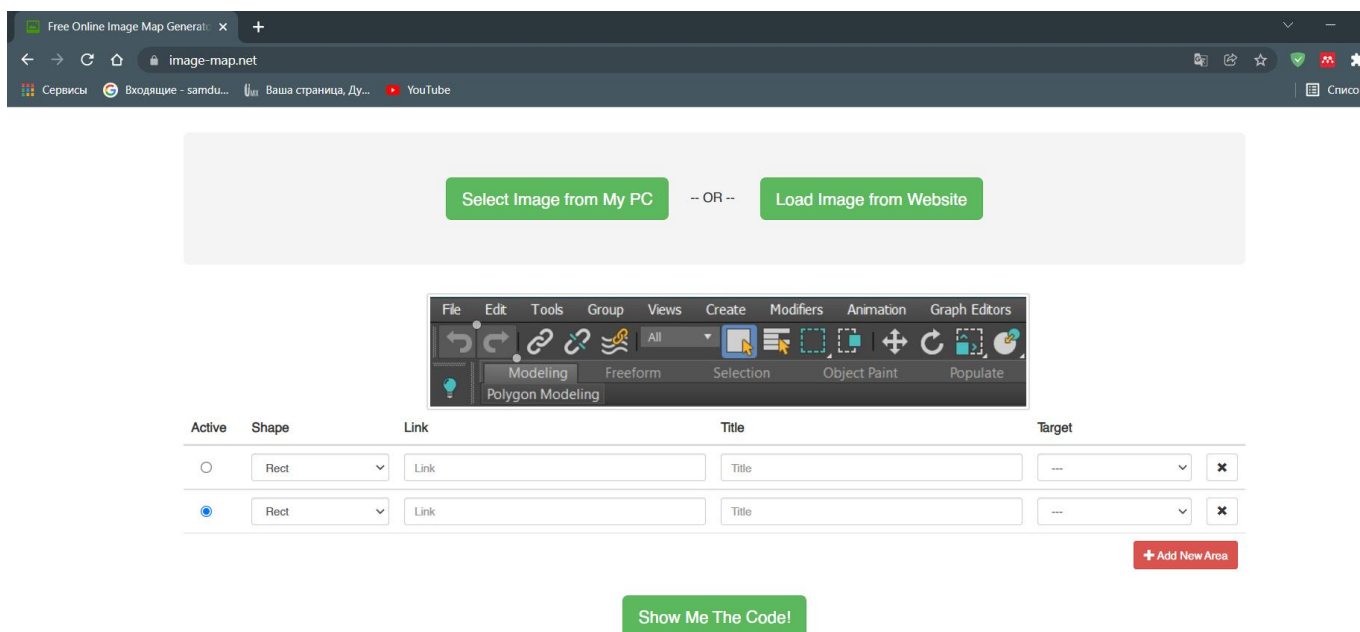


Рисунок 4.10 – Створення областей для зображення на сайті Free Online Image Map Generator

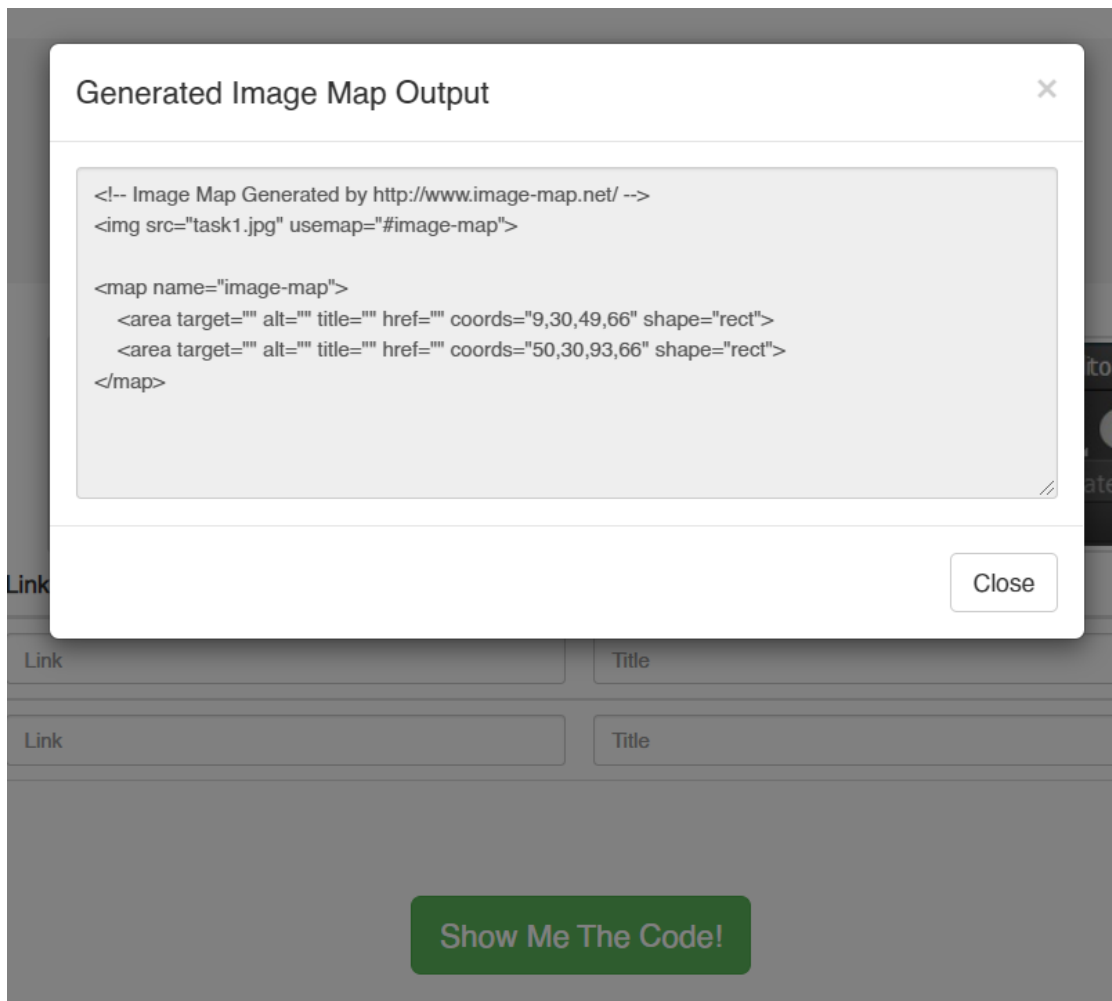


Рисунок 4.11 – Отримання коду для сформованих областей зображення

Оскільки все одно можна використовувати лише пропонувані фреймворком елементи вводу, тому що тренажер лише для них може виконувати перевірку, тому розмістимо під доданим зображення приховане `input` поле, в яке завдяки обробнику натискання на області зображення додаватиметься значення, що відповідатиме `id` області. Правильною відповіддю в такому випадку буде `id` правильної області.

Для елемента `area` неможна задати стилі відображення при наведенні, тому для цієї мети було використано скрипт `jquery.maphilight.js`, що дозволяє це зробити [29].

Оскільки `map` нестійкий до адаптивного дизайну або зміни масштабу сторінки, було використано скрипт `jquery.rwdImageMaps.js`, що переобчислює координати областей при зміні розширення екрану [30]. В результаті ми маємо завдання із інтерактивними областями, приклади деяких із них наведено на рисунках 4.12-4.13.

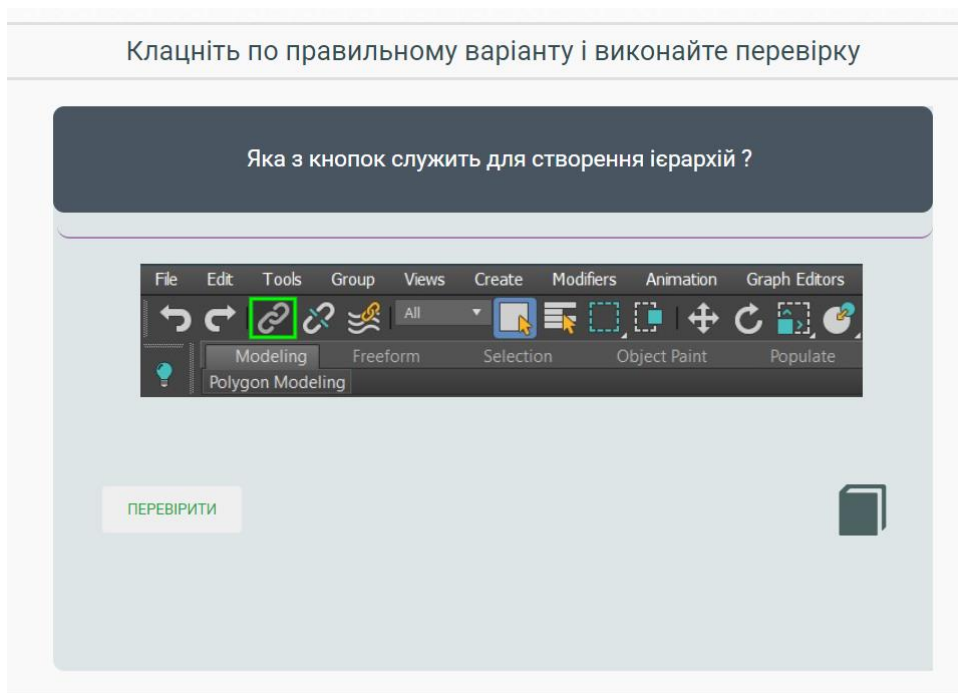


Рисунок 4.12 – Типи завдань із ітерактивними областями

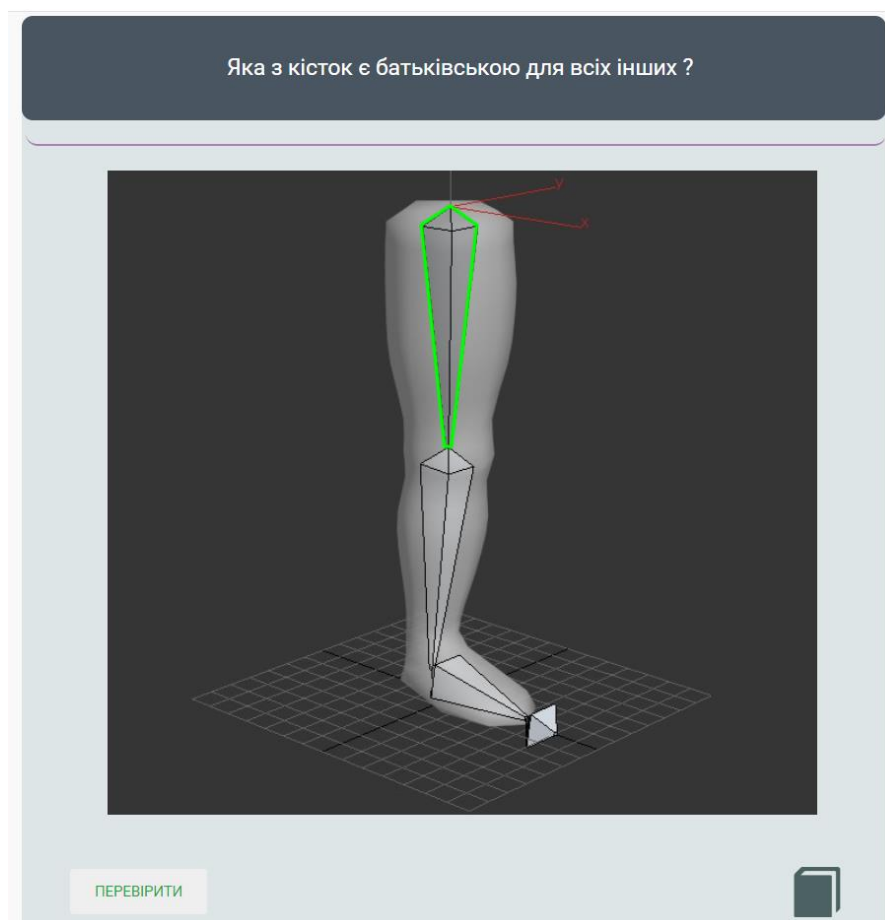


Рисунок 4.13 – Типи завдань із ітерактивними областями

Код зображення із мапою активних областей представлено нижче (рис. 4.14). Також представлено код, що запускає скрипт переобчислення координат area у випадку зміни масштабу сторінки, та обробник натискання на area, що додає id елемента area в приховане input поле (рис. 4.15).

```

  <area id="1" href="javascript:void(0);" coords="12,30,50,66" shape="rect">
  <area id="2" href="javascript:void(0);" coords="91,65,53,30" shape="rect">
  <area id="3" href="javascript:void(0);" coords="137,65,98,31" shape="rect">
  <area id="4" href="javascript:void(0);" coords="141,31,177,65" shape="rect">
  <area id="5" href="javascript:void(0);" coords="3,78,41,114" shape="rect">
  <area id="6" href="javascript:void(0);" coords="390,29,354,66" shape="rect">
  <area id="7" href="javascript:void(0);" coords="432,29,471,65" shape="rect">
  <area id="8" href="javascript:void(0);" coords="479,28,519,65" shape="rect">
</map>
<br>
{{{STEP1_INPUT}}}
```

Рисунок 4.14 – Мапа зображення з елементами area

```
<script>
  $(function() {
    $('img[usemap]').rwdImageMaps();
  });
</script>

<script>
  $('area').click(function() {
    $('input[name="step1-input"]').val(this.id);
  });
</script>
```

Рисунок 4.15 – Обробник натискання на area

Крім завдань з елементами типу radio кнопок, select, text input, в одному з тренажерів було використано типи завдань з елементами drag`n`dropable, що містять в собі зображення замість тексту (рис. 4.16-4.17).

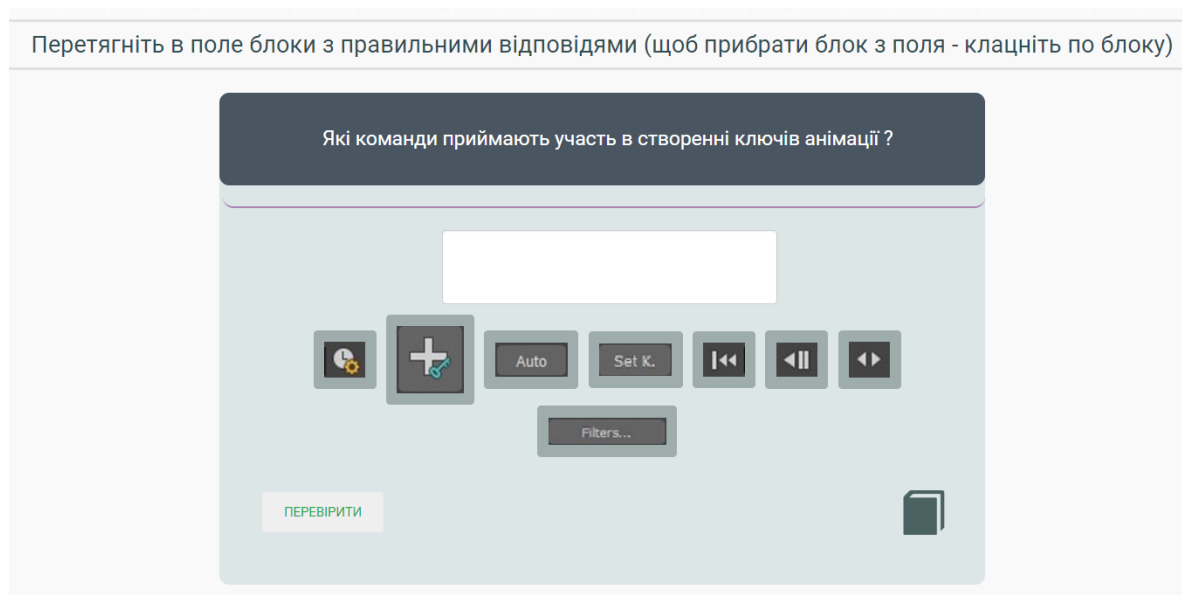


Рисунок 4.16 – Завдання з елементами drag`n`dropable

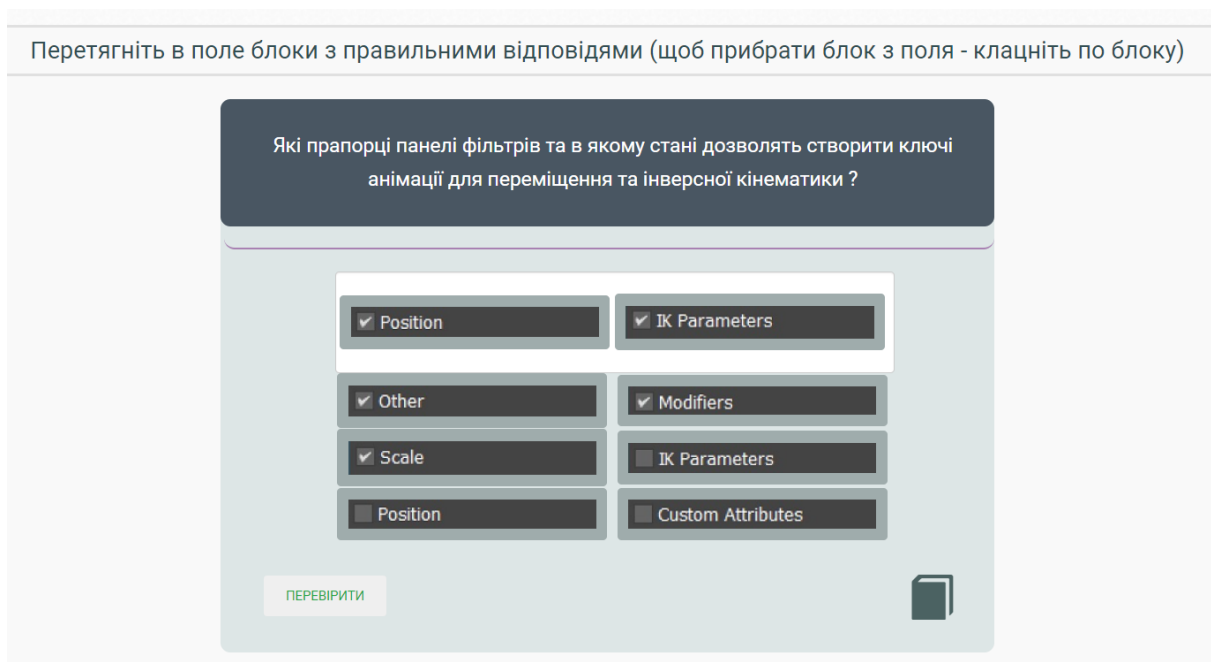


Рисунок 4.17 – Завдання з елементами drag`n`dropable

Перелік створених файлів розміток та скриптів для всіх завдань одного з тренажерів вилядає наступним чином (рис. 4.18-4.19) (завдання містить у назві слово «step»). Кожне завдання являє собою пару одного файла розмітки та відповідного скрипту.

step1	03.12.2021 23:35	Yandex Browser H...	3 КБ
step2	24.11.2021 22:53	Yandex Browser H...	3 КБ
step3	24.11.2021 22:53	Yandex Browser H...	3 КБ
step4	24.11.2021 22:53	Yandex Browser H...	3 КБ
step5	14.11.2021 21:56	Yandex Browser H...	2 КБ
step6	14.11.2021 21:57	Yandex Browser H...	2 КБ
step7	14.11.2021 21:59	Yandex Browser H...	3 КБ
step8	14.11.2021 21:59	Yandex Browser H...	2 КБ
step9	15.11.2021 0:06	Yandex Browser H...	2 КБ
step10	03.12.2021 23:58	Yandex Browser H...	3 КБ

Рисунок 4.18 – Перелік файлів розміток завдань одного з тренажерів

step1	04.12.2021 0:02	файл JavaScript	2 КБ
step2	04.12.2021 0:02	файл JavaScript	2 КБ
step3	04.12.2021 0:02	файл JavaScript	2 КБ
step4	04.12.2021 0:02	файл JavaScript	2 КБ
step5	04.12.2021 0:02	файл JavaScript	2 КБ
step6	04.12.2021 0:02	файл JavaScript	2 КБ
step7	04.12.2021 0:02	файл JavaScript	2 КБ
step8	04.12.2021 0:02	файл JavaScript	2 КБ
step9	04.12.2021 0:02	файл JavaScript	2 КБ
step10	04.12.2021 0:15	файл JavaScript	2 КБ

Рисунок 4.19 – Перелік файлів-скриптів завдань одного з тренажерів

При 1-й неправильній відповіді кількість можливих балів за завдання зменшується на 10 відсотків від можливої максимальної кількості балів за завдання. При неправильній відповіді відображається відповідне повідомлення (рис. 4.20). При неправильній відповіді 3 рази підряд проходження тренажеру припиняється та починається спочатку, супроводжуючись відповідним повідомленням (рис. 4.21).

Помилка! Спробуйте ще

Рисунок 4.20 – Повідомлення при неправильній відповіді на завдання

Ви погано засвоїли теоретичний матеріал. Повторіть його і почніть виконувати завдання з початку.

Рисунок 4.21 – Повідомлення при втраті всіх спроб відповіді

Після проходження всіх завдань відображається вікно результатів (рис. 4.22).

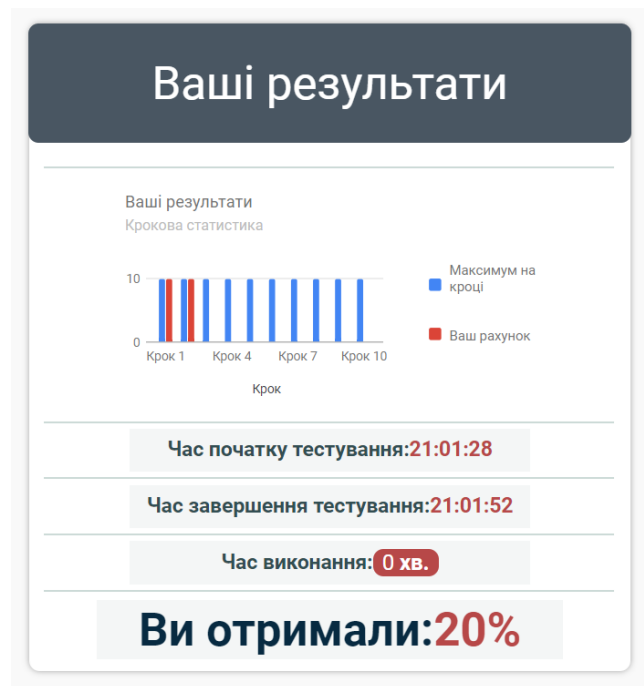


Рисунок 4.22 – Вікно результатів проходження завдань

Бали за завдання розподіляються у файлі конфігурації кроків *trainer.steps.json* (рис. 4.23). Сума всіх балів з файлу конфігурації кроків дорівнює 100.

```

1  [
2  {
3      "filename"    : "introduction",
4      "score"      : 0,
5      "hasScript"  : true
6  },
7  {
8      "filename"    : "theory",
9      "score"      : 0,
10     "hasScript"   : true
11 },
12 {
13     "filename"    : "step1",
14     "score"      : 10,
15     "hasScript"  : true
16 },
17 {
18     "filename"    : "step2",
19     "score"      : 10,
20     "hasScript"  : true
21 },

```

Рисунок 4.23 – Файл конфігурації кроків

Мова інтерфейсів за замовчуванням для всіх тренажерів встановлена як українська. Код файлів розміток завдань, супровідних скриптів завдань, конфігурації кроків та файли із текстовими даними тренажеру «Кінематика пов'язаних об'єктів» наведено в додатку Б.

4.3 Результат розробки комплексу веб-тренажерів

Вступ розроблених тренажерів починається з головної сторінки, що містить логотип та назву тренажеру (рис. 4.24).

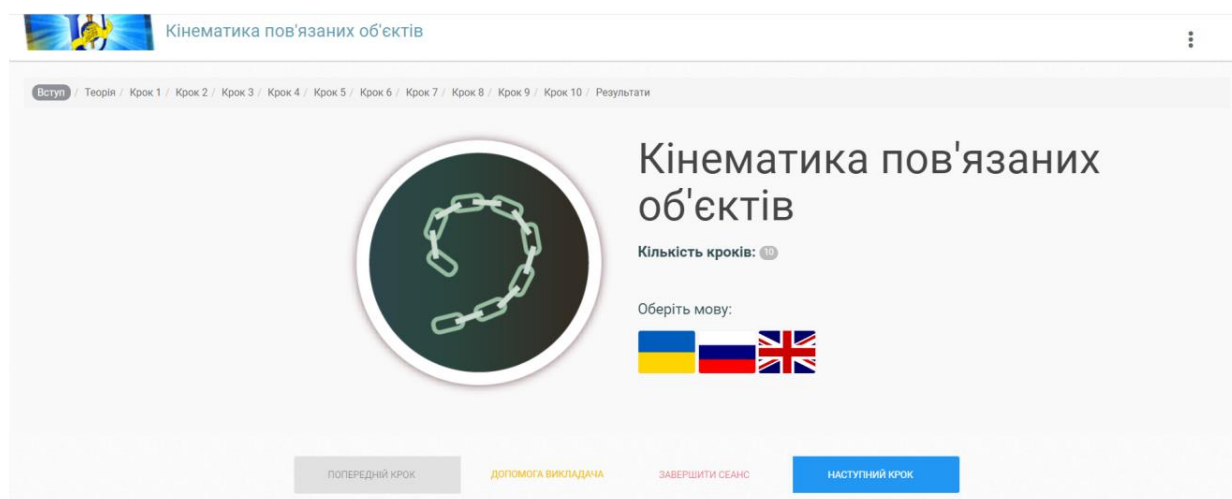


Рисунок 4.24 – Вступ тренажеру

Тут користувач може вибрати бажану мову локалізації інтерфейсу (рис. 4.25). За замовчанням встановлено українську мову.

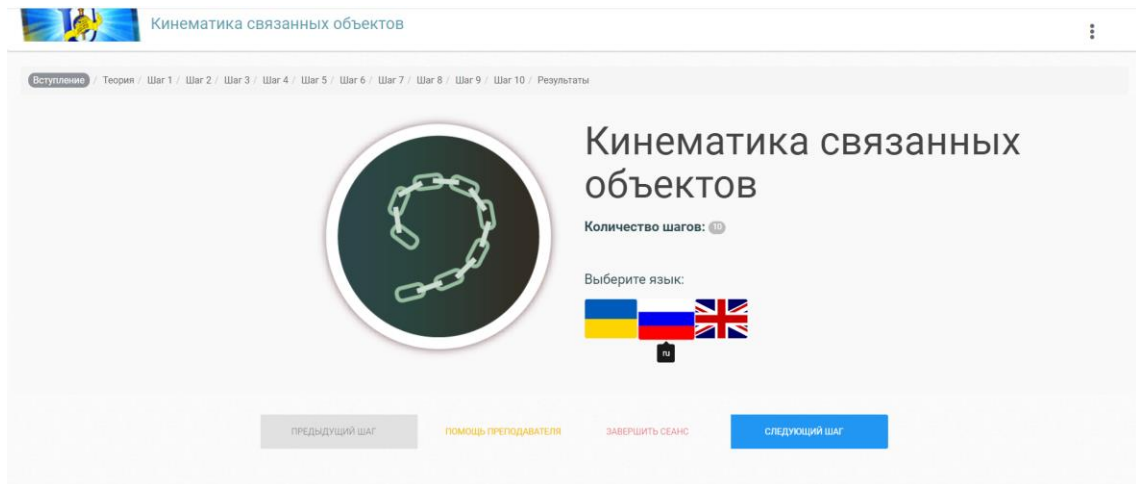


Рисунок 4.25 – Вибір мови локалізації

Далі йде теоретичний розділ, оформлений у вигляді тематичних вкладок, створений на основі теоретичного матеріалу дисципліни «Технології анімації та візуалізації» та інформації з мережі Інтернет (рис. 4.26).

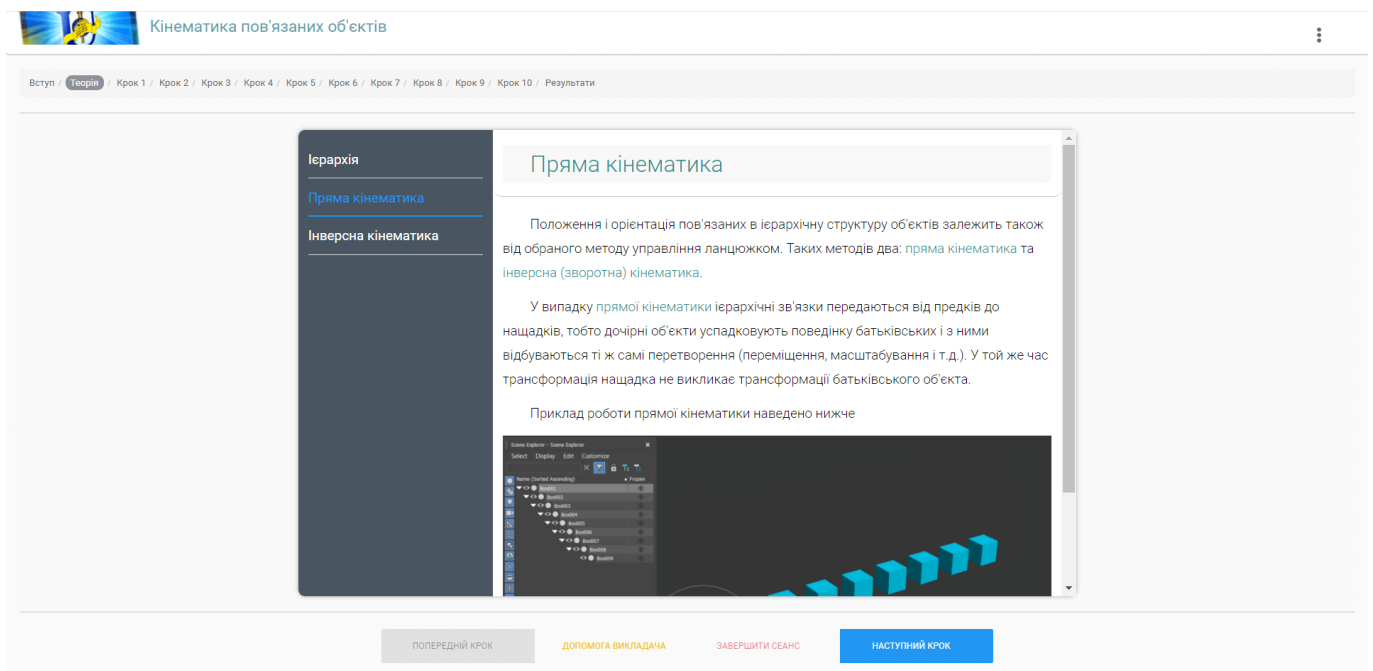


Рисунок 4.26 – Теоретичний розділ тренажерів

Після теоретичного розділу починаються завдання. Кожен тренажер містить тип завдань із інтерактивними зображеннями. У таких завданнях для відповіді слід натиснути на вірну область зображення (рис. 4.27).

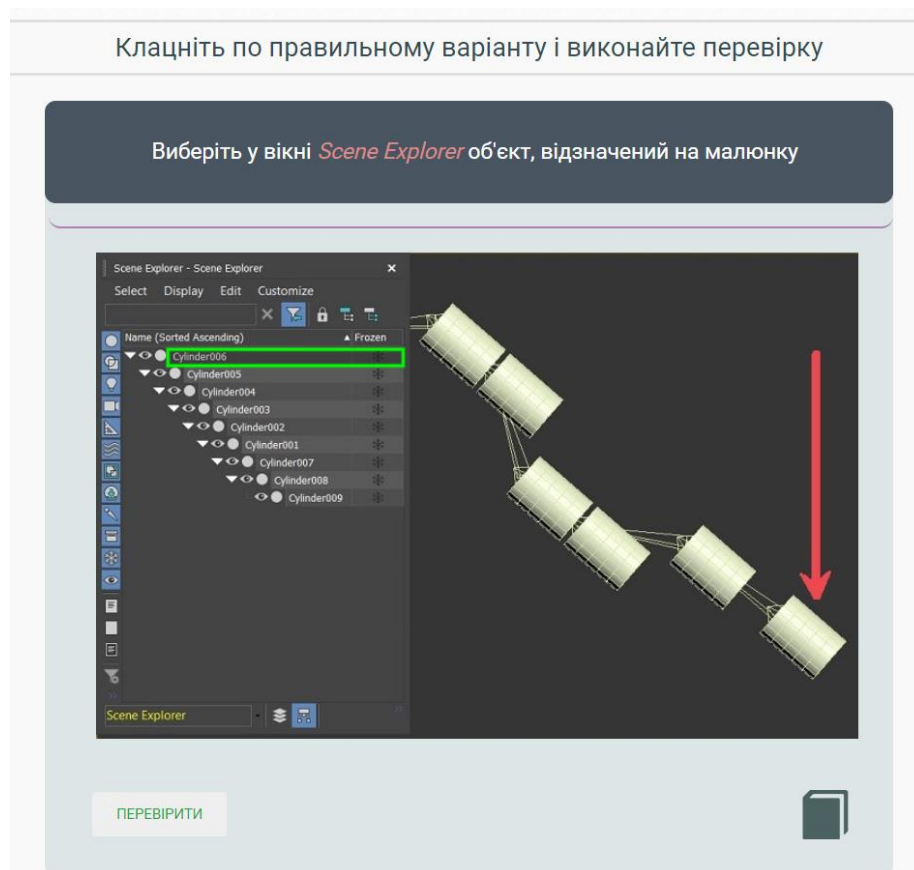


Рисунок 4.27 – Тип завдання із активними областями натискання

Підчас виконання завдань можна переміщатися до теоретичного розділу, натиснувши однойменну кнопку внизу сторінки (рис. 4.28).



Рисунок 4.28 – Переміщення до теоретичного розділу підчас виконання завдань

Після перегляду теоретичного розділу можна переміщатися назад до завдання у той же крок, де було викликано вікно теорії (рис. 4.29).

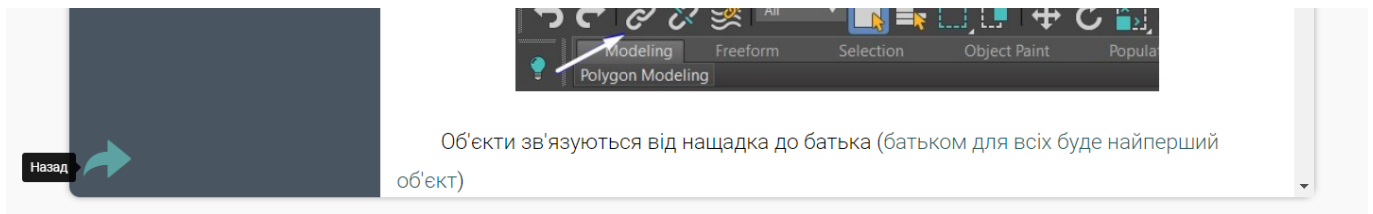


Рисунок 4.29 – Переміщення назад до завдання з теоретичного розділу

Крім завдань з активними областями зображень є тип завдань з введенням правильної відповіді в текстове поле (рис. 4.30).



Рисунок 4.30 – Тип завдань з текстовими полями

У тренажерах використано тип завдань із радіокнопками, які передбачають вибір тільки однієї правильної відповіді (рис. 4.31).

Виберіть правильну відповідь

Що таке *Ієрархія*?

- Відношення "Нащадок-Нащадок"
- Зв'язок "Батько-Батько"
- Відношення "Предок-Нащадок"

ПЕРЕВІРИТИ




Рисунок 4.31 – Тип завдань із радіокнопками

Далі приведено тип завдань із елементом `select` – випадаючим списком із варіантами відповідей (рис. 4.32).

Виберіть коректний елемент зі списку

Який тип кінематики представлений на малюнку

Інверсна кінематика

ПЕРЕВІРИТИ

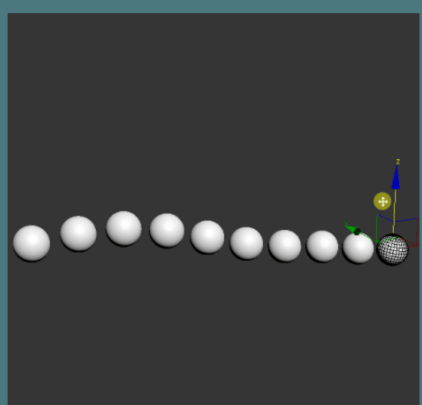



Рисунок 4.32 – Тип завдань із елементом `select`

Останнім типом завдань є завдання з елементами drag`n`draggable – блоки, що потрібно переміщати за допомогою мишки в область білого кольору для відповіді (рис. 4.33).

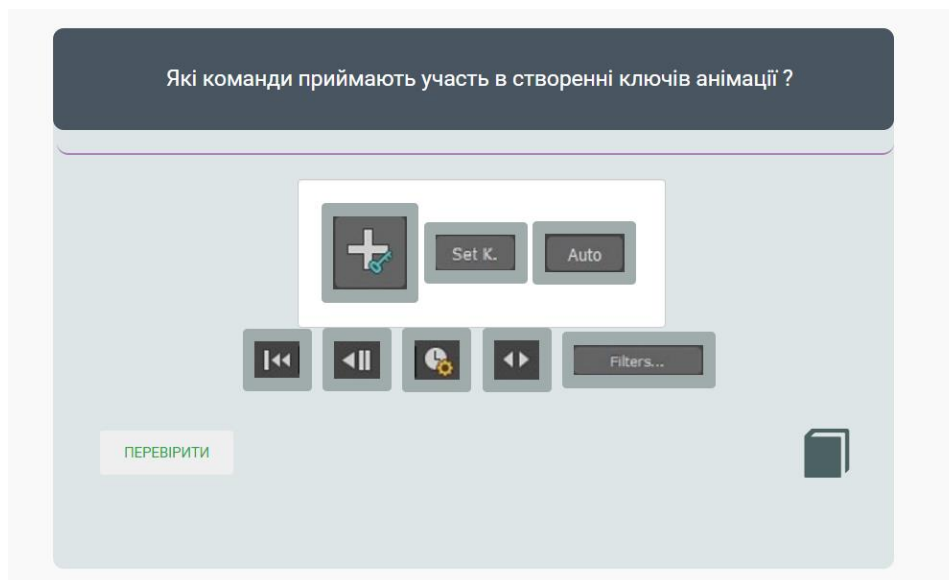


Рисунок 4.33 – Тип завдань з елементами drag`n`draggable

У кінці виконання тренажера, після виконання всіх завдань або після дострокового завершення сеансу опцією «Завершити сеанс», відображається вікно результатів (рис. 4.34).



Рисунок 4.34 – Розділ результатів проходження завдань

Більш детально повністю усі кроки комплексу веб-тренажерів наведені в додатку В.

В результаті виконання кваліфікаційної роботи розроблено 3 тренажери, двоє з яких складаються з 10-ти завдань, а третій – з 12-ти. Для всіх трьох тренажерів реалізовано 3 мови локалізації: українська, російська та англійська. Завдання всіх тренажерів покривають теоретичний матеріал відповідних теоретичних блоків.

ВИСНОВКИ

У ході розробки комплексу веб-тренажерів для вивчення технологій анімації в 3ds Max було сформульовано мету, досліджено актуальність майбутнього продукту. Огляд аналогів показав, що за обраною тематикою не існує тренажерів.

Для досягнення мети проекту вирішено наступні задачі: проведено аналіз предметної області розробки веб-тренажерів, створено теоретичний блок для кожного додатку на основі навчального теоретичного матеріалу, розроблено сценарії виконання завдань та самі завдання для веб-додатків, розроблено варіанти локалізації для інтерфейсу веб-тренажерів, проведено тестування веб-додатків.

Скелетом тренажерів став фреймворк тренажерів дистанційної освіти університету СумДУ.

Середовищем розробки стало IDE PhpStorm. Джерелом графічного контенту (скріншоти та gif-анімація) стала програма 3ds Max. Джерелом текстової інформації для тренажерів став теоретичний матеріал дисципліни «Технології анімації та візуалізації» та інформація з мережі Інтернет. Завдання тренажерів покривають теоретичний матеріал, поданий в теоретичних блоках.

Після успішної розробки комплексу веб-тренажерів їх було викладено на сервер дистанційної освіти університету СумДУ.

Найбільшу користь веб-тренажери приносять студентам у випадку виконання завдань та паралельної практики в програмі 3ds Max. Також в кінці проходження сеансу відображається результат виконання завдань студентом, що потім заноситься в базу даних сервером СумДУ та стає доступним для перегляду викладачем дисципліни.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Игровая индустрия за 2020 год в цифрах — инфографика от GamesIndustry.biz — Индустрия на DTF: URL: <https://dtf.ru/gameindustry/294571-igrovaya-industriya-za-2020-god-v-cifrah-infografika-ot-gamesindustry-biz>(дата звернення: 27.10.21).
2. Global Games Market to Generate \$175.8 Billion in 2021; Despite a Slight Decline, the Market Is on Track to Surpass \$200 Billion in 2023 | Newzoo: URL: <https://newzoo.com/insights/articles/global-games-market-to-generate-175-8-billion-in-2021-despite-a-slight-decline-the-market-is-on-track-to-surpass-200-billion-in-2023/>(дата звернення: 27.10.21).
3. Newzoo: в 2021-м выручка игровой индустрии немного сократится, но в 2023-м она достигнет \$204,6 млрд - ИТС.ua: URL: https://itc.ua/news/newzoo-v-2021-m-vyruchka-igrovoj-industrii-nemnogo-sokratitsya-no-v-2023-m-ona-dostignet-2046-mlrd/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+itc-ua+%28ИТС.ua%29(дата звернення: 28.10.21).
4. Newzoo: в 2020 году доход игрового рынка составит почти \$160 млрд | App2top: URL: https://app2top.ru/?p=168011&utm_source=yxnews&utm_medium=desktop(дата звернення: 28.10.21).
5. Самые популярные профессии в игровой индустрии | GDJob.PRO: URL: <https://gdjob.pro/stati/soiskatelyam/professii-v-igrovoju-industrii/>(дата звернення: 27.10.21).
6. 3D-аниматор: кто такой, чем занимается, востребованность, зарплата, работа, обучение: URL: <https://professii-online.ru/3d-animator/>(дата звернення: 27.10.21).
7. Программы для 3d анимации - 3DRadar: URL: <https://3dradar.ru/post/47797/>(дата звернення: 27.10.21).
8. Лучшая программа для 3D моделирования: URL: <https://videosmile.ru/post/897-luchshaya-programma-dlya-3d-modelirovaniya>(дата звернення: 27.10.21).

9. Maliha, S. G., Diaz-Siso, J. R., Plana, N. M., та ін. Haptic, Physical, and Web-Based Simulators: Are They Underused in Maxillofacial Surgery Training? *Journal of Oral and Maxillofacial Surgery*. 2018. Vol. 76, No. 11. С. 2424.e1-2424.e11.

10. Juszczuk, D., Tarnawski, J., Karla, T., та ін. Real-Time Basic Principles Nuclear Reactor Simulator Based on Client-Server Network Architecture with WebBrowser as User Interface. *Advances in Intelligent Systems and Computing*. 2017. Vol. 577. С. 344–353.

11. Schneider, S. L., Council, M. L. Distance learning in the era of COVID-19. *Archives of Dermatological Research* 2020 313:5. 2020. Vol. 313, No. 5. С. 389–390.

12. Sadeghi, M. A Shift from Classroom to Distance Learning: Advantages and Limitations. *International Journal of Research in English Education*. 2019. Vol. 4, No. 1. С. 80–88.

13. Сайт платформа для изучения GB английского языка для взрослых и детей | Бесплатное обучение в онлайн школе Puzzle English: URL: <https://puzzle-english.com/>(дата звернення: 28.10.21).

14. Учить английский язык онлайн — бесплатные онлайн-курсы английского: URL: <https://www.englishdom.com/online-free/>(дата звернення: 28.10.21).

15. Free Grammar Checker Online – Ginger Software: URL: <https://www.gingersoftware.com/grammarcheck>(дата звернення: 28.10.21).

16. CodeCombat - Игры с программированием для изучения Python и Java Script | CodeCombat: URL: <https://codecombat.com/>(дата звернення: 28.10.21).

17. Ісаєв, В.В. Комплекс інтерактивних тренажерів для дистанційного курсу з вивчення 3ds Max [Текст]: робота на здобуття кваліфікаційного рівня бакалавра; спец.: 122 - комп'ютерні науки / В.В. Ісаєв; наук. кер. І.В. Баранова. – Суми: СумДУ, 2021. – 78 с.

18. 15 Best Web Development IDE in 2021[Updated]: URL: <https://hackr.io/blog/web-development-ide>(дата звернення: 01.12.21).

19. The best vector editors in 2021 | Creative Bloq: URL: <https://www.creativebloq.com/buying-guides/best-vector-editors>(дата звернення: 01.12.21).

20. The Use of IDEF0 for the Design and Specification of Methodologies: URL: https://www.researchgate.net/publication/2447898_The_Use_of_IDEF0_for_the_Design_and_Specification_of_Methodologies(дата звернення: 22.11.21).

21. Перевод стандарта IDEF0 на русском языке / Хабр: URL: <https://habr.com/ru/post/515466/>(дата звернення: 20.11.21).

22. CA ERWIN PROCESS MODELER IS A POWERFUL MODELING TOOL THAT HELPS YOU VISUALIZE, ANALYZE AND IMPROVE COMPLEX BUSINESS PROCESSES, SO YOU CAN INCREASE YOUR UNDERSTANDING OF THE ENTERPRISE AND CREATE COMPETITIVE ADVANTAGE: URL: https://www.danysoft.com/estaticos/free/ERwin_PM.pdf(дата звернення: 22.11.21).

23. Зачем нам UML? Или как сохранить себе нервы и время / Хабр: URL: <https://habr.com/ru/post/458680/>(дата звернення: 20.11.21).

24. UML Distilled: A Brief Guide to the Standard Object Modeling Language - Martin Fowler - Google книги: URL: https://books.google.com.ua/books?hl=uk&lr=&id=nHZslSr1gJAC&oi=fnd&pg=PR21&q=uml&ots=Va4YFRNz4J&sig=f9eqinnoF-fA1Vy0Jy4Jt_XGM2A&redir_esc=y#v=onepage&q=uml&f=false(дата звернення: 20.11.21).

25. Использование диаграммы вариантов использования UML при проектировании программного обеспечения / Хабр: URL: <https://habr.com/ru/post/566218/>(дата звернення: 20.11.21).

26. Уточняем описание функций системы с помощью диаграммы Sequence / Хабр: URL: <https://habr.com/ru/post/450076/>(дата звернення: 20.11.21).

27. ScreenToGif - Record your screen, edit and save as a gif, video or other formats: URL: <https://www.screentogif.com/>(дата звернення: 01.12.21).

28. Free Online Image Map Generator: URL: <https://www.image-map.net/>(дата звернення: 02.12.21).

29. jQuery maphilight documentation: URL: <https://projects.davidlynch.org/maphilight/docs/>(дата звернення: 02.12.21).

30. GitHub - stowball/jquery-rwdImageMaps: Responsive Image Maps jQuery Plugin: URL: <https://github.com/stowball/jquery-rwdImageMaps>(дата звернення: 02.12.21).

31. SMART Goals - Time Management Training From MindTools.com: URL: <https://www.mindtools.com/pages/article/smart-goals.htm>(дата звернення: 22.11.21).

32. What is a Work Breakdown Structure (WBS) | Project Management: URL: <https://www.workbreakdownstructure.com/>(дата звернення: 22.11.21).

33. Organization Breakdown Structure (OBS) - Professional Services Automation: URL: <https://uplandsoftware.com/psa/resources/glossary/organization-breakdown-structure-obs/>(дата звернення: 22.11.21).

34. Что такое диаграмма Ганта? | Atlassian: URL: <https://www.atlassian.com/ru/agile/project-management/gantt-chart>(дата звернення: 22.11.21).

Додаток А

ПЛАНУВАННЯ РОБІТ

Деталізація мети проекту методом SMART. Мета проекту – розробити декілька продуктів, кожен із яких матиме вигляд веб-тренажеру, для вивчення основ створення анімації в 3ds Max для дистанційного курсу «Технології анімації та візуалізації» за темами: «Кінематика пов'язаних об'єктів», «Анімація об'єктів Bones», «Анімація за ключовими кадрами». Продукти будуть у нагоді в процесі дистанційного навчання, поліпшуватимуть процес навчання завдяки наявності інтерактивної взаємодії слухача із тренажерами, допомагатимуть в процесі виконання практичних завдань, закріплюватимуть отриманні знання та перевірятимуть їх якість. Для чіткого визначення цілей та задач використаємо методологію SMART [31]. Результати деталізації методом SMART розміщені у табл. А.1.

Таблиця А.1 – Деталізація мети методом SMART

Specific (конкретна)	Створити комплекс веб-тренажерів для вивчення та перевірки якості засвоєння теорії із основ створення анімації в 3ds Max за темами: «Кінематика пов'язаних об'єктів», «Анімація об'єктів Bones», «Анімація за ключовими кадрами».
Measurable (вимірювана)	Результатом роботи проекту є ухвалення керівником (викладачем дистанційного курсу «Технології анімації та візуалізації» за сумісництвом) результату виконаної роботи та успішне розміщення серії продуктів на сервері дистанційного навчання університету СумДУ.
Achievable (досяжна)	Створення серії продуктів здійснюється із використанням браузерів Yandex та Google Chrome, IDE

	PhpStorm, локального серверу Open Server, фреймворку для веб-тренажерів СумДУ, 3ds Max
Relevant (реалістична)	Розробка веб-тренажерів вимагає базових знань мови розмітки HTML, мови каскадних стилів CSS, мови програмування JavaScript, фреймворку Bootstrap для пришвидшення верстання. Розробник володіє всіма необхідними знаннями, тому виконання поставлених задач є досяжним.
Time-framed (обмежена у часі)	Для успішної реалізації продуктів вся робота повинна бути виконана в строках календарного плану.

Планування змісту структури робіт. Планування змісту структури робіт представлено WBS-діаграмою. Вона наглядно демонструє декомпозицію робіт на елементарні одиниці, послідовність їх виконання та сам зміст робіт [32]. Діаграма WBS зображена на рис. А.1.



Рисунок А.1 – WBS-структура робіт проекту

Планування структури організації, для впровадження готового проекту (OBS). Внутрішня організаційна структура виконання робіт графічно представлена за допомогою OBS [33]. Діаграма OBS зображена на рис. А.2.

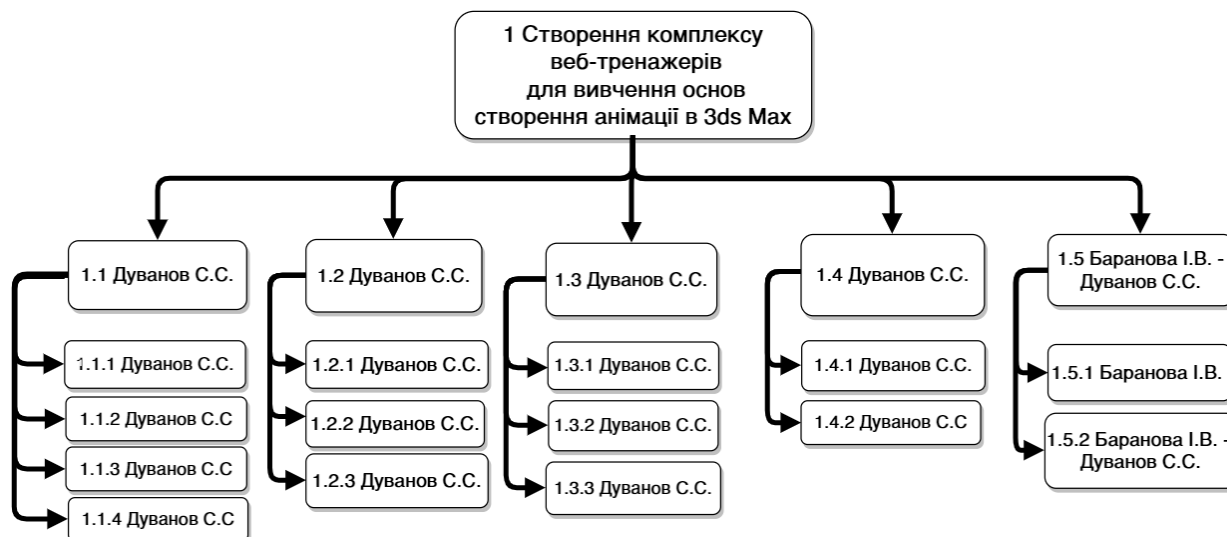


Рисунок А.2 – Організаційна структура проекту (OBS)

Діаграма Ганта. Календарний план виконання робіт графічно представлено діаграмою Ганта, який ілюструє тривалість виконання робіт у залежності з датою початку виконання, розташовуючи роботи у хронологічному порядку [34]. В середньому на один день роботи приділяється 3-4 години часу. Діаграму Ганта представлено на рис. А.3-А.5.

№	Задачі	Дата початку	Дата кінця	Кількість днів
1	Створення комплексу веб-тренажерів для вивчення основ створення анімації в 3ds Max	02.11.2021	04.12.2021	32
1.1	Планування	02.11.2021	09.11.2021	7
1.1.1	Розробка WBS-структури	02.11.2021	03.11.2021	1
1.1.2	Розробка OBS-структури	03.11.2021	04.11.2021	1
1.1.3	Оцінка ризиків	04.11.2021	06.11.2021	2
1.1.4	Розробка UML-діаграм	07.11.2021	09.11.2021	2
1.2	Розробка веб-інтерфейсів	10.11.2021	16.11.2021	6
1.2.1	Підбір гармонійних кольорів для інтерфейсів	10.11.2021	12.11.2021	2
1.2.2	Створення макетів	10.11.2021	12.11.2021	2
1.2.3	Верстання	12.11.2021	16.11.2021	4
1.3	Розробка веб-додатків	16.11.2021	27.11.2021	11
1.3.1	Додавання розділів із теорією	16.11.2021	20.11.2021	4
1.3.2	Створення завдань	20.11.2021	24.11.2021	4
1.3.3	Впровадження різних варіантів локалізації інтерфейсів	24.11.2021	27.11.2021	3
1.4	Тестування	27.11.2021	30.11.2021	3
1.4.1	Виправлення багів	27.11.2021	30.11.2021	3
1.4.2	Виправлення недоліків розмітки	27.11.2021	30.11.2021	3
1.5	Розгортання веб-додатків	30.11.2021	04.12.2021	4
1.5.1	Розміщення веб-додатків на хостингу дистанційного курсу	30.11.2021	01.12.2021	1
1.5.2	Завершення оформлення документації	30.11.2021	04.12.2021	4

Рисунок А.3 – Список робіт

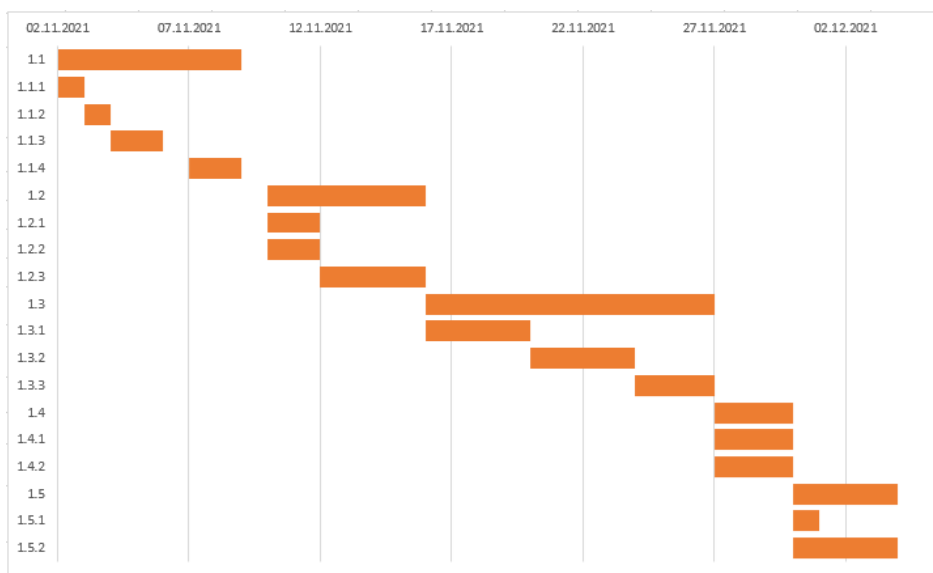


Рисунок А.4 – Діаграма Ганта

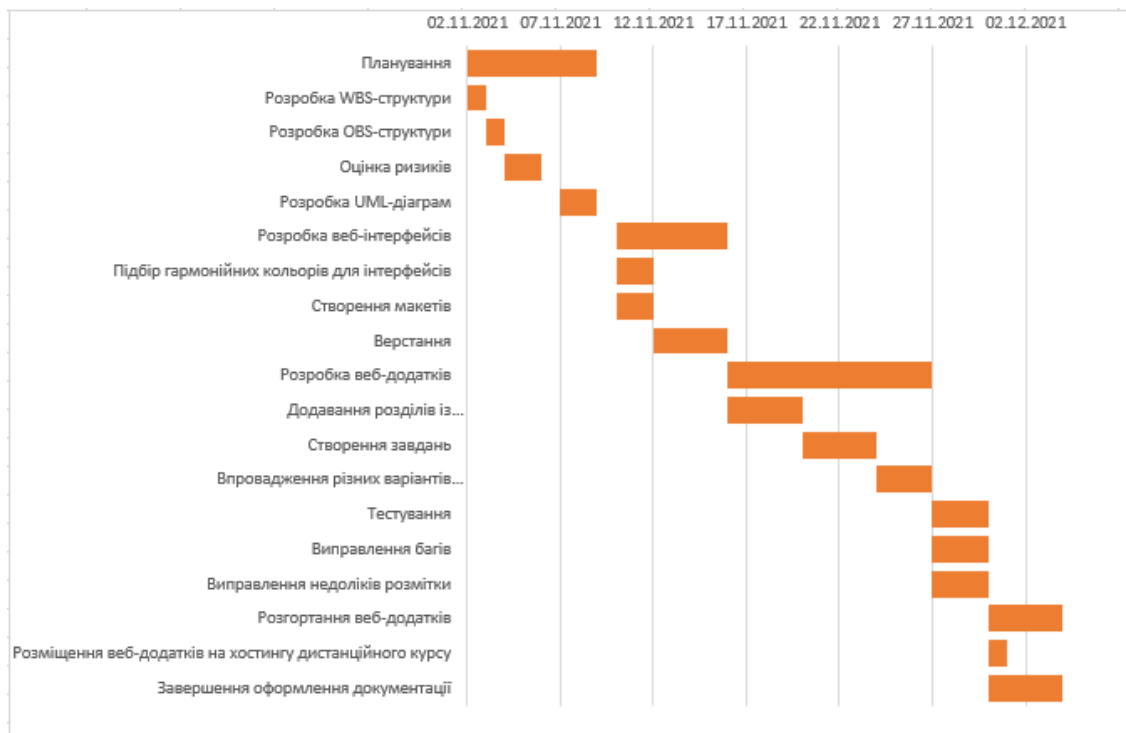


Рисунок А.5 – Діаграма Ганта

Управління ризиками. Основними ризиками розробки комплексу інтерактивних тренажерів для вивчення технологій анімації в 3ds Max є:

- невірна оцінка часу розробки під час планування;
- збільшення вимог до функціональності продуктів під час розробки;
- виникнення складнощів у ході комунікації з керівником;
- недостатня компетентність розробника;
- відмова обладнання, на якому ведеться розробка;
- поява великої кількості помилок у роботі додатків під час здачі в експлуатацію;
- форс-мажорні ситуації із розробником.

Для оцінки ризиків створено таблицю (табл. А.2), де ймовірність виникнення ділиться на наступні типи: слабоймовірний, малоймовірний, ймовірний, дуже ймовірний, можливий (ранжування йде від 1-го до 5-ти відповідно). Втрати ж діляться на: мінімальні, низькі, середні, високі, максимальні (ранжування так само йде від 1-го до 5-ти відповідно).

Таблиця А.2. Ймовірність виникнення і величина ризику

№	Ризики	Ймовірність	Вплив
R1	Невірна оцінка часу розробки під час планування	3	5
R2	Збільшення вимог до функціональності продуктів під час розробки	3	3
R3	Виникнення складнощів у ході комунікації з керівником	2	4
R4	Недостатня компетентність розробника	2	5
R5	Відмова обладнання, на якому ведеться розробка	2	5
R6	Поява великої кількості помилок у роботі додатків під час здачі в експлуатацію	3	4
R7	Форс-мажорні ситуації із розробником	3	5

Далі створено матрицю ймовірності виникнення та впливу ризиків у загальному вигляді (рис. А.6) та матрицю (рис. А.7) на основі таблиці табл. А.2, де ігноровані ризики помічені світло-зеленим, незначні – зеленим, помірні – жовтим, істотні – червоним, критичні – темно-червоним.

Ймовірність виникнення	5	5	10	15	20	25
	4	4	8	12	16	20
	3	3	6	9	12	15
	2	2	4	6	8	10
	1	1	2	3	4	5
		1	2	3	4	5
		Вплив ризику				

Рисунок А.6 – Матриця ймовірності виникнення та впливу ризиків у загальному вигляді

Ймовірність виникнення	5					
	4					
	3			R2	R6	R1, R7
	2				R3	R4, R5
	1					
		1	2	3	4	5
		Вплив ризику				

Рисунок А.7 – Матриця ймовірності виникнення та впливу ризиків

На основі матриці було виділено 3 істотних ризики:

- невірна оцінка часу розробки під час планування (R1);
- поява великої кількості помилок у роботі додатків під час здачі в експлуатацію (R6);
- форс-мажорні ситуації із розробником (R7);

Перший ризик можна мінімізувати шляхом ранньої переоцінки календарного плану. Другий – шляхом постійного тестування упродовж процесу розробки. Третій – лише шляхом більшої обачності розробника.

Додаток Б

ЛІСТИНГ КОДУ ТРЕНАЖЕРУ «КІНЕМАТИКА ПОВ'ЯЗАНИХ ОБ'ЄКТІВ»

trainer.html

```

<!DOCTYPE html>
<html>
<head>
  <title>jTrainer :: {{ TRAINER_NAME }}</title>
  <meta charset="UTF-8"/>
  <meta name="description" content=""/>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta http-equiv="Cache-Control" content="private">
  <meta http-equiv="Cache-Control" content="no-cache">

  <script>
    var startTime = (new Date()).getTime();
  </script>

  <script src="/cdn/javascripts/jtrainer/current/js/jquery.min.js"></script>

  <script>
    $(window).load(function () {
      var endTime = (new Date()).getTime();
      var secondsLoading = ((endTime - startTime) / 1000).toFixed(4);
      $('<span>.page-loading</span>').append(secondsLoading + ' seconds.');
```

```

    <div><a class="navbar-brand" href="#" onclick="setTimeout(function() {window.location.reload();}, 200)"><span
class="trainer-brand"></span>{{ TRAINER_NAME }}</a></div>
</div>
<div id="navbar" class="navbar-collapse collapse">
  <div class="navbar-right info">
    <div class="dropdown">
      <button type="button" class="btn btn-link dropdown-toggle" id="menu" data-toggle="dropdown"
        aria-haspopup="true" aria-expanded="true" aria-label="Option Vertical"><span
        class="glyphicon glyphicon-option-vertical" aria-hidden="true"></span></button>
      <ul class="dropdown-menu" aria-labelledby="menu">
        <li><a href="#" data-toggle="modal" data-target="#aboutModal">About</a></li>
        <li><a href="#" class="page-loading text-muted"><span class="is-faded">Page generated in </span></a>
        </li>
      </ul>
    </div>
    <div class="modal fade" id="aboutModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">
      <div class="modal-dialog" role="document">
        <div class="modal-content">
          <div class="modal-header">
            <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
              aria-hidden="true">&times;</span></button>
            <h4 class="modal-title" id="aboutModalLabel">About</h4>
          </div>
          <div id="aboutBody" class="modal-body">
            ...
          </div>
          <div class="modal-footer">
            <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="modal modal-static fade" id="cogwheel-modal" role="dialog" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-body">
        <div class="text-center">
          
          <h4><span id="cogwheel-desc">Processing</span>... </h4>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="text-center animated bounceInUp">
  <div class="btn-group btn-group-md rotators">
    <a id="prevController" href="javascript:void(0)" class="btn btn-default col-xs-6 col-md-3" >{{ PREV_STEP }}</a>
    <a class="teacher-help btn btn-outline-warning col-xs-6 col-md-3" data-toggle="modal"
      data-target="#helpModal">{{ TEACHER_HELP }}</a>
    <a id="endTrainer" class="btn btn-outline-danger col-xs-6 col-md-3" data-toggle="modal" data-
target="#endModal">{{ END_TRAINER }}</a>
    <a id="begin" class="btn btn-outline-danger col-xs-6 col-md-3"
      onclick="window.location.reload(true)">{{ BEGIN_TRAINER }}</a>
    <a id="nextController" href="javascript:void(0)" class="btn btn-default col-xs-6 col-md-3">{{ NEXT_STEP }}</a>
  </div>

```

```

</div>

<!-- This is modal popup which shows up after click on "Teacher Help" button -->
<div class="modal fade" id="helpModal" tabindex="-1" role="dialog" aria-labelledby="helpModalLabel">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-label="Close" id="closeButton"><span
          aria-hidden="true">&times;</span></button>
        <h4 class="modal-title" id="helpModalLabel">{{ TEACHER_HELP }}</h4>
      </div>
      <div class="modal-body">
        <label class="control-label">{{ ENTER_HELP_TEXT }}</label>
        <input type="text" class="form-control" id="helpInput"/>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default" data-dismiss="modal">{{ HELP_REQUEST_CLOSE }}</button>
        <button type="button" class="btn btn-primary"
onclick="Service.getHelpModal()">{{ HELP_REQUEST_SEND }}
      </button>
      </div>
    </div>
  </div>
</div>

<!-- This is modal popup which shows up after click on "End trainer" button -->
<div class="modal fade" id="endModal" tabindex="-1" role="dialog" aria-labelledby="endModalLabel">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-label="Close" id="closeButton"><span
          aria-hidden="true">&times;</span></button>
        <h4 class="modal-title" id="endModalLabel">{{ END_TRAINER }}</h4>
      </div>
      <div class="modal-body">
        <label class="control-label">{{ END_ARE_YOU_SURE }}</label>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default" data-dismiss="modal">{{ END_TRAINER_CLOSE }}</button>
        <button type="button" class="btn btn-primary" onclick="Service.pushResultsEarly()"
          >{{ END_TRAINER_SEND }}
      </button>
      </div>
    </div>
  </div>
</div>

<div class="progress trainer-progress-bar">
  <div class="progress-bar"></div>
</div>

<div class="alert alert-dismissible alert-success validation-alert validation-alert-success">
<!-- <button type="button" class="close" data-dismiss="alert">×</button>-->
  <strong>{{ ALERT_WELL_DONE }}</strong>{{ ALERT_STEP_SUCCESS }}
</div>

<div class="alert alert-dismissible alert-danger validation-alert validation-alert-danger">
  <button type="button" class="close" data-dismiss="alert" onclick="window.location.reload(true)">×</button>
  {{ ALERT_STEP_FAIL }}
</div>
<div class="under-alert-danger"></div>

<div class="under-alert-danger"></div>

```

```

<div class="alert alert-dismissible alert-danger validation-alert validation-alert-middle">
  {{ALERT_FAIL}}
</div>
</body>
</html>

<script type="text/javascript" src="/cdn/javascripts/jtrainer/current/js/additions/html2canvas.js"></script>

```

introduction.html

```

<div class="row jumbotron start">
  <div class="col-xs-6 animated zoomIn">
    
  </div>
  <div class="col-xs-6 animated fadeInDown">
    <h1>{{TRAINER_NAME}}</h1>
    <p>
      <h5><strong>{{STEP_COUNT}}</strong> <span class="badge">{{STEPS_COUNT}}</span></h5>
    <br/>
    <h5>{{CHOOSE_LANG}}</h5>
    <div class="flag-icon flag-icon-ua" data-toggle="tooltip" data-placement="bottom" title="ua" id="ua"></div>
    <div class="flag-icon flag-icon-ru" data-toggle="tooltip" data-placement="bottom" title="ru" id="ru"></div>
    <div class="flag-icon flag-icon-en" data-toggle="tooltip" data-placement="bottom" title="en" id="en"></div>
  </p>
</div>
</div>

<script>
$(document).ready(function(){
  $('[data-toggle="tooltip"]').tooltip();
});
</script>

```

introduction.js

```

var introduction = function () {
  this.preDispatch = function () {
    Scorer.start();
  };

  this.postDispatch = function () {
    $('div.flag-icon').each(function() {
      $(this).attr('onclick', 'window.location.href = \'?lang=\' + $(this).attr("id");
    });

    Rotator.disablePrevButton();
    Rotator.disableEndButton();
    $('#a#prevController').attr('ondblclick', 'return 0');
  };

  this.mustache = function () {
    return {
      STEPS_COUNT: Rotator.getStepsCount()-3
    }
  }
}

```

```

}
};

```

theory.html

```

<div class="jumbotron jumbotronT">
  <hr>

<div class="container theory_block">
  <div class="row">
    <div class="col-lg-12 col-xs-12 block4 shadow">
      <div class="row">
        <div class="col dec theory">

          <!-- Navigation Buttons -->
          <div class="col-lg-3 col-xs-3 Tcolumn1">
            <ul class="nav nav-tabs nav-stacked" id="myTabs">
              <li class="active"><a href="#Hierarchy" data-toggle="pill">{{THEORY_COLUMN1}}</a></li>
              <li><a href="#DK" data-toggle="pill">{{THEORY_COLUMN2}}</a></li>
              <li><a href="#IK" data-toggle="pill">{{THEORY_COLUMN3}}</a></li>
            </ul>

            <!-- Return From Theory Button -->
            <br>
            <a href="javascript:void(0)" class="return-style" onclick="Rotator.switchStep(Rotator.lastLoadedStepId())"
data-toggle="tooltip" data-placement="left" title="{{FROM_THEORY}}">
              <span class="glyphicon glyphicon-share-alt" aria-hidden="true"></span>
            </a>
          </div>

          <!-- Content -->
          <div class="col-lg-9 col-xs-9 Tcolumn2">
            <div class="tab-content">

              <div class="tab-pane active" id="Hierarchy">
                <h3>{{TH1_HEAD}}</h3>
                <hr>
                <p style="text-indent: 5%;">{{TH1_TEXT1}}</p>
                <p style="text-indent: 5%;">{{TH1_TEXT2}}</p>
                
                <br>
                <p style="text-indent: 5%;">{{TH1_TEXT3}}</p>
                
                <br>
                <p style="text-indent: 5%;">{{TH1_TEXT4}}</p>
                
                <br>
                <p style="text-indent: 5%;">{{TH1_TEXT5}}</p>
                
                <br>
                <p style="text-indent: 5%;">{{TH1_TEXT6}}</p>
                
              </div>

              <div class="tab-pane" id="DK">
                <h3>{{TH2_HEAD}}</h3>
                <hr>
                <p style="text-indent: 5%;">{{TH2_TEXT1}}</p>
                <p style="text-indent: 5%;">{{TH2_TEXT2}}</p>

```



```

};

this.mustache = function () {
  return {

  }
}
};

```

step1.html

```

<div class="jumbotron jumbotronT page1">
  <hr>
  <h4 style="text-align: center;line-height: 0.0;" >{{TASK1_TITLE}}</h4>
  <hr>

  <div class="container task">
    <div class="row">
      <div class="col-lg-8 col-md-10 col-sm-12 col-lg-offset-2 col-md-offset-1 block0">
        <div class="row">
          <div class="col dec">
            <h5>
              <div class="task-dec rounded-4 text-center">
                {{{TASK1}}}
              </div>
            <hr>
            
            <map name="image-map1">
              <area id="1" href="javascript:void(0);" coords="12,30,50,66" shape="rect">
              <area id="2" href="javascript:void(0);" coords="91,65,53,30" shape="rect">
              <area id="3" href="javascript:void(0);" coords="137,65,98,31" shape="rect">
              <area id="4" href="javascript:void(0);" coords="141,31,177,65" shape="rect">
              <area id="5" href="javascript:void(0);" coords="3,78,41,114" shape="rect">
              <area id="6" href="javascript:void(0);" coords="390,29,354,66" shape="rect">
              <area id="7" href="javascript:void(0);" coords="432,29,471,65" shape="rect">
              <area id="8" href="javascript:void(0);" coords="479,28,519,65" shape="rect">
            </map>
            {{{STEP1_INPUT}}}
          </h5>
        </div>
      </div>
    </div>

    <div class="row">
      <div class="col-lg-6 col-xs-6 pull-left">
        <button class="btn btn-outline-success check">{{CHECK}}</button>
      </div>
      <div class="col-xs-6 pull-right">
        <a href="javascript:void(0)" class="book-style" onclick="Rotator.switchStep(1)" data-toggle="tooltip" data-
placement="right" title="{{TO_THEORY}}">
          <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
        </a>
      </div>
    </div>
  </div>
</div>
<hr>
</div>

```



```

<script>
  $(document).ready(function(){
    $('[data-toggle="tooltip"]').tooltip();
  });
</script>

<script>
  $(function() {
    $('img[usemap]').rwdImageMaps();
  });
</script>

<script>
  $('area').click(function() {
    $('input[name="step1-input"]').val(this.id);
  });
</script>

```

step1.js

```

var VStep1;// A variable for future validator
var step1 = function () {

  this.preDispatch = function () {

  };

  this.postDispatch = function () {
    VStep1 = new Validator();

    VStep1
      .addValidator($('input[name="step1-input"]'), 3)
      .setStrictMode(true)
      .setIgnoreCase(false)
      .enableStepFinishAlert(true)
      .setPenalty(Rotator.getStepScore(2)*0.1);

    Rotator.enablePrevButton();
    Rotator.enableEndButton();
    enableTheoryLinks();

    setTimeout(showReturnFromTheoryButton, 1000);
    enableTheoryLinks();

    $('button.check').click(function () {
      VStep1.setAttemptsOnCheckButton($(this)); //dynamically changing amount of attempts left on check button

      if(VStep1.validate() === true) // validate the validators
      {
        Rotator.disablePrevButton();
        Rotator.disableEndButton();
        disableTheoryLinks();
      }
      else
      {
        if(VStep1.getAttempts() != 0)
        {
          $('div.validation-alert-middle').fadeIn();
          setTimeout(() => {$('div.validation-alert-middle').fadeOut()}, 1200);
        }
      }
    });
  };
};

```

```

    }
  }
});
};

this.mustache = function () {
  return {
    STEP1_INPUT: new TextInput('step1-input')
      .render()
  }
}
};

```

step2.html

```

<div class="jumbotron jumbotronT page2">
  <hr>
  <h4 style="text-align: center;line-height: 0.0;" >{{TASK2_TITLE}}</h4>
  <hr>

<div class="container task">
  <div class="row">
    <div class="col-lg-8 col-md-10 col-sm-12 col-lg-offset-2 col-md-offset-1 block0">
      <div class="row">
        <div class="col dec">
          <h5>
            <div class="task-dec rounded-4 text-center">
              {{{TASK2}}}
            </div>
            <hr>
            
            <map name="image-map2">
              <area id="1" href="javascript:void(0);" coords="53,64,150,140" shape="rect">
              <area id="2" href="javascript:void(0);" coords="156,157,257,228" shape="rect">
              <area id="3" href="javascript:void(0);" coords="262,159,360,229" shape="rect">
              <area id="4" href="javascript:void(0);" coords="367,157,465,227" shape="rect">
              <area id="5" href="javascript:void(0);" coords="471,157,570,229" shape="rect">
              <area id="6" href="javascript:void(0);" coords="575,155,674,229" shape="rect">
            </map>
            {{{STEP2_INPUT}}}
          </h5>
        </div>
      </div>
    </div>

    <div class="row">
      <div class="col-lg-6 col-xs-6 pull-left">
        <button class="btn btn-outline-success check">{{CHECK}}</button>
      </div>
      <div class="col-xs-6 pull-right">
        <a href="javascript:void(0)" class="book-style" onclick="Rotator.switchStep(1)" data-toggle="tooltip" data-
placement="right" title="{{TO_THEORY}}">
          <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
        </a>
      </div>
    </div>
  </div>
  <br>
</div>
</div>

```

```

<hr>
</div>

<script>
$(document).ready(function(){
  $('[data-toggle="tooltip"]').tooltip();
});
</script>

<script>
$(function() {
  $('img[usemap]').rwdImageMaps();
});
</script>

<script>
$('area').click(function() {
  $('input[name="step2-input"]').val(this.id);
});
</script>

```

step2.js

```

var VStep2;
var step2 = function () {

  this.preDispatch = function () {

  };

  this.postDispatch = function () {
    VStep2 = new Validator();

    VStep2
      .addValidator($('input[name="step2-input"]'), 6)
      .setStrictMode(true)
      .setIgnoreCase(false)
      .enableStepFinishAlert(true)
      .setPenalty(Rotator.getStepScore(3)*0.1);

    Rotator.enablePrevButton();
    Rotator.enableEndButton();

    enableTheoryLinks();

    $('button.check').click(function () {
      VStep2.setAttemptsOnCheckButton($(this));

      if(VStep2.validate() === true)
      {
        Rotator.disablePrevButton();
        Rotator.disableEndButton();
        disableTheoryLinks();
      }
      else
      {
        if(VStep2.getAttempts() != 0)
        {
          $('div.validation-alert-middle').fadeIn();
          setTimeout(() => {$('div.validation-alert-middle').fadeOut()}, 1200);
        }
      }
    });
  };
};

```

```

    }
  }
});
};

this.mustache = function () {
  return {
    STEP2_INPUT: new TextInput('step2-input')
      .render()
  }
}
};

```

step3.html

```

<div class="jumbotron jumbotronT page3">
  <hr>
  <h4 style="text-align: center;line-height: 0.0;" >{{TASK3_TITLE}}</h4>
  <hr>

<div class="container task">
  <div class="row">
    <div class="col-lg-8 col-md-10 col-sm-12 col-lg-offset-2 col-md-offset-1 block0">
      <div class="row">
        <div class="col dec">
          <h5>
            <div class="task-dec rounded-4 text-center">
              {{{TASK3}}}
            </div>
            <hr>
            
            <map name="image-map3">
              <area id="1" href="javascript:void(0);" coords="66,89,282,105" shape="rect">
              <area id="2" href="javascript:void(0);" coords="80,105,282,122" shape="rect">
              <area id="3" href="javascript:void(0);" coords="94,123,282,138" shape="rect">
              <area id="4" href="javascript:void(0);" coords="107,139,282,154" shape="rect">
              <area id="5" href="javascript:void(0);" coords="119,155,282,170" shape="rect">
              <area id="6" href="javascript:void(0);" coords="134,171,282,187" shape="rect">
              <area id="7" href="javascript:void(0);" coords="146,188,282,203" shape="rect">
              <area id="8" href="javascript:void(0);" coords="160,204,282,219" shape="rect">
              <area id="9" href="javascript:void(0);" coords="173,220,282,236" shape="rect">
            </map>
            {{{STEP3_INPUT}}}
          </h5>
        </div>
      </div>
    </div>

    <div class="row">
      <div class="col-lg-6 col-xs-6 pull-left">
        <button class="btn btn-outline-success check">{{CHECK}}</button>
      </div>
      <div class="col-xs-6 pull-right">
        <a href="javascript:void(0)" class="book-style" onclick="Rotator.switchStep(1)" data-toggle="tooltip" data-
placement="right" title="{{TO_THEORY}}">
          <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
        </a>
      </div>
    </div>
  </div>
  <br>

```

```

    </div>
  </div>
</div>
<hr>
</div>

<script>
  $(document).ready(function(){
    $('[data-toggle="tooltip"]').tooltip();
  });
</script>

<script>
  $(function() {
    $('img[usemap]').rwdImageMaps();
  });
</script>

<script>
  $('area').click(function() {
    $('input[name="step3-input"]').val(this.id);
  });
</script>

```

step3.js

```

var VStep3;
var step3 = function () {

  this.preDispatch = function () {

  };

  this.postDispatch = function () {
    VStep3 = new Validator();

    VStep3
      .addValidator($('input[name="step3-input"]'), 1)
      .setStrictMode(true)
      .setIgnoreCase(false)
      .enableStepFinishAlert(true)
      .setPenalty(Rotator.getStepScore(4)*0.1);

    Rotator.enablePrevButton();
    Rotator.enableEndButton();

    enableTheoryLinks();

    $('button.check').click(function () {
      VStep3.setAttemptsOnCheckButton($(this));

      if(VStep3.validate() === true)
      {
        Rotator.disablePrevButton();
        Rotator.disableEndButton();
        disableTheoryLinks();
      }
      else
      {
        if(VStep3.getAttempts() != 0)

```

```

        {
            $('div.validation-alert-middle').fadeIn();
            setTimeout(() => {'div.validation-alert-middle').fadeOut()},1200);
        }
    });
};

this.mustache = function () {
    return {
        STEP3_INPUT: new TextInput('step3-input')
            .render()
    }
};

```

step4.html

```

<div class="jumbotron jumbotronT page4">
    <hr>
    <h4 style="text-align: center;line-height: 0.0;" >{{TASK4_TITLE}}</h4>
    <hr>

<div class="container task">
    <div class="row">
        <div class="col-lg-6 col-md-8 col-sm-10 col-lg-offset-3 col-md-offset-2 col-sm-offset-1 block0">
            <div class="row">
                <div class="col dec">
                    <h5>
                        <div class="task-dec rounded-4 text-center">
                            {{{TASK4}}}
                        </div>
                    <hr>
                    
                    <map name="image-map4">
                        <area id="1" href="javascript:void(0);" coords="5,7,81,32" shape="rect">
                        <area id="2" href="javascript:void(0);" coords="90,8,162,34" shape="rect">
                        <area id="3" href="javascript:void(0);" coords="173,8,246,35" shape="rect">
                        <area id="4" href="javascript:void(0);" coords="24,78,218,102" shape="rect">
                        <area id="5" href="javascript:void(0);" coords="24,110,218,134" shape="rect">
                        <area id="6" href="javascript:void(0);" coords="23,140,36,155" shape="rect">
                        <area id="7" href="javascript:void(0);" coords="23,165,36,179" shape="rect">
                        <area id="8" href="javascript:void(0);" coords="22,189,36,203" shape="rect">
                    </map>
                    {{{STEP4_INPUT}}}
                </h5>
            </div>
        </div>
    </div>

    <div class="row">
        <div class="col-lg-6 col-xs-6 pull-left">
            <button class="btn btn-outline-success check">{{CHECK}}</button>
        </div>
        <div class="col-xs-6 pull-right">
            <a href="#" class="book-style" onclick="Rotator.switchStep(1)" data-toggle="tooltip" data-placement="right"
title="{{TO_THEORY}}">
                <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
            </a>
        </div>
    </div>

```

```

        </div>
        <br>
    </div>
</div>
</div>
<hr>
</div>

<script>
    $(document).ready(function(){
        $('[data-toggle="tooltip"]').tooltip();
    });
</script>

<script>
    $(function() {
        $('img[usemap]').rwdImageMaps();
    });
</script>

<script>
    $('area').click(function() {
        $('input[name="step4-input"]').val(this.id);
    });
</script>

```

step4.js

```

var VStep4;
var step4 = function () {

    this.preDispatch = function () {

    };

    this.postDispatch = function () {
        VStep4 = new Validator();

        VStep4
            .addValidator($('input[name="step4-input"]'), 4)
            .setStrictMode(true)
            .setIgnoreCase(false)
            .enableStepFinishAlert(true)
            .setPenalty(Rotator.getStepScore(5)*0.1);

        Rotator.enablePrevButton();
        Rotator.enableEndButton();

        enableTheoryLinks();

        $('button.check').click(function () {
            VStep4.setAttemptsOnCheckButton($(this));

            if(VStep4.validate() === true)
            {
                Rotator.disablePrevButton();
                Rotator.disableEndButton();
                disableTheoryLinks();
            }
            else

```

```

    {
      if(VStep4.getAttempts()!==0)
      {
        $('div.validation-alert-middle').fadeIn();
        setTimeout(() => {'div.validation-alert-middle').fadeOut(),1200);
      }
    }
  });
};

this.mustache = function () {
  return {
    STEP4_INPUT: new TextInput('step4-input')
      .render()
  }
}
};

```

step5.html

```

<div class="jumbotron jumbotronT page5">
  <hr>
  <h4 style="text-align: center;line-height: 0.0;" >{{TASK5_TITLE}}</h4>
  <hr>

<div class="container task">
  <div class="row">
    <div class="col-lg-5 col-lg-offset-1 col-xs-6 block1">
      <div class="row">
        <div class="col dec">
          <h5>
            <div class="task-dec rounded-3">
              {{{TASK5}}}
            </div>
            <hr><br>
            {{{STEP5_INPUT}}}
          </h5>
        </div>
      </div>
      <br>
      <div class="row">
        <div class="col-xs-6 pull-left">
          <button class="btn btn-outline-success check">{{CHECK}}</button>
        </div>
        <div class="col-xs-6 pull-right">
          <a href="javascript:void(0)" class="book-style" onclick="Rotator.switchStep(1)" data-toggle="tooltip" data-
placement="right" title="{{TO_THEORY}}">
            <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
          </a>
        </div>
      </div>
    </div>

    <div class="col-lg-5 col-xs-6 block2 color-1">
      
    </div>
  </div>
</div>
<hr>
</div>

```



```
<script>
  $(document).ready(function(){
    $('[data-toggle="tooltip"]').tooltip();
    $('input[name="step5-input"]').attr('autocomplete','off');
  });
</script>
```

step5.js

```
var VStep5;
var step5 = function () {

  this.preDispatch = function () {

  };

  this.postDispatch = function () {
    VStep5 = new Validator();

    VStep5
      .addValidator($('input[name="step5-input"]'), 1)
      .setStrictMode(true)
      .setIgnoreCase(false)
      .enableStepFinishAlert(true)
      .setPenalty(Rotator.getStepScore(3)*0.1);

    Rotator.enablePrevButton();
    Rotator.enableEndButton();

    enableTheoryLinks();

    $('button.check').click(function () {
      VStep5.setAttemptsOnCheckButton($(this));

      if(VStep5.validate() === true)
      {
        Rotator.disablePrevButton();
        Rotator.disableEndButton();
        disableTheoryLinks();
      }
      else
      {
        if(VStep5.getAttempts() != 0)
        {
          $('div.validation-alert-middle').fadeIn();
          setTimeout(() => {$('div.validation-alert-middle').fadeOut()},1200);
        }
      }
    });
  };

  this.mustache = function () {
    return {
      STEP5_INPUT: new TextInput('step5-input')
        .render()
    }
  };
};
```

step6.html

```

<div class="jumbotron jumbotronT page6">
  <hr>
  <h4 style="text-align: center;line-height: 0.0;" >{{TASK6_TITLE}}</h4>
  <hr>

<div class="container task">
  <div class="row row-flex">
    <div class="col-lg-6 col-md-6 col-xs-8 col-lg-offset-3 col-md-offset-3 col-xs-offset-2 block3 color-3 rounded">
      <div class="row">
        <div class="col-lg-12">
          <h5>
            <div class="task-dec rounded-4">
              {{{TASK6}}}
            </div>

            <hr>
            <br>
            {{{STEP6_RADIOS}}}
          </h5>
        </div>
      </div>
      <br>
      <div class="row">
        <div class="col-lg-6 col-xs-6 pull-left">
          <button class="btn btn-outline-success check">{{CHECK}}</button>
        </div>
        <div class="col-lg-6 col-xs-6 pull-right">
          <a href="javascript:void(0)" class="book-style" onclick="Rotator.switchStep(1)" data-toggle="tooltip" data-
placement="right" title="{{TO_THEORY}}">
            <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
          </a>
        </div>
      </div>
    </div>
  </div>
  <hr>
</div>

<script>
$(document).ready(function(){
  $('[data-toggle="tooltip"]').tooltip();
});
</script>

```

step6.js

```

var VStep6;
var step6 = function () {

  this.preDispatch = function () {

  };
};

```

```

this.postDispatch = function () {
  VStep6 = new Validator();

  VStep6
    .addValidator($('input[name="step6-radios"]'), 'two')
    .setStrictMode(true)
    .setIgnoreCase(false)
    .enableStepFinishAlert(true)
    .setPenalty(Rotator.getStepScore(7)*0.1);

  Rotator.enablePrevButton();
  Rotator.enableEndButton();

  enableTheoryLinks();

  $('button.check').click(function () {
    VStep6.setAttemptsOnCheckButton($(this));

    if(VStep6.validate() === true)
      {
        Rotator.disablePrevButton();
        Rotator.disableEndButton();
        disableTheoryLinks();
      }
    else
      {
        if(VStep6.getAttempts() != 0)
          {
            $('div.validation-alert-middle').fadeIn();
            setTimeout(() => {$('div.validation-alert-middle').fadeOut()}, 1200);
          }
      }
  });
};

this.mustache = function () {
  return {
    STEP6_RADIOS: new Radios('step6-radios')
      .addRadio('{{STEP6_RADIO_TEXT_1}}', 'one')
      .addRadio('{{STEP6_RADIO_TEXT_2}}', 'two')
      .addRadio('{{STEP6_RADIO_TEXT_3}}', 'three')
      .randomize()
      .render(),
  }
};

```

step7.html

```

<div class="jumbotron jumbotronT page1">
  <hr>
  <h4 style="text-align: center;line-height: 0.0;" >{{TASK7_TITLE}}</h4>
  <hr>

<div class="container task">
  <div class="row row-flex">
    <div class="col-lg-8 col-md-10 col-sm-12 col-lg-offset-2 col-md-offset-1 block0 color-3 rounded">
      <div class="row">
        <div class="col dec">

```



```

var step7 = function () {

  this.preDispatch = function () {

  };

  this.postDispatch = function () {
    VStep7 = new Validator();

    VStep7
      .addValidator($('input[name="step7-textinput1"]'), ['1'], false, true)
      .addValidator($('input[name="step7-textinput2"]'), ['3'], false, true)
      .addValidator($('input[name="step7-textinput3"]'), ['2'], false, true)

      .setStrictMode(true)
      .setIgnoreCase(false)
      .enableStepFinishAlert(true)
      .setPenalty(Rotator.getStepScore(8)*0.1);

    Rotator.enablePrevButton();
    Rotator.enableEndButton();

    enableTheoryLinks();

    $('button.check').click(function () {
      VStep7.setAttemptsOnCheckButton($(this)); //dynamically changing amount of attempts left on check button

      if(VStep7.validate() === true)
        {
          Rotator.disablePrevButton();
          Rotator.disableEndButton();
          disableTheoryLinks();
        }
      else
        {
          if(VStep7.getAttempts() != 0)
            {
              $('div.validation-alert-middle').fadeIn();
              setTimeout(() => {$('div.validation-alert-middle').fadeOut()}, 1200);
            }
        }
    });
  };

  this.mustache = function () {
    return {
      STEP7_TEXTINPUT1: new TextInput('step7-textinput1')
        .render(),
      STEP7_TEXTINPUT2: new TextInput('step7-textinput2')
        .render(),
      STEP7_TEXTINPUT3: new TextInput('step7-textinput3')
        .render()
    }
  }
};

```

step8.html

```

<div class="jumbotron jumbotronT page8">
  <hr>

```

```

<h4 style="text-align: center;line-height: 0.0;" >{{TASK8_TITLE}}</h4>
<hr>

<div class="container task">
  <div class="row row-flex">
    <div class="col-lg-8 col-xs-8 col-lg-offset-2 col-xs-offset-2 block3 color-3 rounded">
      <div class="row">
        <div class="col-lg-12">
          <h5>
            <div class="task-dec rounded-4">
              {{{TASK8}}}
            </div>

            <hr>
            <br>
            {{{STEP8_RADIOS}}}
          </h5>
        </div>
      </div>
      <br>
      <div class="row">
        <div class="col-lg-6 col-xs-6 pull-left">
          <button class="btn btn-outline-success check">{{CHECK}}</button>
        </div>
        <div class="col-lg-6 col-xs-6 pull-right">
          <a href="javascript:void(0)" class="book-style" onclick="Rotator.switchStep(1)" data-toggle="tooltip" data-
placement="right" title="{{TO_THEORY}}">
            <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
          </a>
        </div>
      </div>
    </div>
  </div>
</div>
<hr>
</div>

<script>
  $(document).ready(function(){
    $('[data-toggle="tooltip"]').tooltip();
  });
</script>

```

step8.js

```

var VStep8;
var step8 = function () {

  this.preDispatch = function () {

  };

  this.postDispatch = function () {
    VStep8 = new Validator();

    VStep8
      .addValidator($('input[name="step8-radios"]'), 'one')
      .setStrictMode(true)
      .setIgnoreCase(false)
      .enableStepFinishAlert(true)

```

```

        .setPenalty(Rotator.getStepScore(9)*0.1);

Rotator.enablePrevButton();
Rotator.enableEndButton();

enableTheoryLinks();

$('button.check').click(function () {
    VStep8.setAttemptsOnCheckButton($(this));

    if(VStep8.validate() === true)
    {
        Rotator.disablePrevButton();
        Rotator.disableEndButton();
        disableTheoryLinks();
    }
    else
    {
        if(VStep8.getAttempts() != 0)
        {
            $('div.validation-alert-middle').fadeIn();
            setTimeout(() => {$('div.validation-alert-middle').fadeOut()},1200);
        }
    }
});
};

this.mustache = function () {
    return {
        STEP8_RADIOS: new Radios('step8-radios')
            .addRadio('{{STEP8_RADIO_TEXT_1}}', 'one')
            .addRadio('{{STEP8_RADIO_TEXT_2}}', 'two')
            .addRadio('{{STEP8_RADIO_TEXT_3}}', 'three')
            .randomize()
            .render()
    }
};
};

```

step9.html

```

<div class="jumbotron jumbotronT page10">
  <hr>
  <h4 style="text-align: center;line-height: 0.0;" >{{TASK9_TITLE}}</h4>
  <hr>

<div class="container task">
  <div class="row row-flex">
    <div class="col-lg-5 col-lg-offset-1 col-xs-6 block1 rounded">
      <div class="row">
        <div class="col dec">
          <h5>
            <div class="task-dec rounded-3">
              {{{TASK9}}}
            </div>
          <hr>
          <div class="greyBlock">
            {{{STEP9_SELECT}}}
          <br>
        </div>
      </div>
    </div>
  </div>

```

```

        </h5>
    </div>
</div>
<br>
<div class="row">
    <div class="col-lg-6 col-xs-6 pull-left">
        <button class="btn btn-outline-success check">{{ CHECK }}</button>
    </div>
    <div class="col-lg-6 col-xs-6 pull-right">
        <a href="javascript:void(0)" class="book-style" onclick="Rotator.switchStep(1)" data-toggle="tooltip" data-
placement="right" title="{{ TO_THEORY }}">
            <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
        </a>
    </div>
</div>
</div>
</div>

<div class="col-lg-5 col-xs-6 block2 color-1">
    
</div>
</div>
</div>
<hr>
</div>

<script>
    $(document).ready(function(){
        $('[data-toggle="tooltip"]').tooltip();
    });
</script>

```

step9.js

```

var VStep9;
var step9 = function () {

    this.preDispatch = function () {

    };

    this.postDispatch = function () {
        VStep9 = new Validator();

        VStep9
            .addValidator($('select[name="step9-select"]'), 1)
            .setStrictMode(true)
            .setIgnoreCase(false)
            .enableStepFinishAlert(true)
            .setPenalty(Rotator.getStepScore(10)*0.1);

        Rotator.enablePrevButton();
        Rotator.enableEndButton();

        enableTheoryLinks();

        $('button.check').click(function () {
            VStep9.setAttemptsOnCheckButton($(this));

            if(VStep9.validate() === true)
                {

```



```

    Rotator.disablePrevButton();
    Rotator.disableEndButton();
    disableTheoryLinks();
  }
  else
  {
    if(VStep9.getAttempts()!==0)
    {
      $('div.validation-alert-middle').fadeIn();
      setTimeout(() => {'div.validation-alert-middle').fadeOut(),1200);
    }
  }
});
};

this.mustache = function () {
  return {
    STEP9_SELECT: new Select('step9-select')
      .addOption('{{STEP9_OPTION_ONE}}', 0)
      .addOption('{{STEP9_OPTION_TWO}}', 1)
      .randomize()
      .render()
  }
}
};

```

step10.html

```

<div class="jumbotron jumbotronT page10">
  <hr>
  <h4 style="text-align: center;line-height: 0.0;" >{{TASK10_TITLE}}</h4>
  <hr>

  <div class="container task">
    <div class="row">
      <div class="col-lg-8 col-md-8 col-sm-10 col-lg-offset-2 col-md-offset-2 col-sm-offset-1 block0">
        <div class="row">
          <div class="col dec">
            <h5>
              <div class="task-dec rounded-4 text-center">
                {{{TASK10}}}
              </div>
            <hr>
            
            <map name="image-map10">
              <area id="1" href="javascript:void(0);" coords="-1,0,378,212" shape="rect">
              <area id="2" href="javascript:void(0);" coords="0,213,379,414" shape="rect">
            </map>
            {{{STEP10_INPUT}}}
          </h5>
        </div>
      </div>
    </div>

    <div class="row">
      <div class="col-lg-6 col-xs-6 pull-left">
        <button class="btn btn-outline-success check">{{CHECK}}</button>
      </div>
      <div class="col-xs-6 pull-right">

```

```

        <a href="javascript:void(0)" class="book-style" onclick="Rotator.switchStep(1)" data-toggle="tooltip" data-
placement="right" title="{{TO_THEORY}}">
        <span class="glyphicon glyphicon-book" aria-hidden="true"></span>
    </a>
</div>
</div>
<br>
</div>
</div>
</div>
<hr>
</div>

<script>
$(document).ready(function(){
    $('[data-toggle="tooltip"]').tooltip();
});
</script>

<script>
$(function() {
    $('img[usemap]').rwdImageMaps();
});
</script>

<script>
$('area').click(function() {
    $('input[name="step10-input"]').val(this.id);
});
</script>

```

step10.js

```

var VStep10;// A variable for future validator
var step10 = function () {

```

```

    this.preDispatch = function () {

    };

```

```

    this.postDispatch = function () {
        VStep10 = new Validator();

```

```

        VStep10
            .addValidator($('input[name="step10-input"]'), 1)
            .setStrictMode(true)
            .setIgnoreCase(false)
            .enableStepFinishAlert(true)
            .setPenalty(Rotator.getStepScore(11)*0.1);

```

```

        Rotator.enablePrevButton();
        Rotator.enableEndButton();
        enableTheoryLinks();

```

```

        setTimeout(showReturnFromTheoryButton, 1000);
        enableTheoryLinks();

```

```

        $('button.check').click(function () {
            VStep10.setAttemptsOnCheckButton($(this)); //dynamically changing amount of attempts left on check button

```

```

if(VStep10.validate() === true) // validate the validators
{
  Rotator.disablePrevButton();
  Rotator.disableEndButton();
  disableTheoryLinks();
}
else
{
  if(VStep10.getAttempts() != 0)
  {
    $('div.validation-alert-middle').fadeIn();
    setTimeout(() => {'div.validation-alert-middle').fadeOut(),1200);
  }
}
});
};

this.mustache = function () {
  return {
    STEP10_INPUT: new TextInput('step10-input')
      .render()
  }
}
};

```

results.html

```

<div class="row jumbotron jumbotronT">
  <hr>

<div class="container">
  <div class="row">
    <div class="col-lg-6 col-xs-6 col-sm-8 col-lg-offset-3 col-xs-offset-3 col-xs-offset-2 results shadow">
      <div class="column1 col-lg-12 col-xs-12">
        <h2>{{RESULTS_TITLE}}</h2>
      </div>
      <br>
      <hr>
      <br>
      <div id="gc"></div>
      <br>
      <hr>
      <h5>
        <div class="column2">
          <strong>{{RESULTS_START_TIME}}:</strong> <b>{{START_TIME}}</b>
        </div>
      </h5>
      <hr>
      <h5>
        <div class="column2">
          <strong>{{RESULTS_END_TIME}}:</strong> <b>{{END_TIME}}</b>
        </div>
      </h5>
      <hr>
      <h5>
        <div class="column2">
          <strong>{{RESULTS_TIME_DIFF}}:</strong> <span class="badge">{{TIME_DIFF}}
        </div>
      </h5>
      <b>{{RESULTS_TIME_SEC}}</b></span>
    </div>

```

```

</h5>
<hr>
<div>
  <h2>
    <div class="column3">
      <strong>{{RESULTS_YOUR_SCORE}}: </strong><b>{{RESULTS_POINTS_IN_PERCENT}}%</b>
    </div>
  </h2>
</div>
</div>
<div class="clearfix"></div>
</div>
<hr>
</div>

```

results.js

```

var results = function () {

  this.preDispatch = function () {
    Scorer.end();
  };

  this.postDispatch = function () {
    //you can use or not this diagram
    var g = new GoogleCharts(); // Making a call to google charts api to build a column chart

    var scores = Rotator.getAllStepScores();
    var userScores = Scorer.getUserStepScores();

    var header;
    var step_name;
    var _title;
    var _subtitle;

    // Set the language to Google Chart marks
    var header_eng = ['Step', 'Step maximum', 'Your score'];
    var step_eng = ['Introduction', 'Theory', 'Step 1', 'Step 2', 'Step 3', 'Step 4', 'Step 5', 'Step 6', 'Step 7', 'Step 8', 'Step 9', 'Step
10', 'Step 11', 'Step 12', 'Results'];
    var title_eng = 'Your results';
    var subtitle_eng = 'Step statistics for current trainer run';

    var header_ru = ['Шаг', 'Максимум шага', 'Ваш счет'];
    var step_ru = ['Вступление', 'Теория', 'Шаг 1', 'Шаг 2', 'Шаг 3', 'Шаг 4', 'Шаг 5', 'Шаг 6', 'Шаг 7', 'Шаг 8', 'Шаг 9', 'Шаг
10', 'Шаг 11', 'Шаг 12', 'Результаты'];
    var title_ru = 'Ваши результаты';
    var subtitle_ru = 'Пошаговая статистика';

    var header_ua = ['Крок', 'Максимум на кроці', 'Ваш рахунок'];
    var step_ua = ['Вступ', 'Теорія', 'Крок 1', 'Крок 2', 'Крок 3', 'Крок 4', 'Крок 5', 'Крок 6', 'Крок 7', 'Крок 8', 'Крок 9', 'Крок
10', 'Крок 11', 'Крок 12', 'Результати'];
    var title_ua = 'Ваші результати';
    var subtitle_ua = 'Крокова статистика';

    if (I18N.getCurrentLang() === 'en')
    {
      header = header_eng;
      step_name = step_eng;
    }
  };
}

```

```

        _title = title_eng;
        _subtitle = subtitle_eng;
    }
    else
    if (I18N.getCurrentLang() === 'ua')
    {
        header = header_ua;
        step_name = step_ua;
        _title = title_ua;
        _subtitle = subtitle_ua;
    }
    else
    if (I18N.getCurrentLang() === 'ru')
    {
        header = header_ru;
        step_name = step_ru;
        _title = title_ru;
        _subtitle = subtitle_ru;
    }

    var data = [
        [header[0], header[1], header[2]],
//      [Introduction', scores[0], userScores[0]],
//      [Theory', scores[1], userScores[1]],
        [step_name[2], scores[2], userScores[2]],
            [step_name[3], scores[3], userScores[3]],
            [step_name[4], scores[4], userScores[4]],
            [step_name[5], scores[5], userScores[5]],
            [step_name[6], scores[6], userScores[6]],
            [step_name[7], scores[7], userScores[7]],
            [step_name[8], scores[8], userScores[8]],
            [step_name[9], scores[9], userScores[9]],
            [step_name[10], scores[10], userScores[10]],
            [step_name[11], scores[11], userScores[11]],
//      [Results',scores[14], userScores[14]]
    ];

    var options = {
        chart: {
            title: _title,
            subtitle: _subtitle
        }
    };

    g.setType("bar");
    g.setLibrary("google.charts.Bar");
    g.setData(data);
    g.setOptions(options);
    g.doQuery($('#div#gc'));

    if (PRODUCTION)
        Service.pushResults();

    Rotator.enablePrevButton();
    Rotator.disableEndButton();

    // Disable prevButton for EarlyEnd client
    if($('#div[data-step="11"]').length==0)
    Rotator.disablePrevButton();

    // Hide endButton and show a beginButton
    $('#endTrainer').hide();
    $('#begin').show();

```

```

};

this.mustache = function () {
  return {
    START_TIME: Scorer.getFormattedStartTime(),
    END_TIME: Scorer.getFormattedEndTime(),
    TIME_DIFF: Scorer.getTimeDifference(),
    SCORE: Scorer.getScore(),
    RESULTS_POINTS: getNumEnding(Scorer.getScore(), [ '{{POINTS1}}', '{{POINTS2}}', '{{POINTS3}}' ]),
    RESULTS_POINTS_IN_PERCENT: Scorer.getScoreInPercent()
  }
}
};

```

trainer.config.json

```

{
  "LANGUAGES": {
    "ua": "Українська",
    "ru": "Русский",
    "en": "English"
  },
  "DEFAULT_LANG": "ua",
  "TRAINER_SCORE": 100,
  "TRAINER_AUTHOR": "Semen Duvanov",
  "COURSE_AUTHOR": "Irina Baranova",
  "PRODUCTION": false
}

```

trainer.steps.json

```

[
  {
    "filename" : "introduction",
    "score" : 0,
    "hasScript" : true
  },
  {
    "filename" : "theory",
    "score" : 0,
    "hasScript" : true
  },
  {
    "filename" : "step1",
    "score" : 10,
    "hasScript" : true
  },
  {
    "filename" : "step2",
    "score" : 10,
    "hasScript" : true
  },
  {
    "filename" : "step3",
    "score" : 10,

```

```

    "hasScript" : true
  },
  {
    "filename" : "step4",
    "score" : 10,
    "hasScript" : true
  },
  {
    "filename" : "step5",
    "score" : 10,
    "hasScript" : true
  },
  {
    "filename" : "step6",
    "score" : 10,
    "hasScript" : true
  },
  {
    "filename" : "step7",
    "score" : 10,
    "hasScript" : true
  },
  {
    "filename" : "step8",
    "score" : 10,
    "hasScript" : true
  },
  {
    "filename" : "step9",
    "score" : 10,
    "hasScript" : true
  },
  {
    "filename" : "step10",
    "score" : 10,
    "hasScript" : true
  },
  {
    "filename" : "results",
    "score" : 0,
    "hasScript" : true
  }
]

```

ua.json

```

{
  "local": "Українська",
  "lang": {
    "TRAINER_NAME": "Кінематика пов'язаних об'єктів",
    "MAX_SCORE": "Кількість балів",
    "NEXT_STEP": "Наступний крок",
    "PREV_STEP": "Попередній крок",
    "ENTER_TEXT": "Введіть текст",
    "CHOOSE_LANG": "Оберіть мову",
    "CHOOSE_SELECT": "Оберіть елемент зі списку",
    "TEACHER_HELP" : "Допомога викладача",
    "ENTER_HELP_TEXT": "Опишіть з чим Вам потрібна допомога...",
    "HELP_REQUEST_SEND": "Відправити",
    "HELP_REQUEST_CLOSE": "Відмінити",

```

"ALERT_WELL_DONE": "Відмінно!",
 "ALERT_FAIL": "Помилка! Спробуйте ще",
 "ALERT_STEP_SUCCESS": " Крок пройдений! Перейдіть до наступного.",
 "ALERT_STEP_FAIL": "Ви погано засвоїли теоретичний матеріал. Повторіть його і почніть виконувати завдання з початку.",

"STEP_COUNT": "Кількість кроків",
 "END_TRAINER": "Завершити сеанс",
 "BEGIN_TRAINER": "Розпочати знову",
 "END_ARE_YOU_SURE": "Ви впевнені, що хочете завершити виконання тренажера?",
 "END_TRAINER_CLOSE": "Продовжити поточний сеанс",
 "END_TRAINER_SEND": "Завершити поточний сеанс",
 "CHECK": "Перевірити",
 "ATTEMPTS_LEFT": "Ще спроб - ",

"THEORY_TITLE": "Теорія",
 "TO_THEORY": "До теорії",
 "FROM_THEORY": "Назад",

"THEORY_COLUMN1": "Ієрархія",
 "THEORY_COLUMN2": "Пряма кінематика",
 "THEORY_COLUMN3": "Інверсна кінематика",

"TH1_HEAD": "Ієрархія",
 "TH1_TEXT1": "Ієрархія – це набір об'єктів зі зв'язками між ними за принципом «предок-нащадок» (інша назва: «батьківський об'єкт — дочірній об'єкт»)). Предок контролює поведінку одного або більше нащадків. У той же час предки можуть контролюватися іншими предками, що мають більш високий рівень ієрархії. Нашадок, контрольований деяким предком, може виявитися предком для інших нащадків, що знаходяться за ієрархією нижче нього. Найвищий об'єкт ієрархії управляє всіма об'єктами ієрархії і називається кореневим або Root-об'єктом.",
 "TH1_TEXT2": "Ієрархія (зв'язок) об'єктів задається з використанням команди Select and Link, представленої на зображенні нижче",
 "TH1_TEXT3": "Об'єкти зв'язуються від нащадка до батька (батьком для всіх буде найперший об'єкт)",
 "TH1_TEXT4": "В результаті буде отримана наступна ієрархія об'єктів",
 "TH1_TEXT5": "Розірвати ієрархію об'єктів можна наступним чином",
 "TH1_TEXT6": "Переглянути зв'язки (при їх наявності) також можна, активувавши вкладку командної панелі Display (Відображення), що відповідає за управління відображенням об'єктів сцени у вікнах проєкцій, і в сувої Link Display для кожного з об'єктів включивши прапорець Display Links (Відобразити зв'язок)",

"TH2_HEAD": "Пряма кінематика",
 "TH2_TEXT1": "Положення і орієнтація пов'язаних в ієрархічну структуру об'єктів залежить також від обраного методу управління ланцюжком. Таких методів два: пряма кінематика та інверсна (зворотна) кінематика.",
 "TH2_TEXT2": "У випадку прямої кінематики ієрархічні зв'язки передаються від предків до нащадків, тобто дочірні об'єкти успадковують поведінку батьківських і з ними відбуваються ті ж самі перетворення (переміщення, масштабування і т.д.). У той же час трансформація нащадка не викликає трансформації батьківського об'єкта.",
 "TH2_TEXT3": "Приклад роботи прямої кінематики наведено нижче",

"TH3_HEAD": "Інверсна кінематика",
 "TH3_TEXT1": "Інверсна кінематика працює по-іншому — нащадки приводять в рух батьківські об'єкти. Іншими словами, задається положення і орієнтація нащадка, і за цими даними програма сама розраховує положення і орієнтацію батьківських по відношенню до нащадка об'єктів.",
 "TH3_TEXT2": "Вибір методу проводиться на вкладці командної панелі Hierarchy (Ієрархія), призначеної для управління зв'язками. За замовчуванням використовується метод прямої кінематики — для активування режиму зворотної кінематики слід відкрити на панелі Hierarchy закладку IK і клацнути по кнопці Interactive IK.",
 "TH3_TEXT3": "Приклад роботи інверсної кінематики наведено нижче",

"TASK1_TITLE": "Клацніть по правильному варіанту і виконайте перевірку",
 "TASK1": "Яка з кнопок служить для створення ієрархій ?",

"TASK2_TITLE": "Клацніть по правильному варіанту і виконайте перевірку",

"TASK2": "Усі об'єкти на малюнку було зв'язано послідовно в одному напрямку інструментом Select and Link. Після зв'язки один об'єкт було переміщено вгору. Визначте по малюнку який об'єкт буде батьківським для всіх інших",

"TASK3_TITLE": "Клацніть по правильному варіанту і виконайте перевірку",
 "TASK3": "Виберіть у вікні Scene Explorer об'єкт, відзначений на малюнку",

"TASK4_TITLE": "Клацніть по правильному варіанту і виконайте перевірку",
 "TASK4": "Яка кнопка активує інверсну кінематику ?",

"TASK5_TITLE": "Впишіть номер потрібного об'єкту в текстове поле",
 "TASK5": "Який об'єкт потрібно перемістити вгору в режимі інверсної кінематики, щоб отримати форму зв'язки як на малюнку",

"TASK5_TOOLTIP": "Інверсна кінематика",

"TASK6_TITLE": "Виберіть правильну відповідь",
 "TASK6": "Що таке Ієрархія ?",
 "STEP6_RADIO_TEXT_1": "Відношення \"Нащадок-Нащадок\"",
 "STEP6_RADIO_TEXT_2": "Відношення \"Предок-Нащадок\"",
 "STEP6_RADIO_TEXT_3": "Зв'язок \"Батько-Батько\"",

"TASK7_TITLE": "Впишіть числа в текстові поля (від 1 до 3)",
 "TASK7": "Задайте правильну послідовність дій для включення режиму інверсної кінематики",
 "MARK_STEP7_INPUT1": "Створити об'єкти",
 "MARK_STEP7_INPUT2": "Активувати інверсну кінематику",
 "MARK_STEP7_INPUT3": "Налаштувати зв'язки об'єктів",

"TASK8_TITLE": "Виберіть правильну відповідь",
 "TASK8": "Чим відрізняється Інверсна кінематика від Прямої кінематики ?",
 "STEP8_RADIO_TEXT_1": "Нащадок приводить в рух батьків",
 "STEP8_RADIO_TEXT_2": "Батько приводить в рух нащадків",
 "STEP8_RADIO_TEXT_3": "Відсутністю необхідності задавати ієрархію до включення режиму",

"TASK9_TITLE": "Виберіть коректний елемент зі списку",
 "TASK9": "Який тип кінематики представлений на малюнку",
 "STEP9_OPTION_ONE": "Пряма кінематика",
 "STEP9_OPTION_TWO": "Інверсна кінематика",

"TASK10_TITLE": "Клацніть по правильному варіанту і виконайте перевірку",
 "TASK10": "Оберіть рисунок, на якому показано варіант зв'язування об'єктів, коли батьківським стане крайній лівий",

"RESULTS_TITLE": "Ваші результати",
 "RESULTS_START_TIME": "Час початку тестування",
 "RESULTS_END_TIME": "Час завершення тестування",
 "RESULTS_TIME_DIFF": "Час виконання",
 "RESULTS_TIME_SEC": "хв.",
 "RESULTS_YOUR_SCORE": "Ви отримали",

"STEP1_NAME": "Вступ",
 "STEP2_NAME": "Теорія",
 "STEP3_NAME": "Крок 1",
 "STEP4_NAME": "Крок 2",
 "STEP5_NAME": "Крок 3",
 "STEP6_NAME": "Крок 4",
 "STEP7_NAME": "Крок 5",
 "STEP8_NAME": "Крок 6",
 "STEP9_NAME": "Крок 7",
 "STEP10_NAME": "Крок 8",
 "STEP11_NAME": "Крок 9",
 "STEP12_NAME": "Крок 10",
 "STEP13_NAME": "Результати"

}

}

en.json

```
{
  "local": "English",
  "lang": {
    "TRAINER_NAME": "Kinematics of the related objects",
    "MAX_SCORE": "Max score",
    "NEXT_STEP": "Next step",
    "PREV_STEP": "Previous step",
    "ENTER_TEXT": "Enter text",
    "CHOOSE_LANG": "Choose language",
    "CHOOSE_SELECT": "Choose element from the list",
    "TEACHER_HELP": "Teachers help",
    "ENTER_HELP_TEXT": "Describe what you are having problem with...",
    "HELP_REQUEST_SEND": "Send",
    "HELP_REQUEST_CLOSE": "Cancel",
    "CALCULATOR": "Calculator",
    "ALERT_WELL_DONE": "Well done!",
    "ALERT_FAIL": "<strong>Error!</strong> Please try again",
    "ALERT_STEP_SUCCESS": " The step is done perfectly! Go to the next one.",
    "ALERT_STEP_FAIL": "You have not learned the theoretical material well. Start over and repeat it.",
    "STEP_COUNT": "Number of steps",
    "END_TRAINER": "Finish session",
    "BEGIN_TRAINER": "Start again",
    "END_ARE_YOU_SURE": "Are you sure you want to end trainer execution?",
    "END_TRAINER_CLOSE": "Continue current session",
    "END_TRAINER_SEND": "Finish current session",
    "CHECK": "Check",
    "ATTEMPTS_LEFT": "Attempts left - ",

    "THEORY_TITLE": "Theory",
    "TO_THEORY": "To theory",
    "FROM_THEORY": "Back",

    "THEORY_COLUMN1": "Hierarchy",
    "THEORY_COLUMN2": "Direct kinematics",
    "THEORY_COLUMN3": "Inverse kinematics",

    "TH1_HEAD": "<span style='color: #259191;'>Hierarchy</span>",
    "TH1_TEXT1": "<span style='color: #259191;'>Hierarchy</span> – is a set of objects with relationships between them according to the principle «<b>ancestor</b>-<b>descendant</b>» (another name: «<b>parent object</b> – <b>child object</b>»). The ancestor controls the behavior of one or more descendants. At the same time, the ancestors can be controlled by other ancestors having a higher level of hierarchy. A descendant controlled by some ancestor may turn out to be an ancestor for other descendants who are in the hierarchy below him. The highest object in the hierarchy controls all objects in the hierarchy and is called the <b>root</b> or <b>Root-object</b>.",
    "TH1_TEXT2": "<span style='color: #259191;'>Hierarchy (relationship)</span> of objects is set using the command <b>Select and Link</b>, shown in the picture below",
    "TH1_TEXT3": "Objects are linked from child to parent (<span style='color: #2f5c5c;'>the parent for others will be the very first object</span>)",
    "TH1_TEXT4": "The result will be the following hierarchy of objects",
    "TH1_TEXT5": "You can break the hierarchy of objects as follows",
    "TH1_TEXT6": "You can also view the links by activating the command panel tab <b>Display</b>, which is responsible for controlling the display of scene objects in projection windows, and in the <b>Link Display</b> scroll for each of the objects, enabling the <b>Display Links</b> checkbox (Display links)",

    "TH2_HEAD": "<span style='color: #259191;'>Direct kinematics</span>",
```

"TH2_TEXT1": "The position and orientation of objects connected in a hierarchical structure also depends on the chosen method of chain management. There are two such methods: direct kinematics and inverse kinematics.",

"TH2_TEXT2": "In the case of direct kinematics hierarchical relationships can be communicated from ancestors to descendants, that is, the child objects inherit the parental behavior and undergo the same transformation (move, scale, etc.). At the same time, the transformation of the child is not causing transformation of the parent object.",

"TH2_TEXT3": "An example of the work of the direct kinematics is given below",

"TH3_HEAD": "Inverse kinematics",

"TH3_TEXT1": "Inverse kinematics works differently — descendants set parent objects in motion. In other words, the position and orientation of the child is set, and based on this data, the program itself calculates the position and orientation of the parent objects in relation to the child.",

"TH3_TEXT2": "The method is selected on the command panel tab Hierarchy, designed to manage relationships. The default method of direct kinematics — to activate the inverse kinematics should open on Hierarchy panel the bookmark IK and click the button Interactive IK.",

"TH3_TEXT3": "An example of the work of the inverse kinematics is given below",

"TASK1_TITLE": "Click on the correct option and make the check",

"TASK1": "Which of the buttons is used to create hierarchy ?",

"TASK2_TITLE": "Click on the correct option and make the check",

"TASK2": "All the objects in the drawing were connected sequentially in one direction with the Select and Link tool. After the bundle, one object was moved up. Determine from the drawing which object will be the parent for all the others",

"TASK3_TITLE": "Click on the correct option and make the check",

"TASK3": "Select the object in the Scene Explorer window marked on the image",

"TASK4_TITLE": "Click on the correct option and make the check",

"TASK4": "Which button activates the inverse kinematics ?",

"TASK5_TITLE": "Enter the number of an appropriate object into the text field",

"TASK5": "Which object should be moved up in the inverse kinematics mode to get the shape of the bundle as on the picture",

"TASK5_TOOLTIP": "Inverse kinematics",

"TASK6_TITLE": "Choose the right answer",

"TASK6": "What is Hierarchy ?",

"STEP6_RADIO_TEXT_1": "\"Descendant-Descendant\" Relationship",

"STEP6_RADIO_TEXT_2": "\"Ancestor-Descendant\" Relationship",

"STEP6_RADIO_TEXT_3": "\"Parent-Parent\" Relationship",

"TASK7_TITLE": "Enter the numbers in the text fields (from 1 to 3)",

"TASK7": "Set the correct sequence of actions to enable the inverse kinematics mode",

"MARK_STEP7_INPUT1": "Create objects",

"MARK_STEP7_INPUT2": "Activate inverse kinematics",

"MARK_STEP7_INPUT3": "Configure object connections",

"TASK8_TITLE": "Choose the right answer",

"TASK8": "What is difference between Inverse kinematics and Direct kinematics ?",

"STEP8_RADIO_TEXT_1": "The descendant sets the parents in motion",

"STEP8_RADIO_TEXT_2": "The parent sets the descendants in motion",

"STEP8_RADIO_TEXT_3": "There is no need to set the hierarchy before enabling the mode",

"TASK9_TITLE": "Select the correct item from the list",

"TASK9": "What type of kinematics is shown on the picture",

"STEP9_OPTION_ONE": "Direct kinematics",

"STEP9_OPTION_TWO": "Inverse kinematics",

"TASK10_TITLE": "Click on the correct option and make the check",

"TASK10": "Select the image that shows the variant to link objects when the leftmost one becomes the parent",

```

"RESULTS_TITLE": "Your scores",
"RESULTS_START_TIME": "You started at",
"RESULTS_END_TIME": "Finished at",
"RESULTS_TIME_DIFF": "Time to complete",
"RESULTS_TIME_SEC": "min.",
"RESULTS_YOUR_SCORE": "Your score is",

"STEP1_NAME": "Introduction",
"STEP2_NAME": "Theory",
"STEP3_NAME": "Step 1",
"STEP4_NAME": "Step 2",
"STEP5_NAME": "Step 3",
"STEP6_NAME": "Step 4",
"STEP7_NAME": "Step 5",
"STEP8_NAME": "Step 6",
"STEP9_NAME": "Step 7",
"STEP10_NAME": "Step 8",
"STEP11_NAME": "Step 9",
"STEP12_NAME": "Step 10",
"STEP13_NAME": "Results"
}
}

```

ru.json

```

{
  "local": "Русский",
  "lang": {
    "TRAINER_NAME": "Кинематика связанных объектов",
    "MAX_SCORE": "Количество баллов",
    "NEXT_STEP": "Следующий шаг",
    "PREV_STEP": "Предыдущий шаг",
    "ENTER_TEXT": "Введите текст",
    "CHOOSE_LANG": "Выберите язык",
    "CHOOSE_SELECT": "Выберите элемент из списка",
    "TEACHER_HELP" : "Помощь преподавателя",
    "ENTER_HELP_TEXT": "Опишите с чем Вам нужна помощь...",
    "HELP_REQUEST_SEND": "Отправить",
    "HELP_REQUEST_CLOSE": "Отмена",
    "CALCULATOR" : "Калькулятор",
    "ALERT_WELL_DONE": "Отлично! ",
    "ALERT_FAIL": "<strong>Ошибка!</strong> Попробуйте еще",
    "ALERT_STEP_SUCCESS": "Шаг пройден! Перейдите к следующему.",
    "ALERT_STEP_FAIL": "Вы плохо усвоили теоретический материал. Повторите его и начните выполнять задания сначала.",
    "STEP_COUNT": "Количество шагов",
    "END_TRAINER": "Завершить сеанс",
    "BEGIN_TRAINER": "Начать сначала",
    "END_ARE_YOU_SURE": "Вы уверены, что хотите завершить выполнение тренажера?",
    "END_TRAINER_CLOSE": "Продолжить текущий сеанс",
    "END_TRAINER_SEND": "Завершить текущий сеанс",
    "CHECK": "Проверить",
    "ATTEMPTS_LEFT": "Еще попыток - ",

    "THEORY_TITLE": "Теория",
    "TO_THEORY": "К теории",
    "FROM_THEORY": "Назад",

    "THEORY_COLUMN1": "Иерархия",
    "THEORY_COLUMN2": "Прямая кинематика",

```

"THEORY_COLUMN3": "Инверсная кинематика",

"TH1_HEAD": "Иерархия",

"TH1_TEXT1": "Иерархия – это набор объектов со связями между ними по принципу «предок-потомок» (другое название: «родительский объект – дочерний объект»). Предок контролирует поведение одного или более потомков. В то же время предки могут контролироваться другими предками, имеющими более высокий уровень иерархии. Потомок, контролируемый некоторым предком, может оказаться предком для других потомков, находящихся по иерархии ниже него. Самый верхний объект иерархии управляет всеми объектами иерархии и называется корневым или Root-объектом.",

"TH1_TEXT2": "Иерархия (связь) объектов задается с использованием команды Select and Link, представленной на картинке ниже",

"TH1_TEXT3": "Объекты связываются от потомка к родителю (родителем для всех будет самый первый объект)",

"TH1_TEXT4": "В результате будет получена следующая иерархия объектов",

"TH1_TEXT5": "Разорвать иерархию объектов можно следующим образом",

"TH1_TEXT6": "Просмотреть связи (при их наличии) также можно, активировав вкладку командной панели Display (Отображение), отвечающую за управление отображением объектов сцены в окнах проекций, и в свитке Link Display для каждого из объектов включив флажок Display Links (Отображать связи)",

"TH2_HEAD": "Прямая кинематика",

"TH2_TEXT1": "Положение и ориентация связанных в иерархическую структуру объектов зависит также от выбранного метода управления цепочкой. Таких методов два: прямая кинематика и инверсная (обратная) кинематика.",

"TH2_TEXT2": "В случае прямой кинематики иерархические связи передаются от предков к потомкам, то есть дочерние объекты наследуют поведение родительских и с ними происходят те же самые преобразования (перемещение, масштабирование и т.д.). В то же время трансформация потомка не вызывает трансформации родительского объекта.",

"TH2_TEXT3": "Пример работы прямой кинематики приведен ниже",

"TH3_HEAD": "Инверсная кинематика",

"TH3_TEXT1": "Инверсная кинематика работает по-иному — потомки приводят в движение родительские объекты. Иными словами, задается положение и ориентация потомка, и по этим данным программа сама рассчитывает положение и ориентацию родительских по отношению к потомку объектов.",

"TH3_TEXT2": "Выбор метода производится на вкладке командной панели Hierarchy (Иерархия), предназначенной для управления связями. По умолчанию используется метод прямой кинематики — для активирования режима обратной кинематики следует открыть на панели Hierarchy закладку IK и щелкнуть на кнопке Interactive IK.",

"TH3_TEXT3": "Пример работы инверсной кинематики приведен ниже",

"TASK1_TITLE": "Щелкните по правильному варианту и выполните проверку",

"TASK1": "Какая из кнопок служит для создания иерархий?",

"TASK2_TITLE": "Щелкните по правильному варианту и выполните проверку",

"TASK2": "Все объекты на рисунке были связаны последовательно в одном направлении инструментом Select and Link. После связи один объект был перемещен вверх. Определите по рисунку какой объект будет родительским для всех остальных",

"TASK3_TITLE": "Щелкните по правильному варианту и выполните проверку",

"TASK3": "Выберите в окне Scene Explorer объект, отмеченный на рисунке",

"TASK4_TITLE": "Щелкните по правильному варианту и выполните проверку",

"TASK4": "Какая кнопка активирует инверсную кинематику?",

"TASK5_TITLE": "Впишите номер нужного объекта в текстовое поле",

"TASK5": "Какой объект нужно переместить вверх в режиме инверсной кинематики, чтобы получить форму связки как на рисунке",

"TASK5_TOOLTIP": "Инверсная кинематика",

"TASK6_TITLE": "Выберите правильный ответ",

"TASK6": "Что такое Иерархия?",

"STEP6_RADIO_TEXT_1": "Отношение \"Потомок-Потомок\"",

"STEP6_RADIO_TEXT_2": "Отношение \"Предок-Потомок\"",

```

"STEP6_RADIO_TEXT_3": "Связь \"Родитель-Родитель\"",

"TASK7_TITLE": "Впишите числа в текстовые поля (от 1 до 3)",
"TASK7": "Задайте правильную последовательность действий для включения режима инверсной кинематики",
"MARK_STEP7_INPUT1": "Создать объекты",
"MARK_STEP7_INPUT2": "Активировать инверсную кинематику",
"MARK_STEP7_INPUT3": "Настроить связи объектов",

"TASK8_TITLE": "Выберите правильный ответ",
"TASK8": "Чем отличается <em>Инверсная кинематика</em> от <em>Прямой кинематики</em> ?",
"STEP8_RADIO_TEXT_1": "Потомок приводит в движение родителей",
"STEP8_RADIO_TEXT_2": "Родитель приводит в движение потомков",
"STEP8_RADIO_TEXT_3": "Отсутствием необходимости задавать иерархию до включения режима",

"TASK9_TITLE": "Выберите корректный элемент из списка",
"TASK9": "Какой тип кинематики представлен на рисунке",
"STEP9_OPTION_ONE": "Прямая кинематика",
"STEP9_OPTION_TWO": "Инверсная кинематика",

"TASK10_TITLE": "Щелкните по правильному варианту и выполните проверку",
"TASK10": "Выберите рисунок, на котором показан вариант связывания объектов, когда родительским станет крайний левый",

"RESULTS_TITLE": "Ваши результаты",
"RESULTS_START_TIME": "Начало тестирования",
"RESULTS_END_TIME": "Окончание тестирования",
"RESULTS_TIME_DIFF": "Время выполнения",
"RESULTS_TIME_SEC": "мин.",
"RESULTS_YOUR_SCORE": "Вы набрали",
"STEP": "Шаг",

"STEP1_NAME": "Вступление",
"STEP2_NAME": "Теория",
"STEP3_NAME": "Шаг 1",
"STEP4_NAME": "Шаг 2",
"STEP5_NAME": "Шаг 3",
"STEP6_NAME": "Шаг 4",
"STEP7_NAME": "Шаг 5",
"STEP8_NAME": "Шаг 6",
"STEP9_NAME": "Шаг 7",
"STEP10_NAME": "Шаг 8",
"STEP11_NAME": "Шаг 9",
"STEP12_NAME": "Шаг 10",
"STEP13_NAME": "Результаты"
}
}

```

customScript.js

```

var enableTheoryLinks = function () {
    $('a.book-style').removeClass('isDisabled');
    $('a.book-style').attr('onclick', 'Rotator.switchStep(1)');
};

var disableTheoryLinks = function () {
    $('a.book-style').addClass('isDisabled');
    $('a.book-style').removeAttr('onclick');
};

var hideReturnFromTheoryButton= function () {

```

```
    $('a.return-style').hide();  
};  
  
var showReturnFromTheoryButton= function () {  
    $('a.return-style').show();  
};
```

Додаток В

РЕЗУЛЬТАТ РОЗРОБКИ КОМПЛЕКСУ ВЕБ-ТРЕНАЖЕРІВ

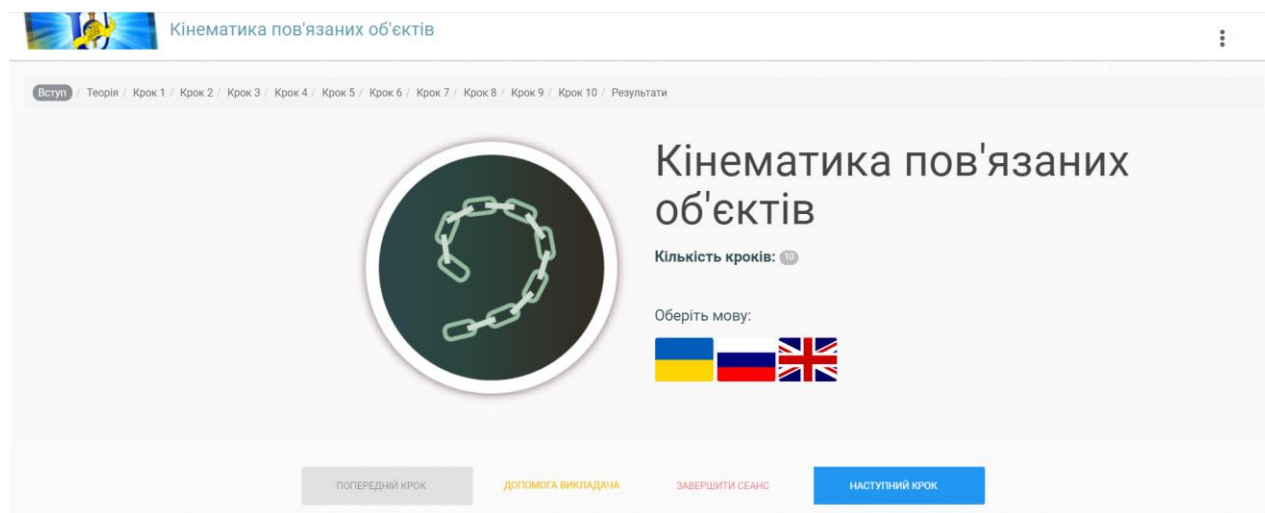


Рисунок В.1 – Вступ тренажеру «Кінематика пов'язаних об'єктів»

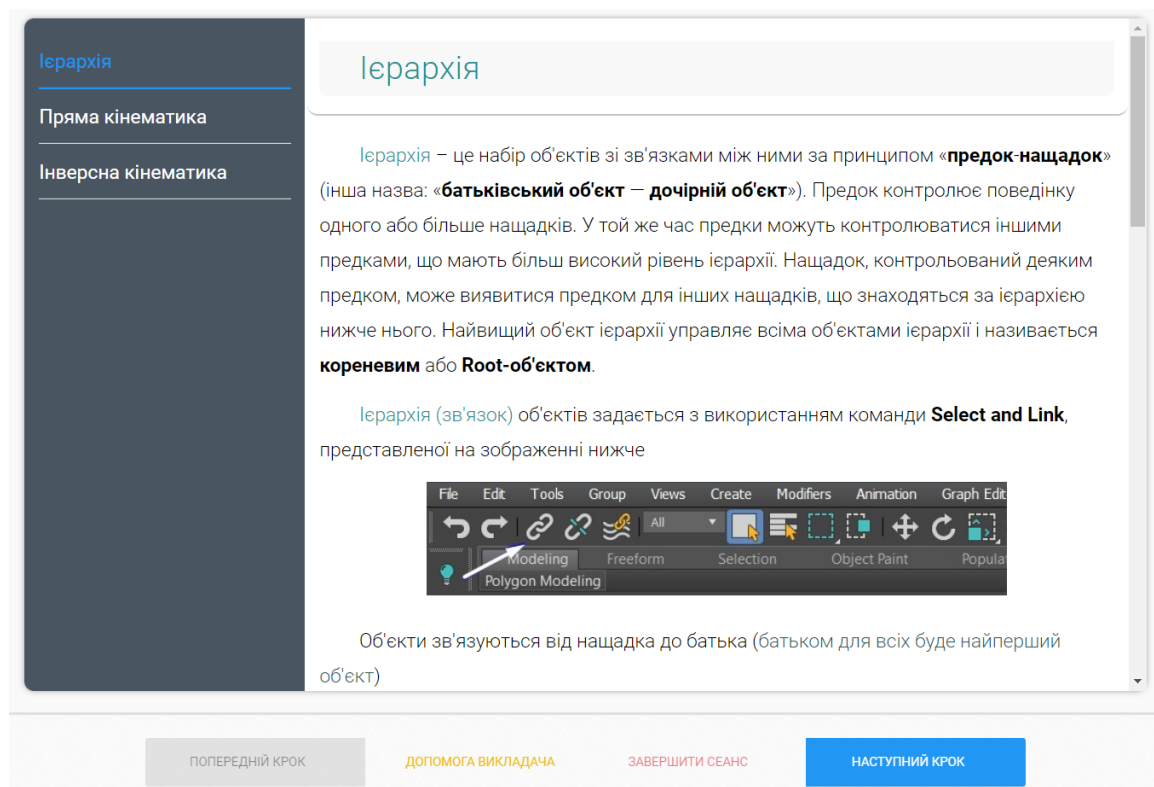


Рисунок В.2 – Теоретичний розділ тренажеру «Кінематика пов'язаних об'єктів»

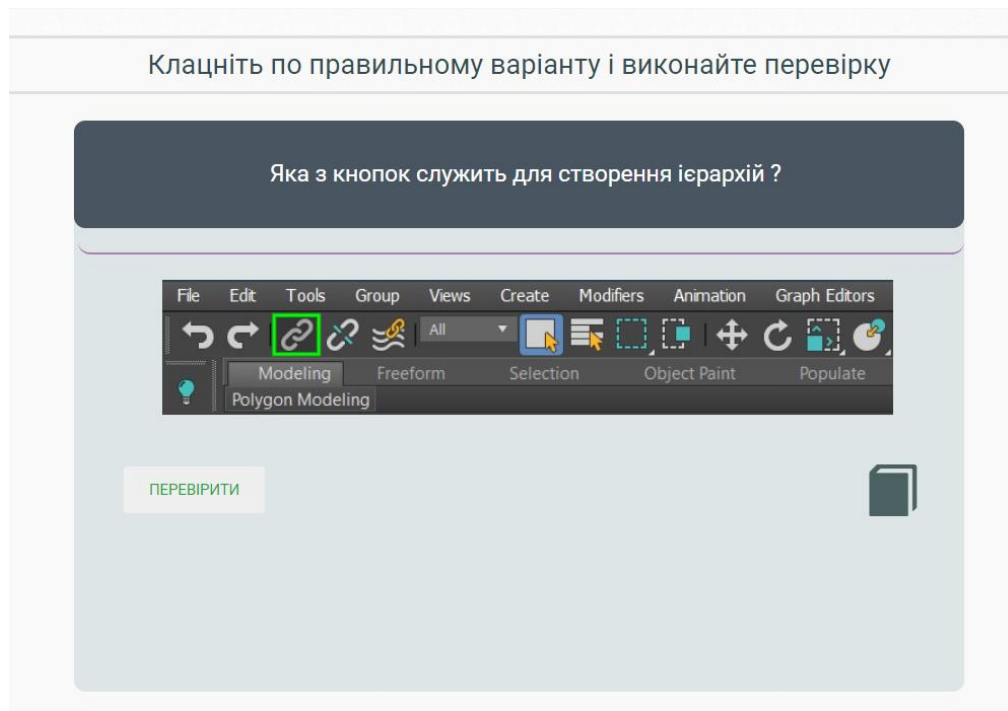


Рисунок В.3 – Крок 1 розділу завдань тренажеру «Кінематика пов'язаних об'єктів»

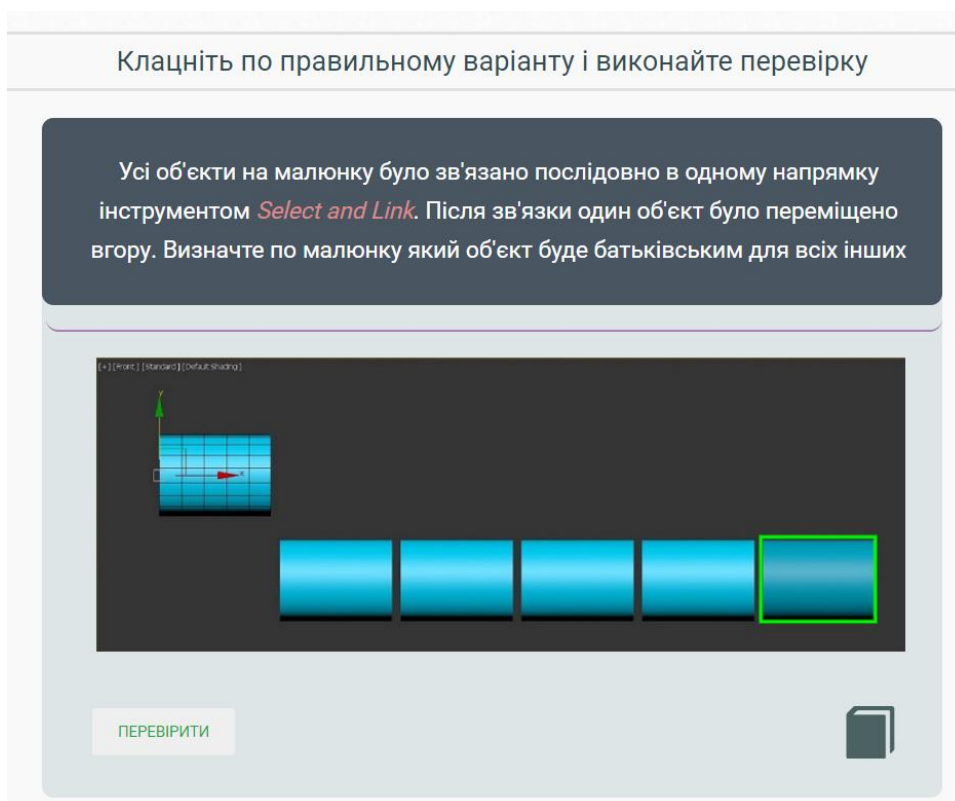


Рисунок В.4 – Крок 2 розділу завдань тренажеру «Кінематика пов'язаних об'єктів»

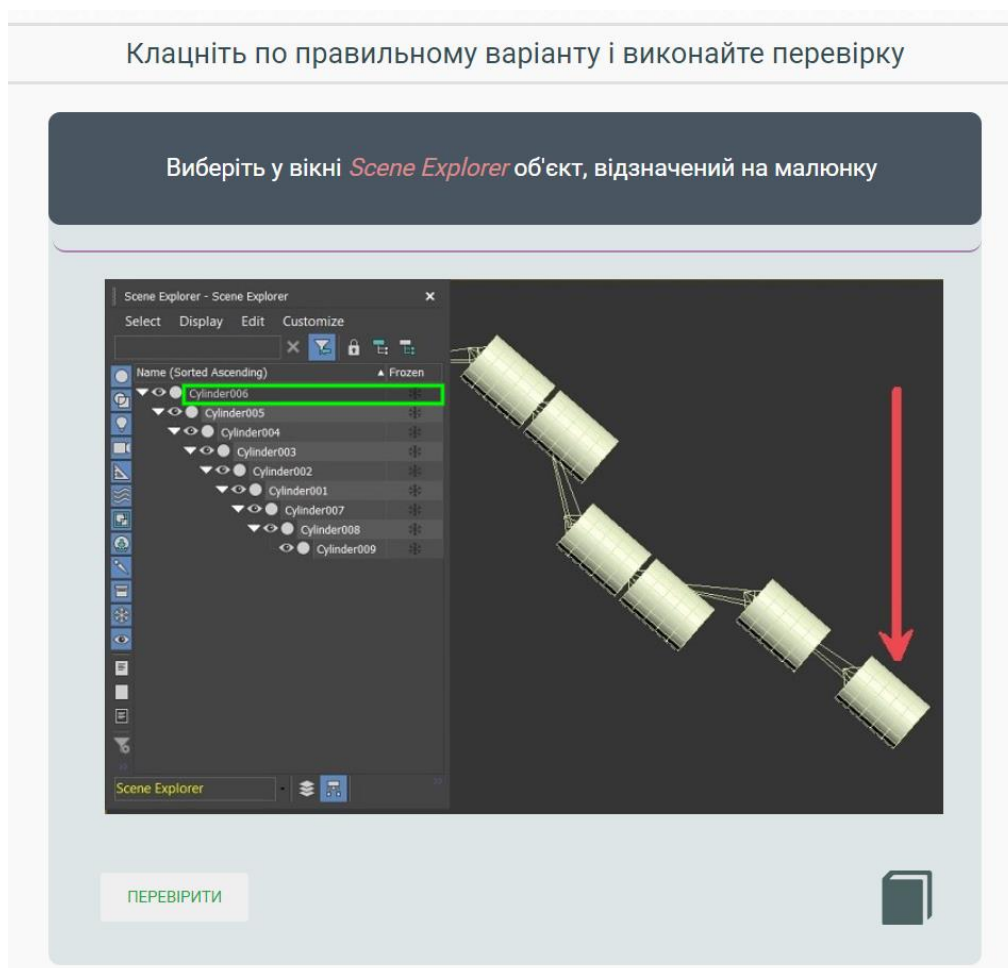


Рисунок В.5 – Крок 3 розділу завдань тренажеру «Кінематика пов'язаних об'єктів»

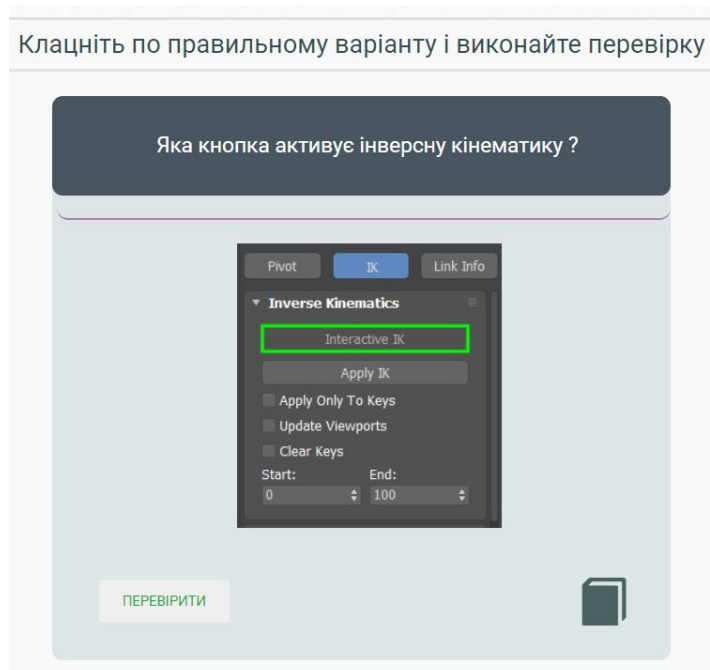


Рисунок В.6 – Крок 4 розділу завдань тренажеру «Кінематика пов'язаних об'єктів»



Рисунок В.7 – Крок 5 розділу завдань тренажеру «Кінематика пов'язаних об'єктів»

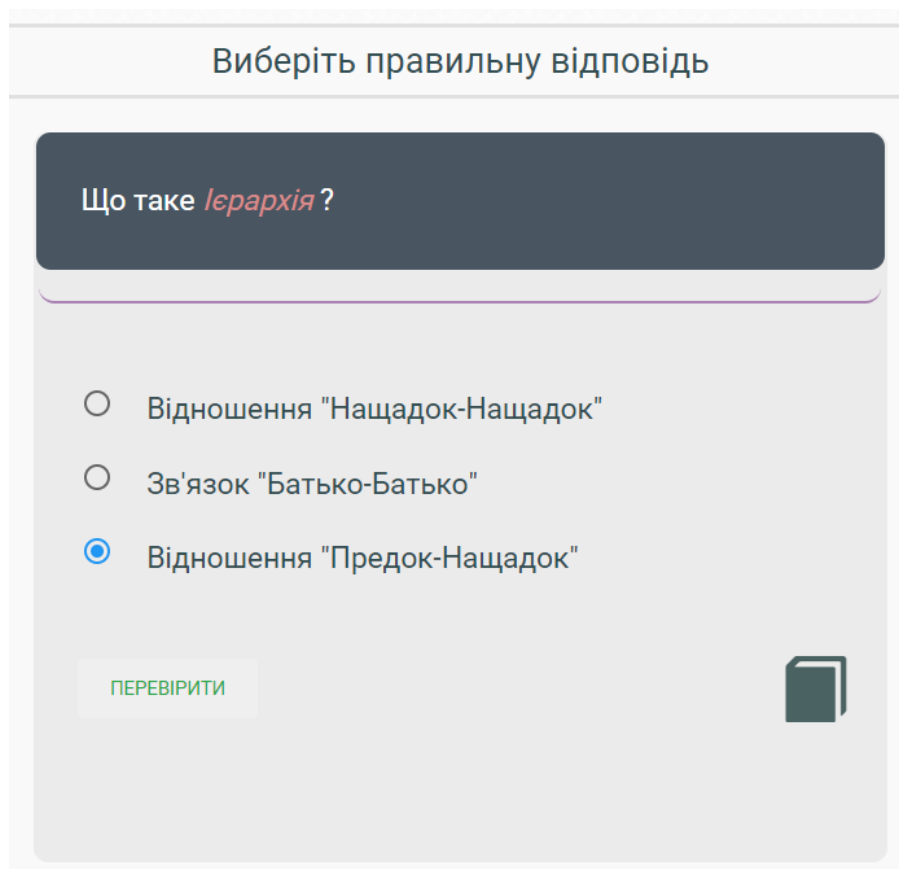


Рисунок В.8 – Крок 6 розділу завдань тренажеру «Кінематика пов'язаних об'єктів»

Впишіть числа в текстові поля (від 1 до 3)

Задайте правильну послідовність дій для включення режиму інверсної кінематики

<i>Створити об'єкти</i>	1
<i>Активувати інверсну кінематику</i>	3
<i>Налаштувати зв'язки об'єктів</i>	2


ПЕРЕВІРИТИ 

Рисунок В.9 – Крок 7 розділу завдань тренажеру «Кінематика пов'язаних об'єктів»

Виберіть правильну відповідь

Чим відрізняється *Інверсна кінематика* від *Прямої кінематики* ?

- Батько приводить в рух нащадків
- Нащадок приводить в рух батьків
- Відсутністю необхідності задавати ієрархію до включення режиму


ПЕРЕВІРИТИ 

Рисунок В.10 – Крок 8 розділу завдань тренажеру «Кінематика пов'язаних об'єктів»

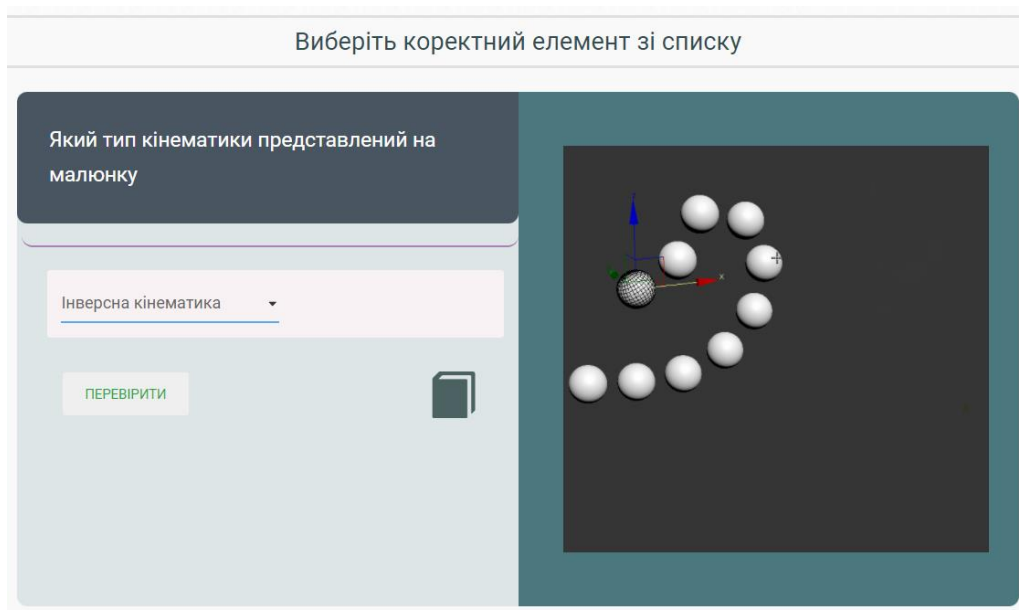


Рисунок В.11 – Крок 9 розділу завдань тренажеру «Кінематика пов'язаних об'єктів»

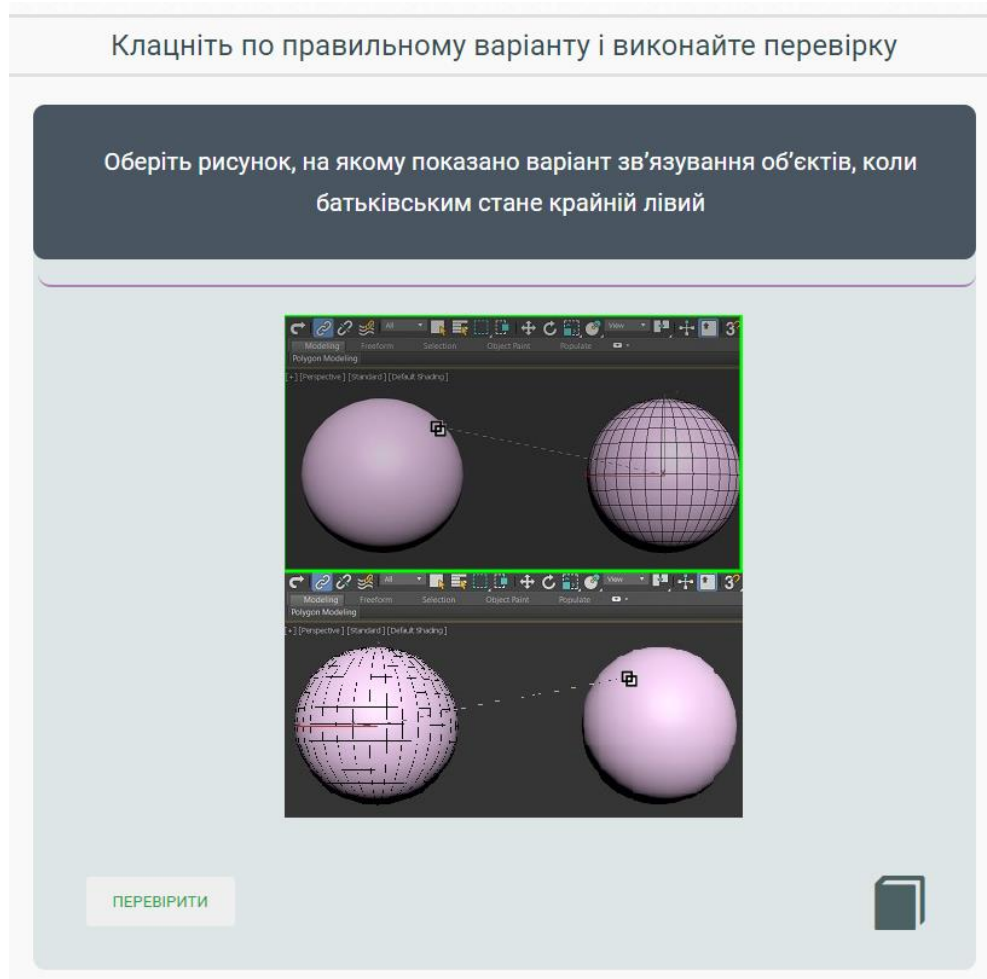


Рисунок В.12 – Крок 10 розділу завдань тренажеру «Кінематика пов'язаних об'єктів»

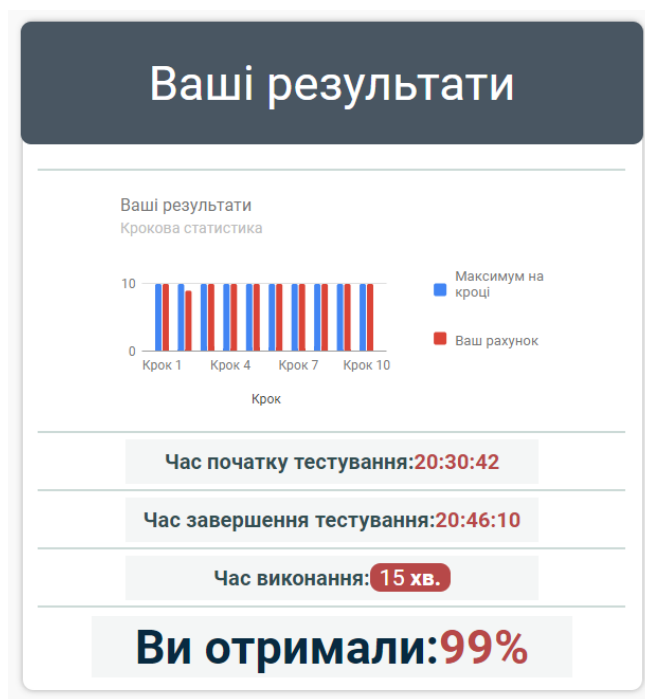


Рисунок В.13 – Розділ результатів тренажеру «Кінематика пов'язаних об'єктів»

Анімація об'єктів Bones

Вступ / Теорія / Крок 1 / Крок 2 / Крок 3 / Крок 4 / Крок 5 / Крок 6 / Крок 7 / Крок 8 / Крок 9 / Крок 10 / Результати

Анімація об'єктів Bones

Кількість кроків: 10

Оберіть мову:

ПОПЕРЕДНІЙ КРОК
ДОПОМОГА ВИКЛАДАЧА
ЗАВЕРШИТИ СЕАНС
НАСТУПНИЙ КРОК

Рисунок В.14 – Вступ тренажеру «Анімація об'єктів Bones»

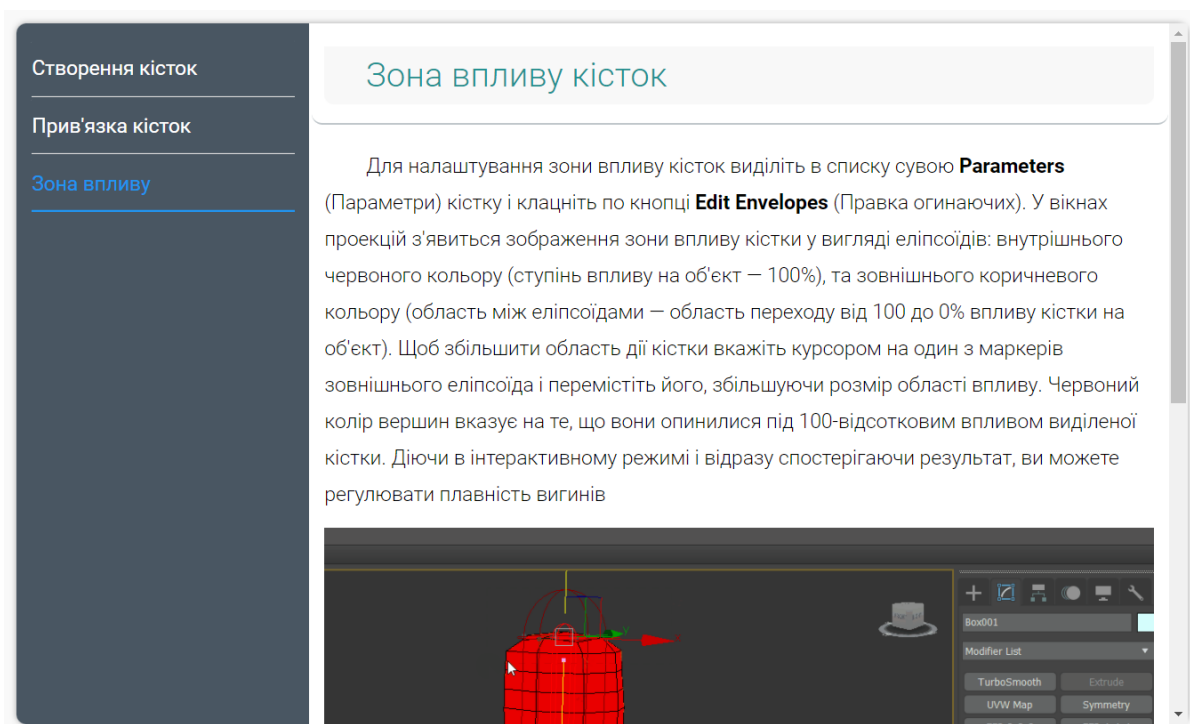


Рисунок В.15 – Теоретичний розділ тренажеру «Анімація об'єктів Bones»

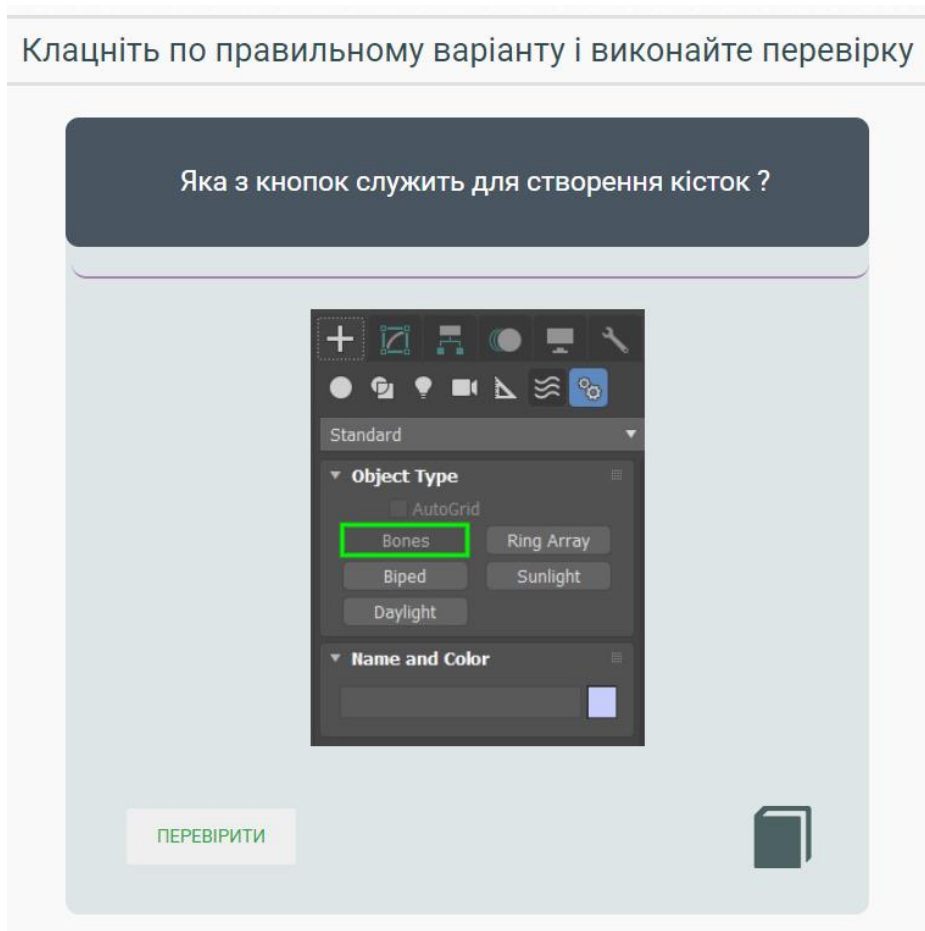


Рисунок В.16 – Крок 1 розділу завдань тренажеру «Анімація об'єктів Bones»



Рисунок В.17 – Крок 2 розділу завдань тренажеру «Анімація об'єктів Bones»

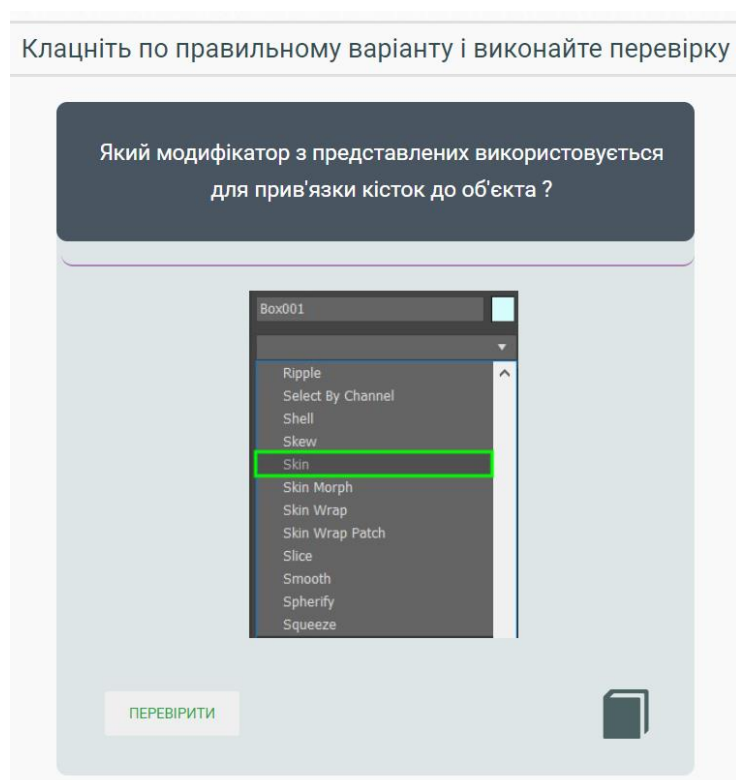


Рисунок В.18 – Крок 3 розділу завдань тренажеру «Анімація об'єктів Bones»

Клацніть по правильному варіанту і виконайте перевірку

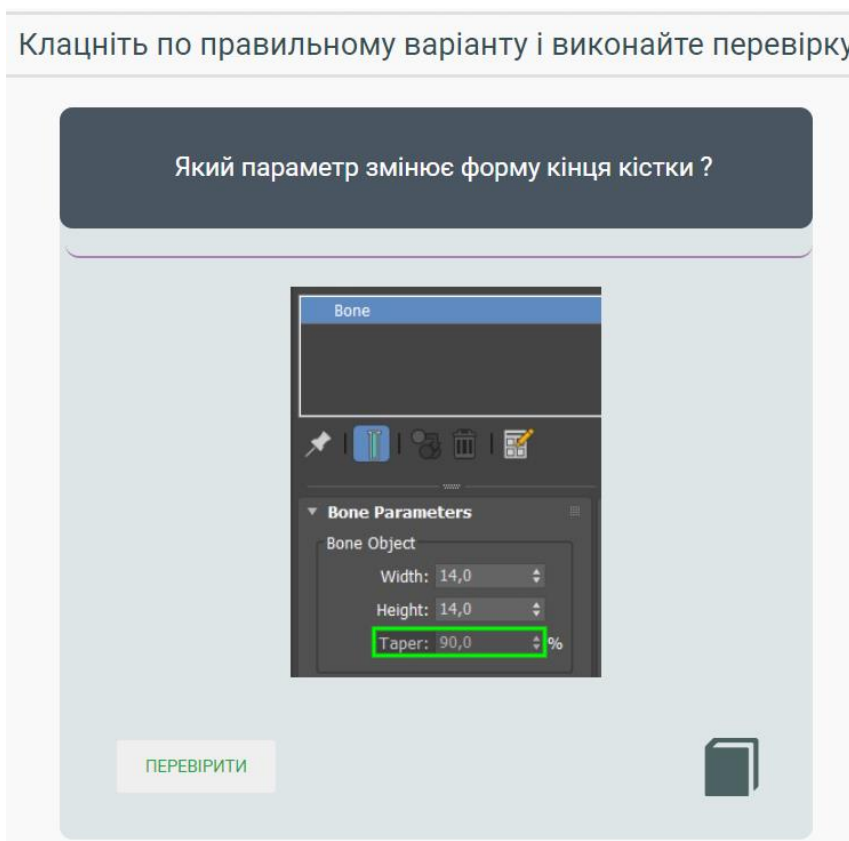


Рисунок В.19 – Крок 4 розділу завдань тренажеру «Анімація об'єктів Bones»

Впишіть одне слово в текстове поле (українською, в називному відмінку, регістр не має значення)

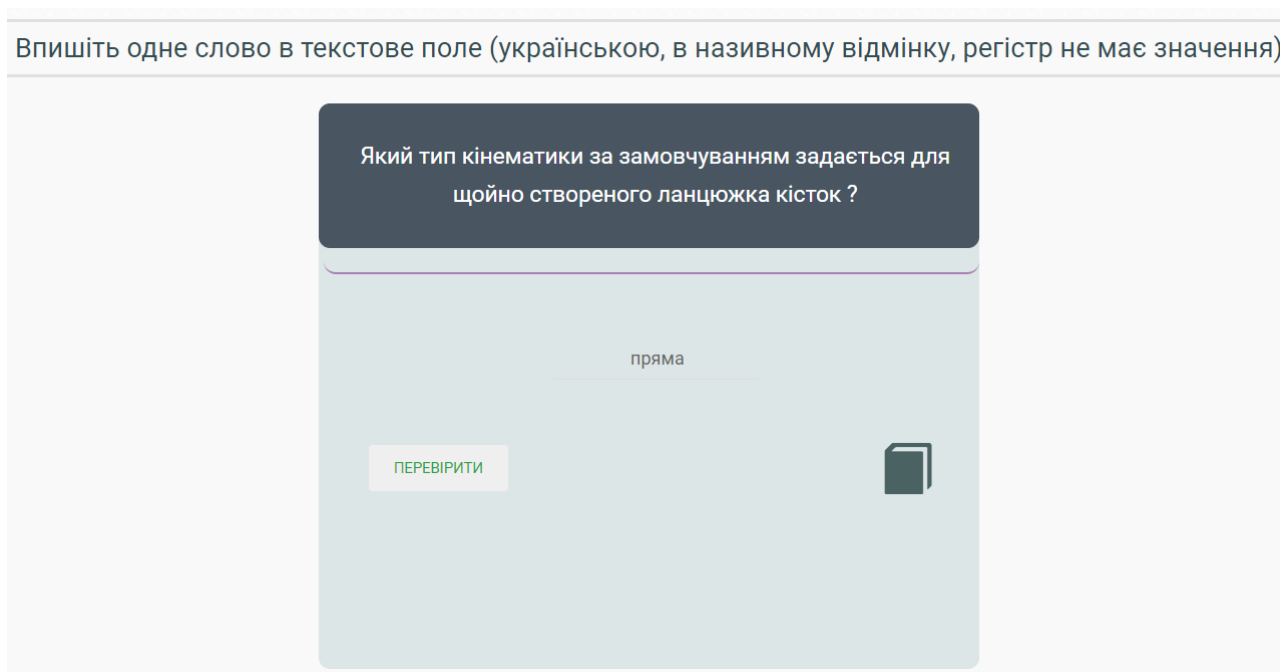


Рисунок В.20 – Крок 5 розділу завдань тренажеру «Анімація об'єктів Bones»

Виберіть правильну відповідь

Де повинні бути розташовані кістки відносно об'єкту прив'язки ?

З лівого краю
 Зовні
 З правого краю
 Всередині


ПЕРЕВІРИТИ 

Рисунок В.21 – Крок 6 розділу завдань тренажеру «Анімація об'єктів Bones»

Впишіть числа в текстові поля (від 1 до 5)

Задайте послідовність дій для прив'язки кісток до об'єкта

1	Виділити об'єкт
3	Клікнути по кнопці <i>Add</i>
2	Застосувати модифікатор <i>Skin</i>
5	Клікнути по кнопці <i>Select</i>
4	Вибрати кістки


ПЕРЕВІРИТИ 

Рисунок В.22 – Крок 7 розділу завдань тренажеру «Анімація об'єктів Bones»

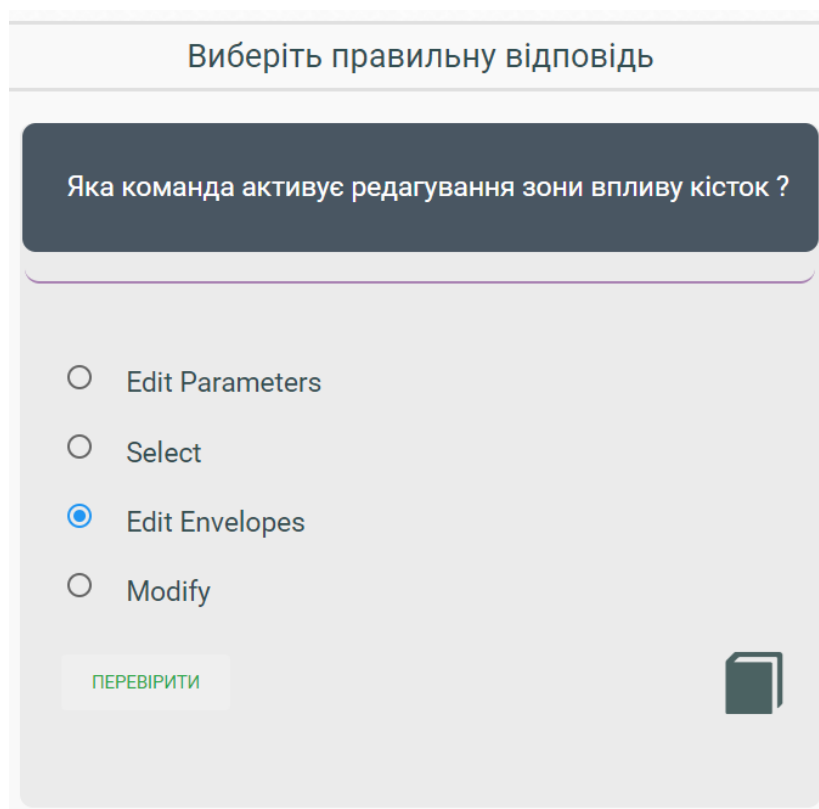


Рисунок В.23 – Крок 8 розділу завдань тренажеру «Анімація об'єктів Bones»

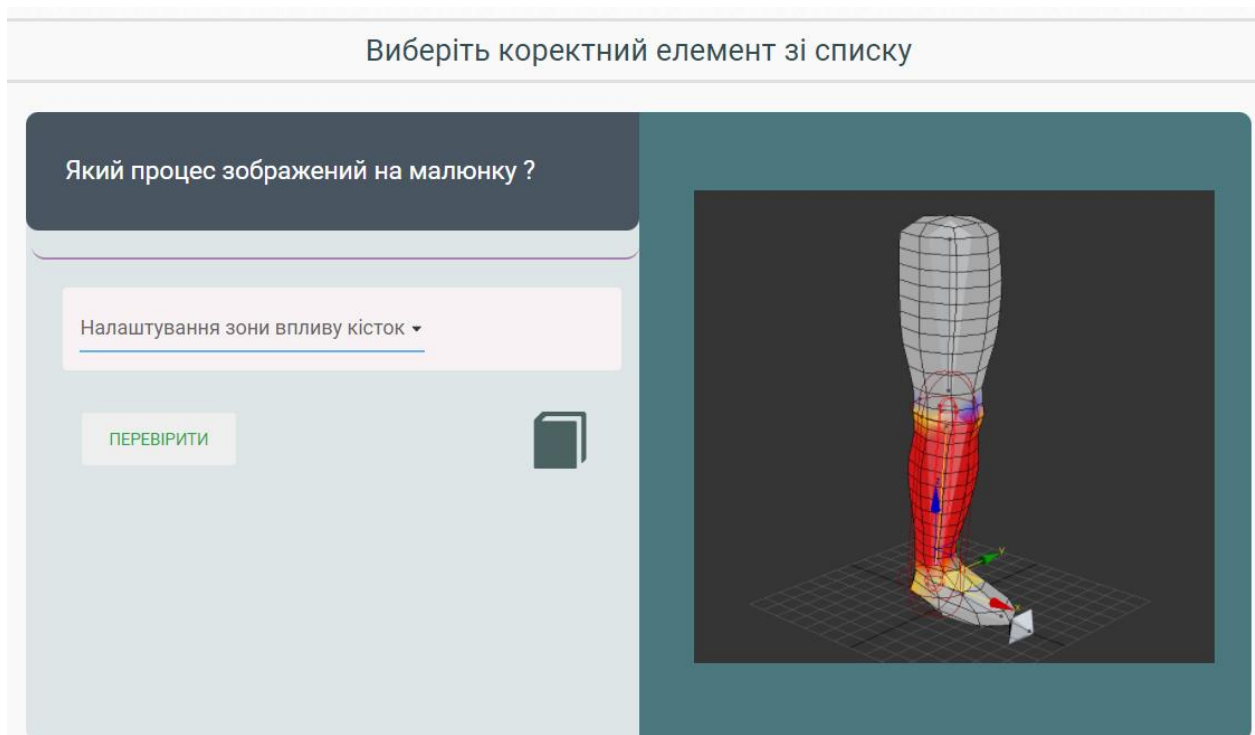


Рисунок В.24 – Крок 9 розділу завдань тренажеру «Анімація об'єктів Bones»

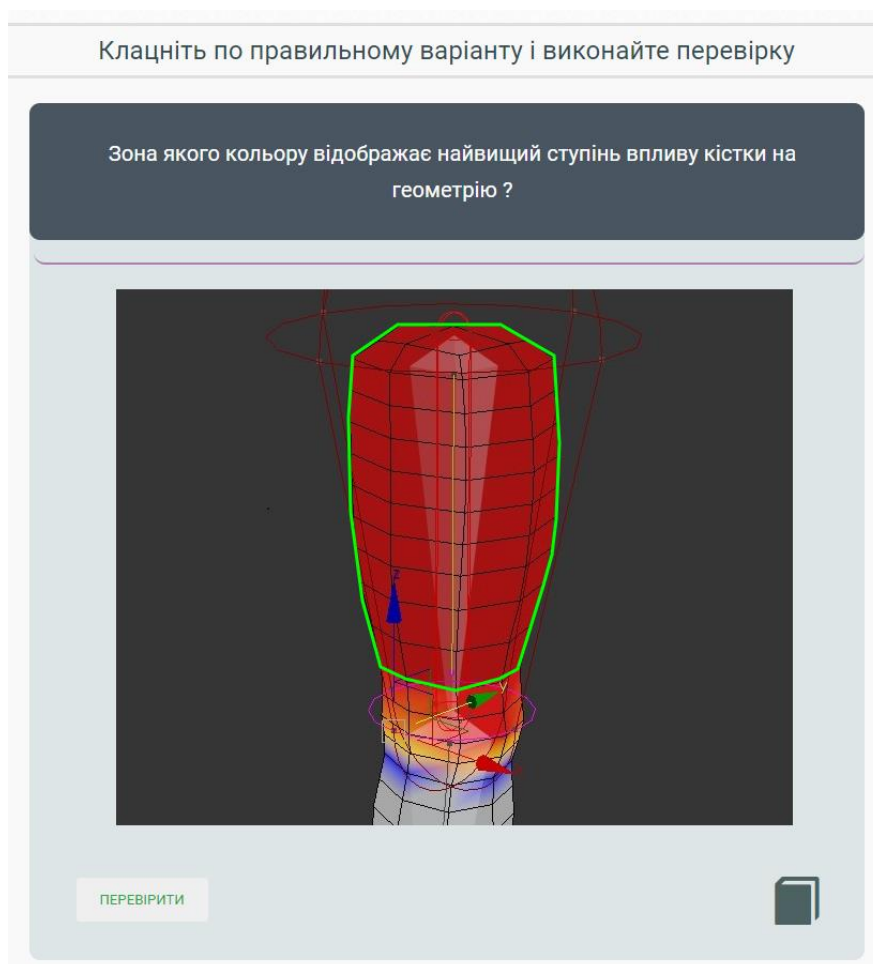


Рисунок В.25 – Крок 10 розділу завдань тренажеру «Анімація об'єктів Bones»

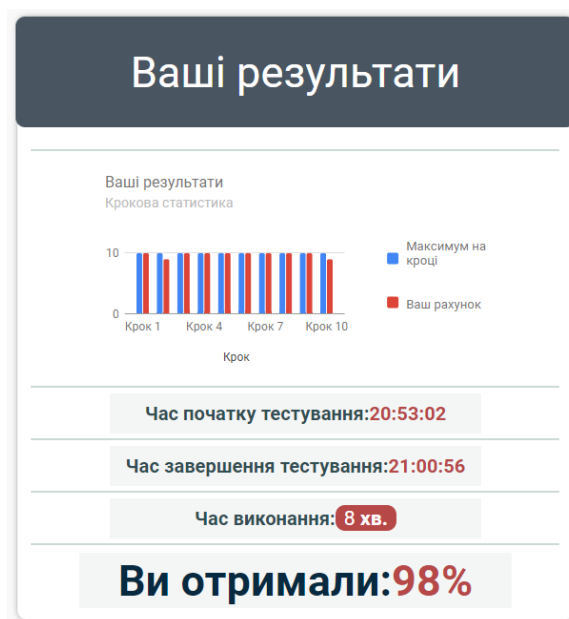


Рисунок В.26 – Розділ результатів тренажеру «Анімація об'єктів Bones»

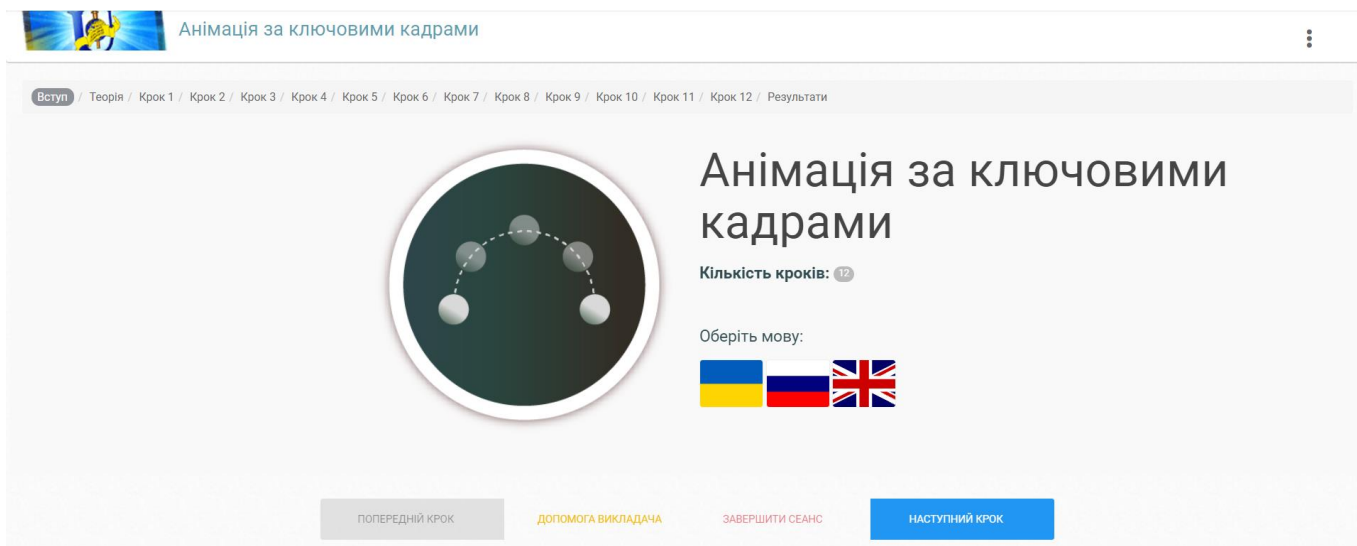


Рисунок В.27 – Вступ тренажеру «Анімація за ключовими кадрами»



Рисунок В.28 – Теоретичний розділ тренажеру «Анімація за ключовими кадрами»

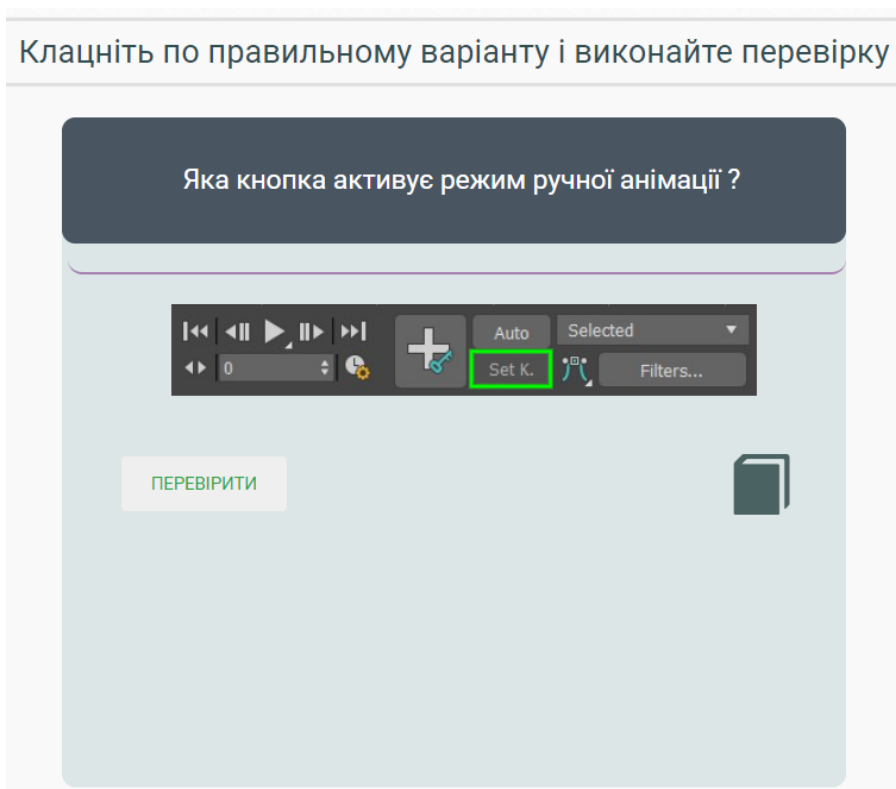


Рисунок В.29 – Крок 1 розділу завдань тренажеру «Анімація за ключовими кадрами»

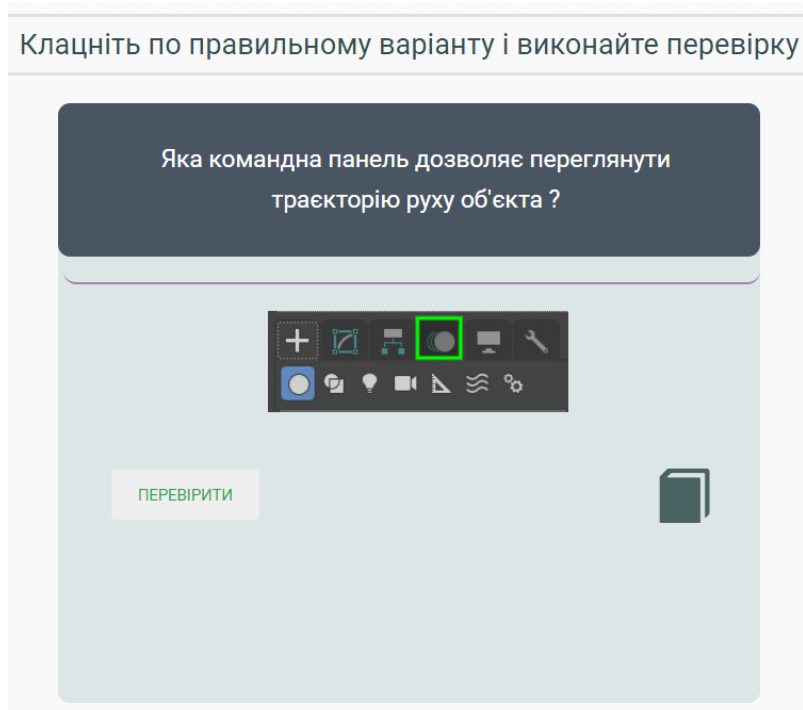


Рисунок В.30 – Крок 2 розділу завдань тренажеру «Анімація за ключовими кадрами»

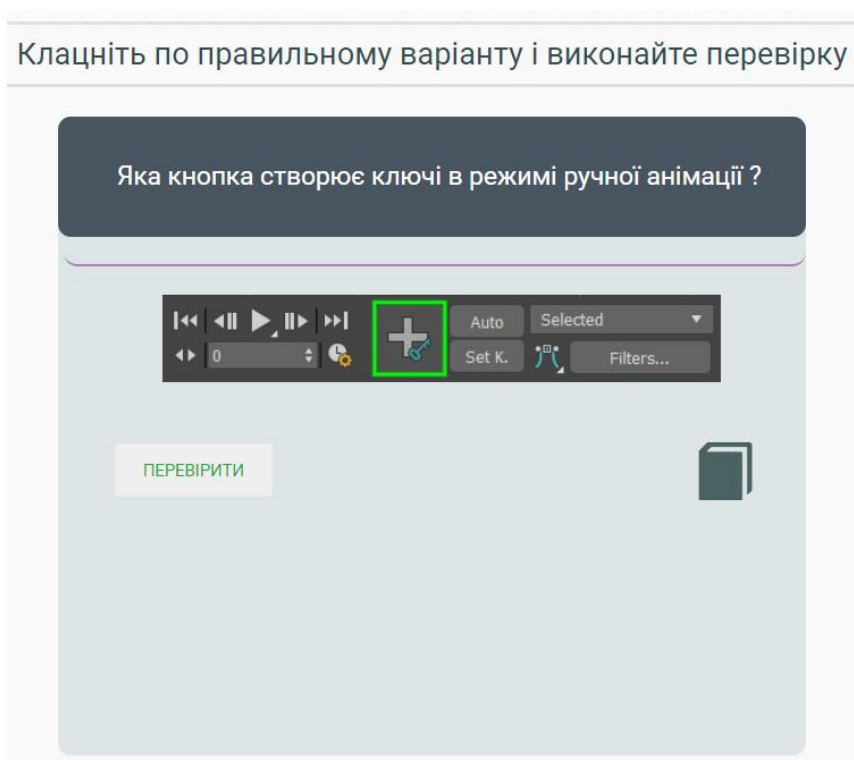


Рисунок В.31 – Крок 3 розділу завдань тренажеру «Анімація за ключовими кадрами»

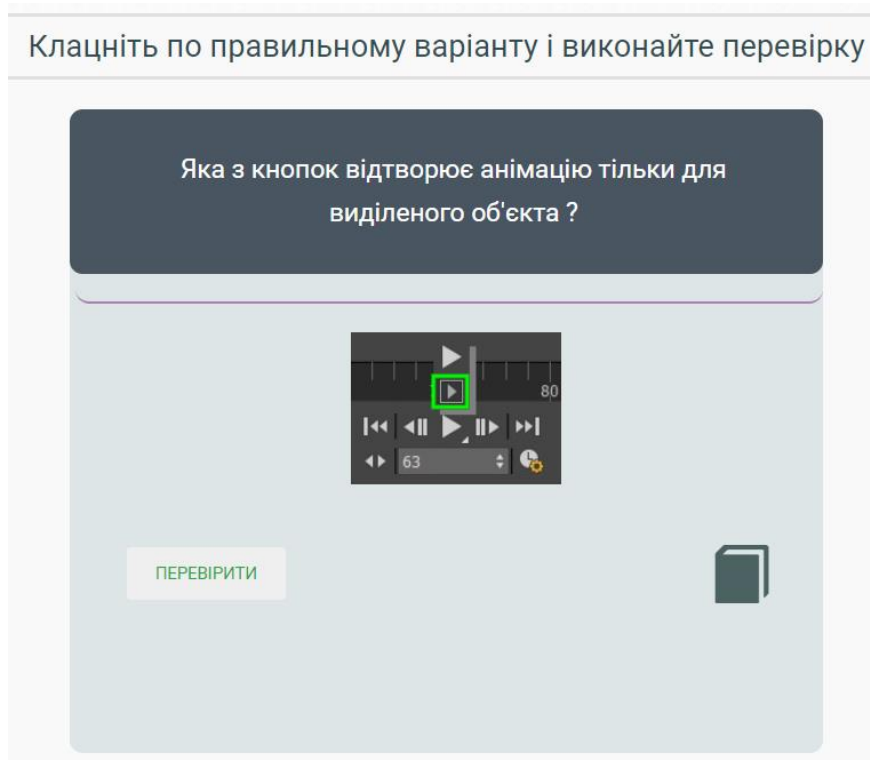


Рисунок В.32 – Крок 4 розділу завдань тренажеру «Анімація за ключовими кадрами»

Впишіть одне слово в текстове поле (українською, в називному відмінку, регістр не має значення)

Режим анімації, при якому ключі анімації створюються самі ?

автоматичний

ПЕРЕВІРИТИ

The screenshot shows a quiz question in a training simulator. The question is: "Режим анімації, при якому ключі анімації створюються самі ?" (Animation mode, in which animation keys are created themselves?). The answer is "автоматичний" (automatic). There is a "ПЕРЕВІРИТИ" (CHECK) button and a book icon.

Рисунок В.33 – Крок 5 розділу завдань тренажеру «Анімація за ключовими кадрами»

Виберіть правильну відповідь

Що робить команда *Set Key Filters* ?

- Скасовує створення ключів для параметрів, зазначених прапорцями
- Зазначає, для яких параметрів повинні створюватися ключі анімації
- Видаляє повністю всі ключі анімації на часовій шкалі

ПЕРЕВІРИТИ

The screenshot shows a quiz question in a training simulator. The question is: "Що робить команда *Set Key Filters* ?" (What does the *Set Key Filters* command do?). The correct answer is "Зазначає, для яких параметрів повинні створюватися ключі анімації" (Specifies for which parameters animation keys should be created). There are three radio button options. There is a "ПЕРЕВІРИТИ" (CHECK) button and a book icon.

Рисунок В.34 – Крок 6 розділу завдань тренажеру «Анімація за ключовими кадрами»

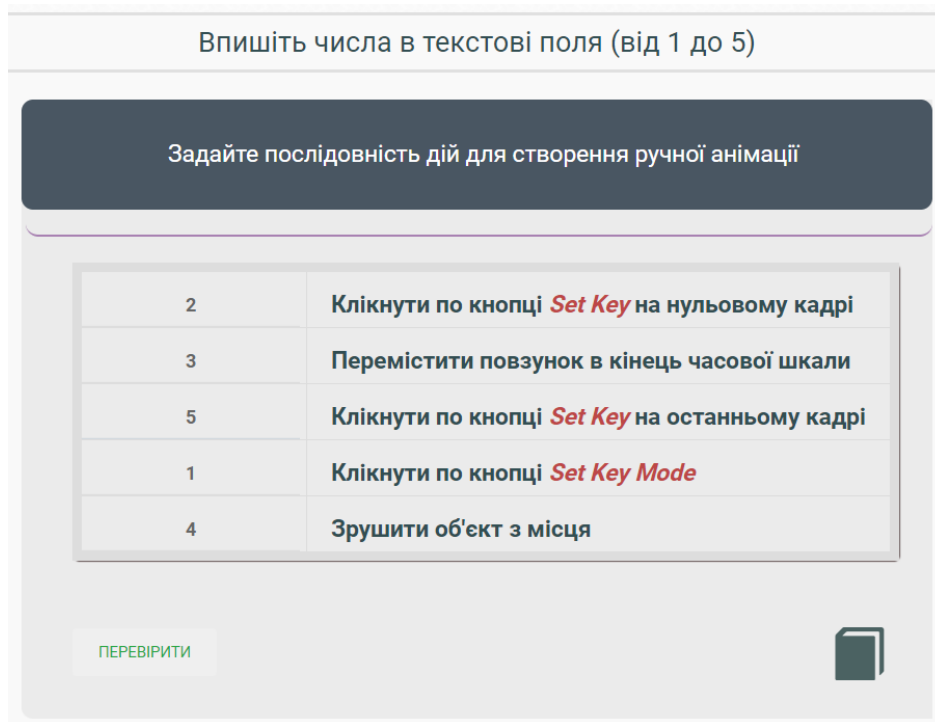


Рисунок В.35 – Крок 7 розділу завдань тренажеру «Анімація за ключовими кадрами»

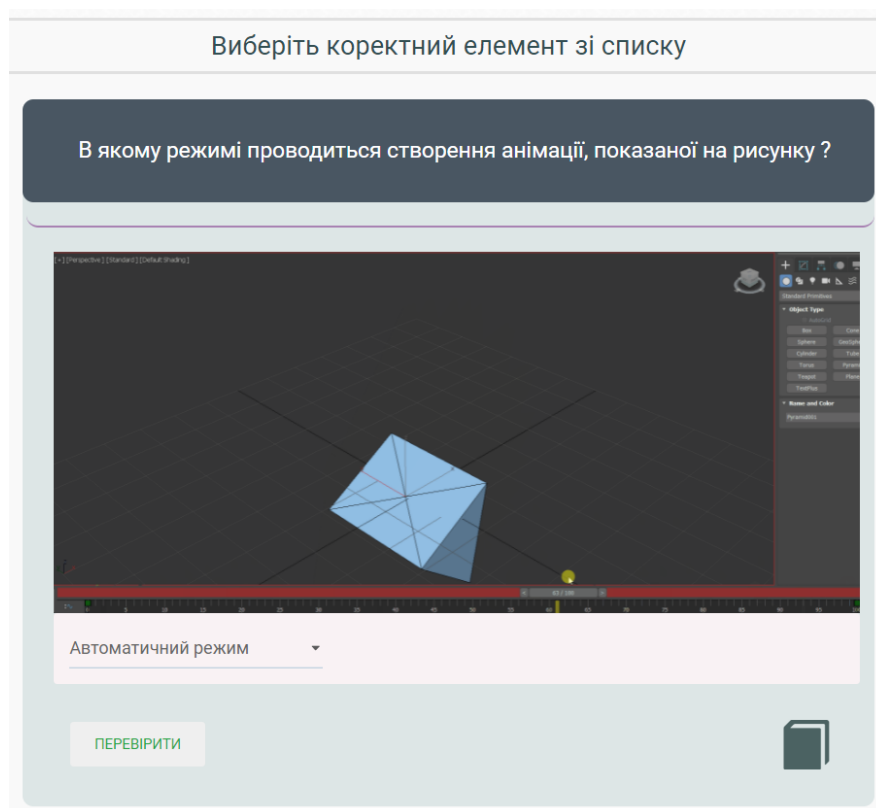


Рисунок В.36 – Крок 8 розділу завдань тренажеру «Анімація за ключовими кадрами»

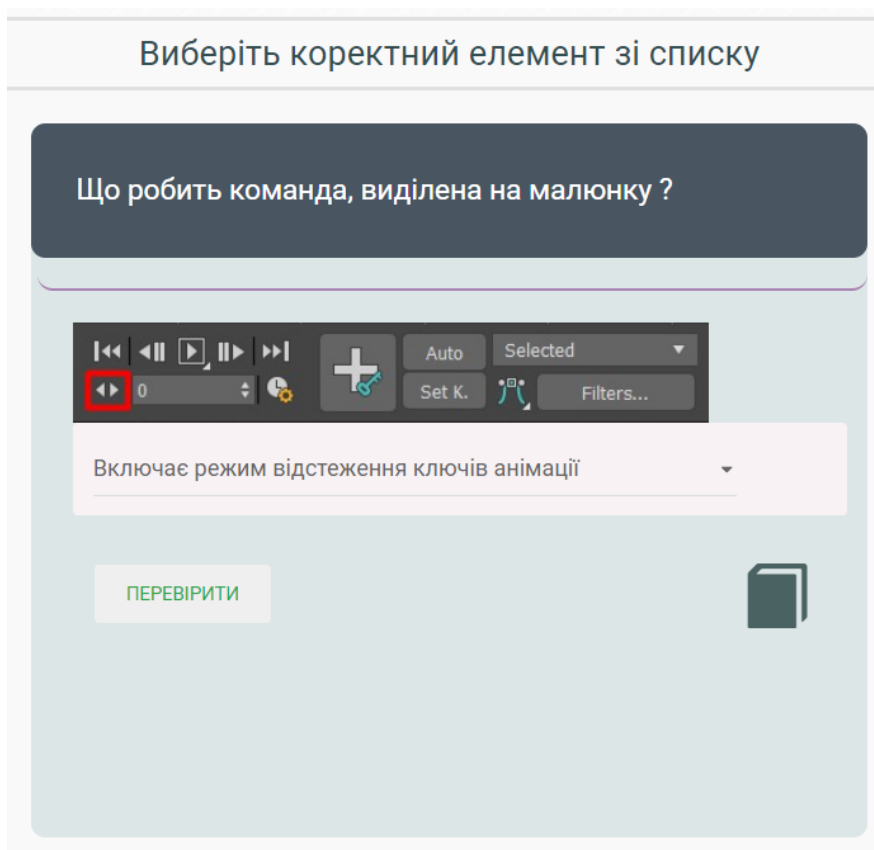


Рисунок В.37 – Крок 9 розділу завдань тренажеру «Анімація за ключовими кадрами»

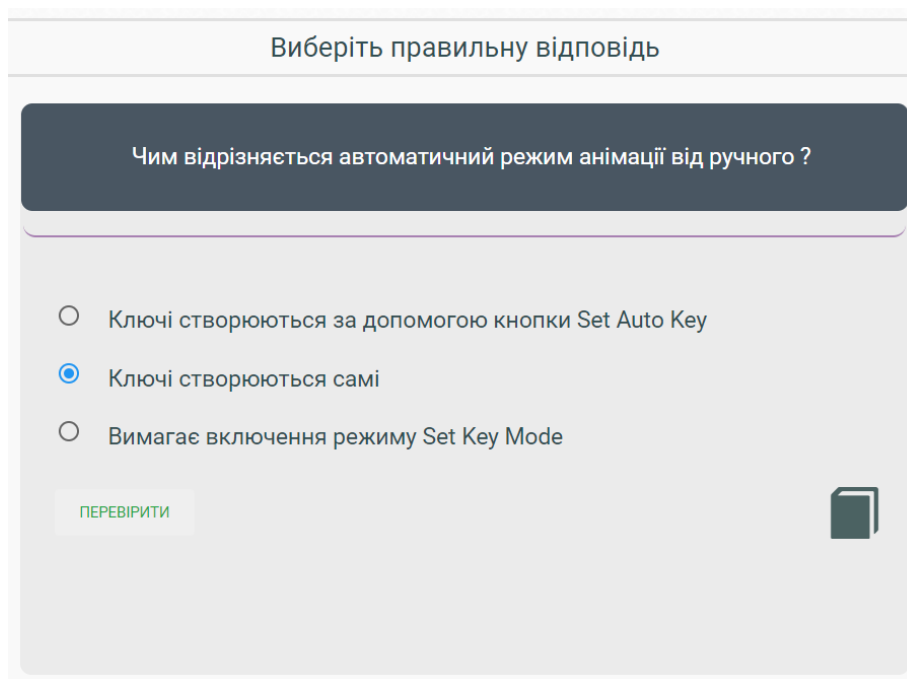


Рисунок В.38 – Крок 10 розділу завдань тренажеру «Анімація за ключовими кадрами»

Перетягніть в поле блоки з правильними відповідями (щоб прибрати блок з поля - клацніть по блоку)

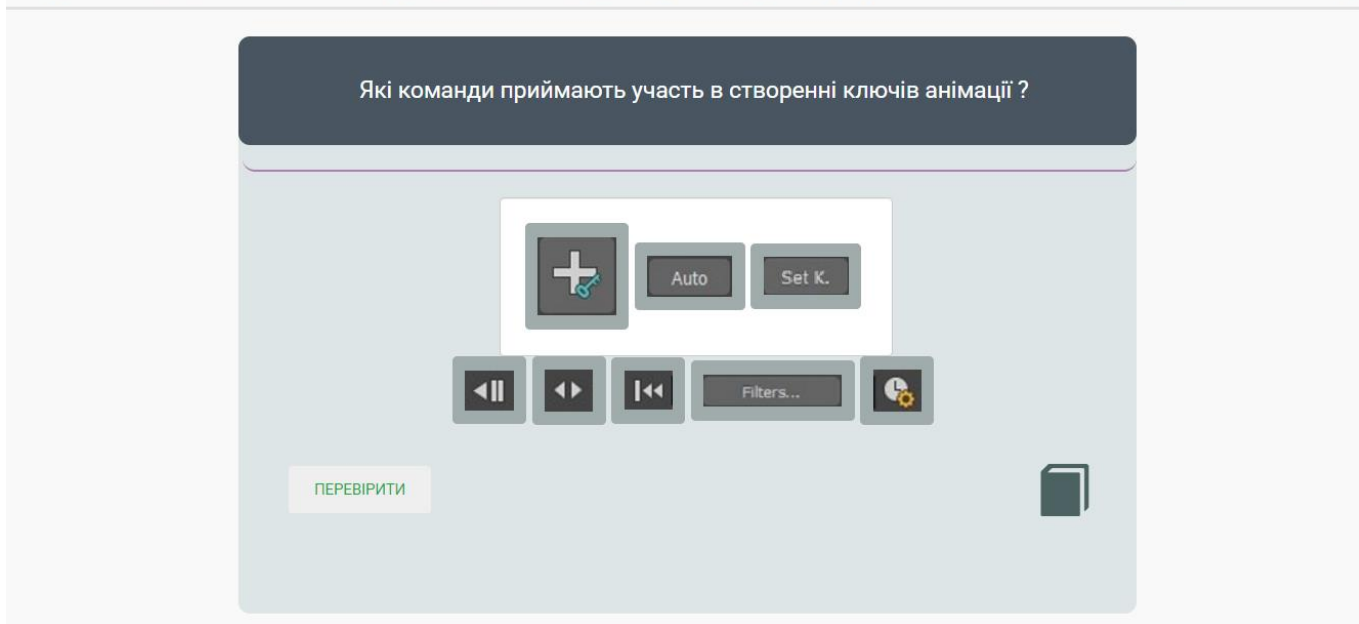


Рисунок В.39 – Крок 11 розділу завдань тренажеру «Анімація за ключовими кадрами»

Перетягніть в поле блоки з правильними відповідями (щоб прибрати блок з поля - клацніть по блоку)

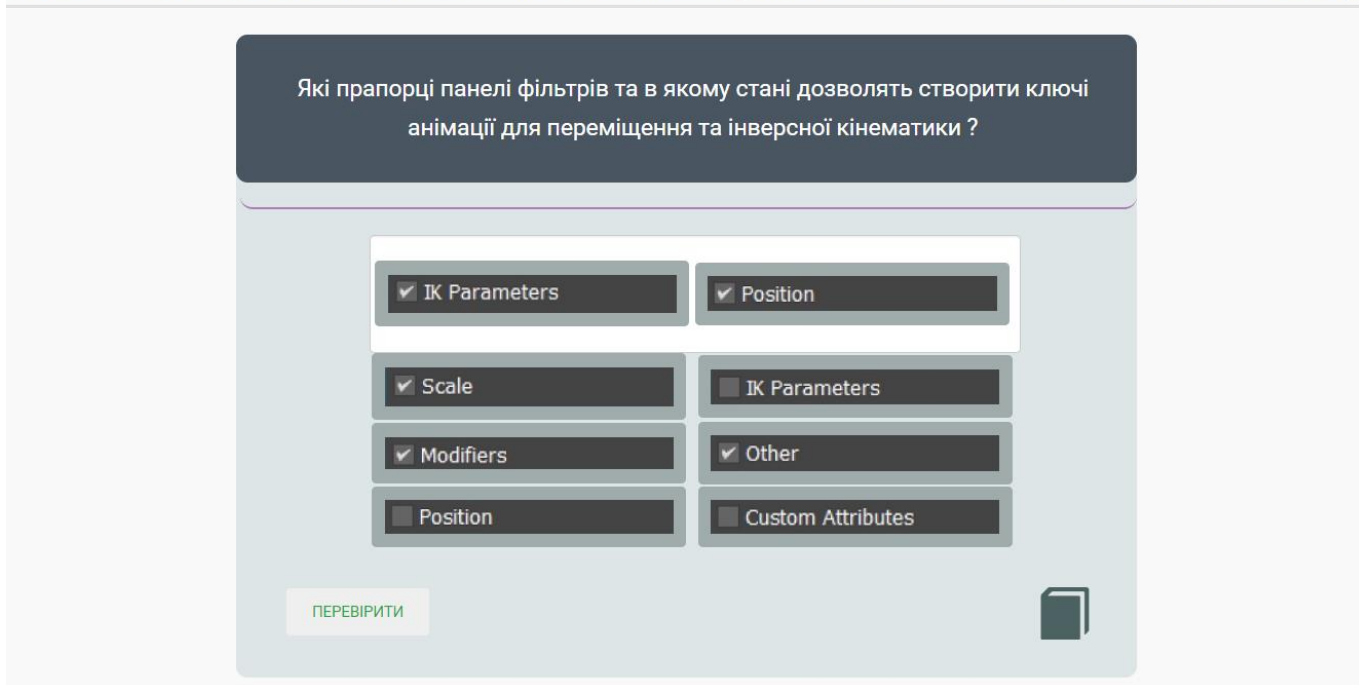


Рисунок В.40 – Крок 12 розділу завдань тренажеру «Анімація за ключовими кадрами»

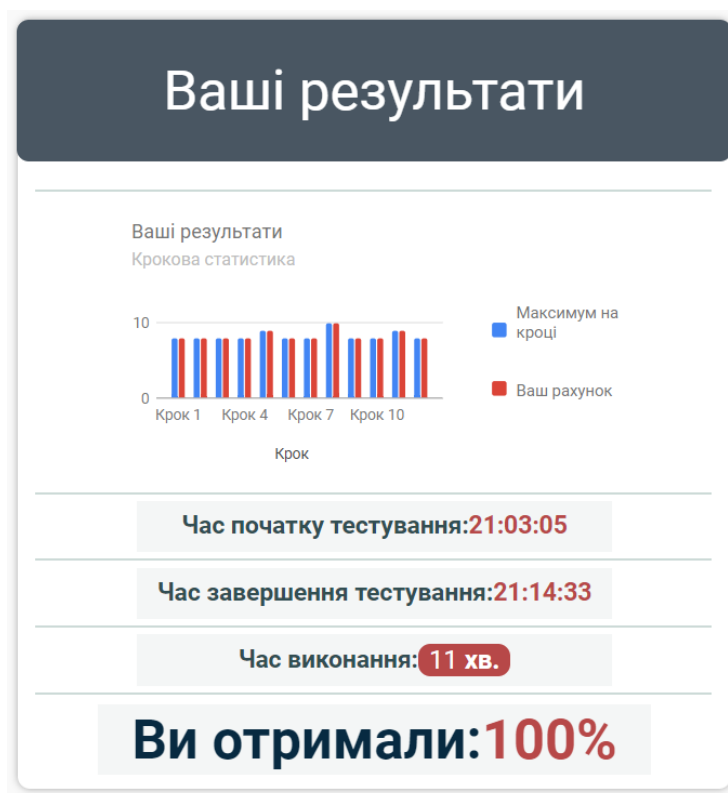


Рисунок В.41 – Розділ результатів тренажеру «Анімація за ключовими кадрами»