

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК  
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Комплекс інтерактивних тренажерів для дистанційного курсу з вивчення 3ds Max»

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

Виконавець роботи: студент групи ІТ-71 Ісаєв Вадим В'ячеславович

Кваліфікаційна робота бакалавра  
захищена на засіданні ЕК  
з оцінкою \_\_\_\_\_

«\_\_» \_\_\_\_\_ 2021 р.

Науковий керівник \_\_\_\_\_

(підпис)

к.т.н., доц., Баранова І.В.

(науковий ступінь, вчене звання, прізвище та ініціали)

Голова комісії \_\_\_\_\_

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвище та ініціали)

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 комп'ютерних наук  
Секція інформаційних технологій проектування  
Спеціальність 122 «Комп'ютерні науки»  
Освітньо-професійна програма «Інформаційні технології проектування»

**ЗАТВЕРДЖУЮ**  
Зав. секцією ІТП

\_\_\_\_\_  
В. В. Шендрик  
«\_\_» \_\_\_\_\_ 2021 р.

## **З А В Д А Н Н Я**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ**

*Ісаєв Вадим В'ячеславович*

**1 Тема роботи** Комплекс інтерактивних тренажерів для дистанційного курсу з вивчення 3ds Max

**керівник роботи** Баранова Ірина Володимирівна, к.т.н., доцент

затверджені наказом по університету від «14» квітня 2021 р. № 0181-VI

**2 Строк подання студентом роботи** «1» червня 2021 р.

**3 Вхідні дані до роботи** технічне завдання на розробку інтерактивних тренажерів

**4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)** аналіз предметної області, функціональне і структурне проектування додатку, практична реалізація інтерактивного веб-додатку

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)** актуальність роботи, постановка задачі, дослідження аналогів, функціональні вимоги до веб-тренажеру, засоби реалізації, моделювання, практична реалізація, тестування

## 6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01.10.2020

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	01.10.2020 – 10.10.2020	
2	Огляд існуючих аналогів	10.10.2020 – 16.10.2020	
3	Постановка задачі	16.10.2020 – 21.10.2020	
4	Функціональні вимоги	21.10.2020 – 28.10.2020	
5	Вибір технологій реалізації	28.10.2020 – 10.11.2020	
6	Структурно-функціональне моделювання	10.11.2020 – 02.12.2020	
7	Розробка інтерфейсу веб-тренажерів	02.12.2020 – 20.12.2020	
8	Наповнення веб-тренажерів	10.11.2020 – 29.11.2020	
9	Розробка всіх кроків веб-тренажерів	29.12.2020 – 23.02.2021	
10	Локалізація інтерфейсу	23.02.2021 – 27.02.2021	
11	Тестування додатку	27.02.2021 – 15.03.2021	
12	Оформлення документації	15.03.2021 – 03.05.2021	
13	Інтегрування веб-тренажерів до elearning	03.05.2021 – 14.05.2021	
14	Перевірка працездатності розробленої системи	14.05.2021 – 01.06.2021	

Студент

\_\_\_\_\_

(підпис)

Ісаєв В.В.

Керівник роботи

\_\_\_\_\_

(підпис)

к.т.н., доц. Баранова І.В..

## РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Комплекс інтерактивних тренажерів для дистанційного курсу з вивчення 3ds Max».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 23 найменувань, 4 додатків. Загальний обсяг роботи – 78 сторінок, в тому числі 40 сторінка основного тексту, 2 сторінки списку джерел та 36 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці інтерактивних веб-тренажерів з дисципліни «Комп'ютерні технології дизайну» для студентів дистанційної форми навчання СумДУ.

В першому розділі було проаналізовано предметну область, актуальність роботи та аналоги. Обрано засоби для реалізації проекту.

В другому розділі виконано моделювання роботи веб-тренажеру, використовуючи методології IDEF та UML.

Третій розділ описує процес розробки веб-тренажера. Наведена архітектура додатка та основні деталі програмної реалізації. Продемонстровано приклад використання готового веб-тренажеру та його працездатність.

Результатом виконаної роботи є два справні інтерактивні веб-тренажери для вивчення 3DS Max з курсу «Комп'ютерні технології дизайну» для системи дистанційної освіти СумДУ.

Ключові слова: тренажер, веб-додаток, SVG, JavaScript, 3ds Max, 3D графіка, JSON.

## ЗМІСТ

Вступ .....	6
1 Аналіз проблеми розробки тренажерів .....	8
1.1 Огляд актуальних досліджень і публікацій .....	8
1.2 Аналіз програмних продуктів – аналогів .....	9
1.3 Постановка задачі .....	10
2 Проектування комплексу тренажерів .....	13
2.1 Структурно-функціональне моделювання .....	13
2.2 Проектування моделі бази даних .....	20
3 Практична реалізація комплексу тренажерів.....	22
3.1 Архітектура програмного додатку .....	22
3.2 Структура проекту .....	23
3.3 Програмна реалізація .....	26
3.4 Перевірки працездатності тренажерів .....	30
3.5 Використання тренажерів .....	32
Висновки.....	40
Список використаних джерел.....	41
Додаток А Технічне завдання.....	43
Додаток Б Планування робіт .....	51
Додаток В Лістинг коду .....	56
Додаток Г Акт впровадження.....	78

## ВСТУП

З кожним роком система навчання стає все більш залежною від інформаційних технологій, і нещодавно світ в цьому переконався. Пандемія змусила перейти всі навчальні заклади світу на дистанційне навчання. Саме змога проводити заняття і не зупиняти навчальний процес є заслугами інформаційних технологій, таких як: засоби інтернет-зв'язку, хмарні сховища, веб тренажери та багато інших. За допомогою систем інтернет-зв'язку студенти мають можливість дистанційно спілкуватися з викладачами, за допомогою хмарних сховищ – обмінюватися даними, а за допомогою веб-тренажерів викладачі можуть навчати студентів та оцінювати їх знання.

Навчальна дисципліна «Комп'ютерна графіка» в цілому, та її напрямок 3D моделювання, зокрема, є одними із найбільш цікавих та затребуваних. 3D моделювання застосовується у багатьох сферах людської діяльності – у промисловості, медицині, будівництві тощо. І для нього необхідне професійне володіння спеціальними графічними редакторами. Серед таких одним із найбільш поширених є програма 3DS Max.

Саме розробка інтерактивних веб-тренажерів для підтримки навчання роботи з програмою 3DS Max і є темою роботи.

Інтерактивний веб-тренажер являє собою програмний комплекс, метою якого є імітація взаємодії людини з реальною системою. При використанні інтерактивних веб-тренажерів контроль якості знань відходить від того, щоб бути узагальненим та стає індивідуальним для кожного студента, враховуючи темп навчання кожного. Також автоматизований контроль підвищує об'єктивність самого контролю та дозволяє оцінювати якість знань студентів не тільки поверхнево, але й забезпечує кількісну оцінку якості засвоєння студентами того чи іншого навчального розділу [1].

Отже, актуальність роботи обґрунтовується сучасними напрямками розвитку інформаційних технологій у системі освіти та попит на фахівців з тривимірної комп'ютерної графіки.

Об'єктом дослідження роботи є інтеграція інтерактивних веб-тренажерів у процес навчання студентів.

Предметом дослідження роботи є використання інтерактивних веб-тренажерів при навчанні студентів.

Метою роботи є створення комплексу інтерактивних веб-тренажерів, які повинні імітувати роботу користувача з програмним продуктом 3DS Max.

Основними задачами роботи є:

- ознайомлення з літературою на тему дослідження;
- вибір технологій для реалізації поставленої задачі;
- створення сценаріїв проходження тренажеру;
- підготовка ресурсів та обладнання для розробки;
- створення інтерфейсу користувача;
- написання скриптів проходження тренажеру.

Практичне значення роботи полягає в тому, що впровадження розроблених тренажерів в систему дистанційної освіти університету дозволить закріпити навички роботи з програмою 3DS Max.

# 1 АНАЛІЗ ПРОБЛЕМИ РОЗРОБКИ ТРЕНАЖЕРІВ

## 1.1 Огляд актуальних досліджень і публікацій

Комп'ютерна графіка є однією з ІТ галузей, яка активно розвивається. Одним із розділів комп'ютерної графіки є 3D графіка, призначення якої є зображення об'єктів у тривимірному просторі. 3D графіка використовується в: кінематографі, архітектурі, комп'ютерних іграх, науці, медицині та в інших галузях [2].

Найвідомішим програмним забезпеченням по роботі із 3D графікою є 3DS Max компанії Autodesk. Відрізняється від аналогів своїм широким функціоналом, тому й має широкий діапазон використання: від формування дизайну інтер'єрів до створення моделей для комп'ютерних ігор [3]. Саме для 3DS Max і буде створюватися інтерактивний веб тренажер у цій роботі.

Інтерактивний тренажер є програмно-апаратним комплексом, який дозволяє імітувати роботу з реальною установкою без посереднього контакту з нею [4].

Оскільки у тренажерів процес взаємодії з користувачем буде відбуватися через зображення інтерфейсу 3DS Max, то для цього потрібен інструмент, який дозволить зробити необхідні області зображення інтерактивними. Через те, що проект є веб-орієнтованим та буде розроблятися на основі наявного шаблону, написаному з використанням стандартних засобів веб-розробки HTML, CSS і JavaScript, то потрібна технологія, яка може бути інтегрована в систему шаблону.

Сьогодні існує багато JavaScript бібліотек, які дозволяють вбудовувати зображення в розмітку та виділяти інтерактивні області з подальшою стилізацією [5]. Серед них: Snap.Svg, SVG.JS, Two.js та інші.

Також в сучасному HTML є потужний вбудований засіб по роботі з SVG форматом, що дозволяє працювати з векторною графікою, яка може бути нанесена на зображення [6].



Для того, щоб розмітити області на зображеннях та експортувати координати розміченої області потрібен графічний редактор, який має відповідні функції. Серед графічних редакторів з потрібним функціоналом найбільш поширеними є: Adobe Illustrator [7], SvgPathEditor [8] та Inkscape [9].

## 1.2 Аналіз програмних продуктів – аналогів

Для розробки тренажеру необхідний індивідуальний підхід і досвід через його вузьку спеціалізацію та різноплановість інформації, яку візуалізує тренажер. Кожен крок тренажеру відрізняється один від одного своїм змістом та даними, які він проектує, тому кожен крок повинен бути продуманий та створений вручну.

На цей час не існує програмних продуктів аналогів, які б повністю відповідали темі дослідження, або таких, за допомогою яких можна б було створити проект, подібний до розроблюваного.

Тому розглянемо проекти, найбільш близькі до теми роботи.

Прикладом веб-додатку для тестування, який відповідає темі розроблюваного тренажера, є комплекс тестів курсу по 3DS Max [10]. В тестах користувачу необхідно обрати правильний варіант відповіді на поставлене питання.

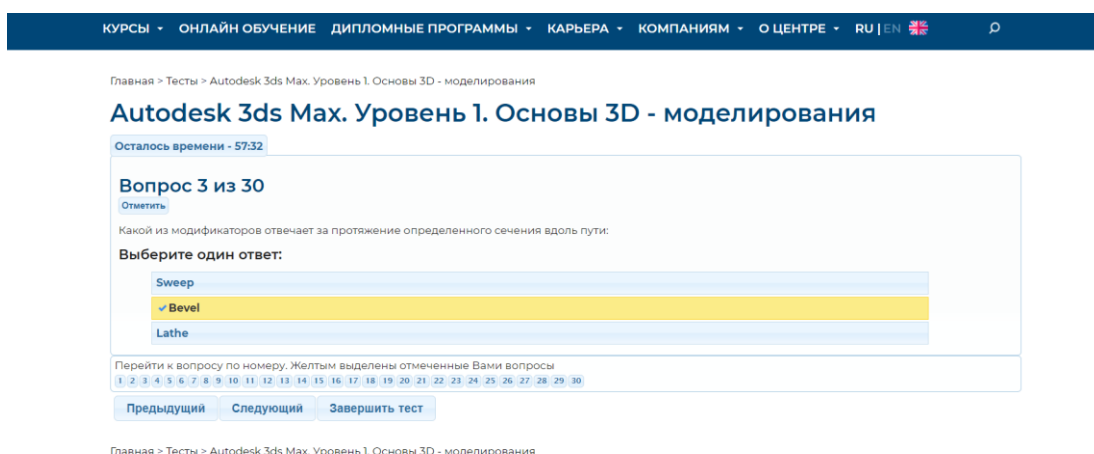


Рисунок 1.1 – Приклад тренажеру з 3DS Max

Даний додаток не надає можливості вибору варіантів, повторної спроби відповіді на невірне питання. Також він не має інтерактивних зображень.

Прикладом тренажеру, який є функціонально схожим до розроблюваного є інтерактивний тренажер з перевірки знань по географії [11]. Користувач повинен шукати на карті країни згідно з заданими критеріями. Продукт не відповідає темі 3DS Max та не може бути використаний для створення тренажеру.

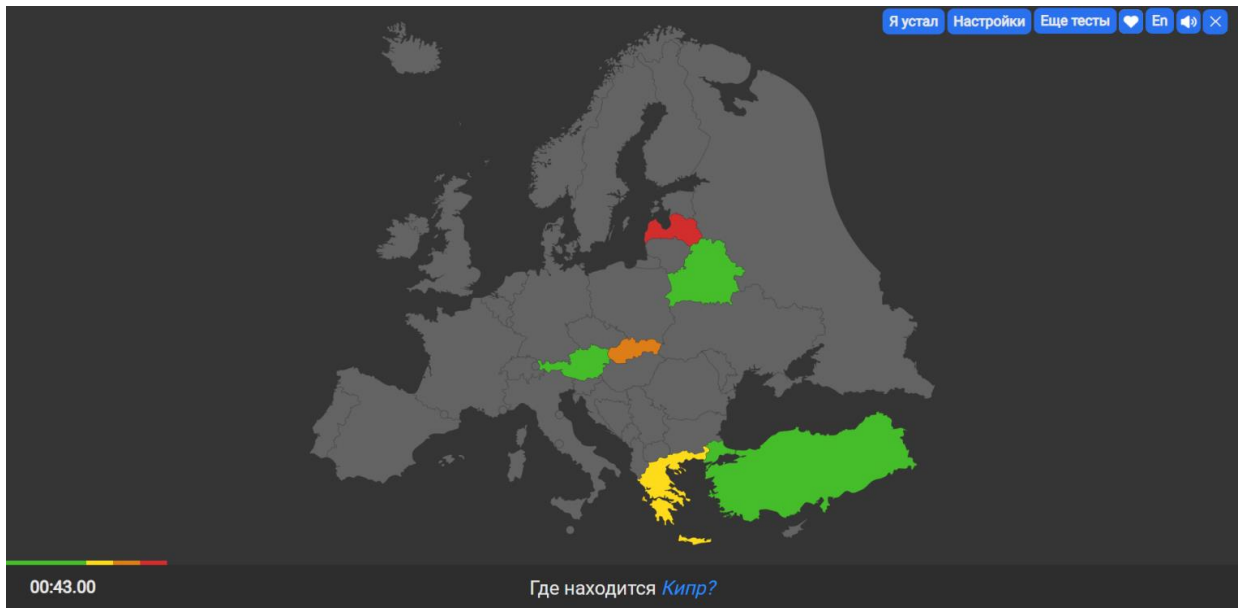


Рисунок 1.2 – Приклад інтерактивного тренажеру для перевірки знань з географії

Також можна проаналізувати розробку тренажеру для дистанційного курсу «Комп'ютерна графіка» [12]. Даний проект схожий за структурою, але він містить лише один тренажер з дисципліни, не має можливості вибору варіантів, і також не має можливості працювати інтерактивно з зображенням.

### 1.3 Постановка задачі

Метою проекту є створення комплексу інтерактивних веб-тренажерів для дистанційного курсу з вивчення роботи з програмним продуктом 3DS Max. Інтерактивні тренажери симулюють взаємодію користувача з програмою.

Передбачається створення тренажерів з вивчення тем «Створення стандартних примітивів» та «Дії з об'єктами» для дистанційного курсу «Комп'ютерні технології дизайну».

Для досягнення мети проекту потрібно розв'язувати окремі задачі роботи:

- ознайомитися з інтернет-джерелами на тему дослідження;
- розглянути наявні поширені технології розробки веб-тренажерів та обрати найбільш відповідну для реалізації проекту;
- ознайомитися зі сценаріями проходження тренажеру, тобто з роботою в 3DS Max, яку необхідно візуалізувати у тренажері;
- підготувати ресурси (зображення інтерфейсу, розмічені області, інфраструктура проекту) та обладнання (вебсервер, середовище розробки та інше необхідне програмне забезпечення) для роботи;
- створити інтерфейс користувача (головне вікно, вікно кроку тренажеру та інші);
- написати скрипти проходження тренажеру (скрипт натиснення на інтерактивну область, скрипт переходу між кроками та інші);
- тестування готового функціоналу.

Готовий тренажер повинен відповідати таким функціональним вимогам:

- мати зручний та зрозумілий інтерфейс;
- реалізувати локалізацію українською, російською та англійською мовами, з можливістю вибору мови перед початком проходження;
- можливість вибору варіанту перед початком проходження;
- користувач повинен мати змогу ознайомитися з інструкцією перед початком проходження;
- реалізувати три спроби на проходження тренажеру, після вибору невірної варіанту одна спроба втрачається;
- показ вікна результатів після завершення проходження.

Як інструмент по розмітці та стилізації інтерактивних областей було вирішено обрати вбудований в HTML засіб. Вибір зумовлений тим, що

використання вбудованих засобів є більш надійним, легким та безпечним рішенням, ніж використання сторонніх бібліотек. Також перелічені в розділі 1.1 бібліотеки є застарілими у порівнянні з вбудованим засобом, який надає зручні та гнучкі рішення по роботі з векторною графікою.

Серед засобів по розмітці інтерактивних областей був обраний десктопний векторний графічний редактор Inkscape. Вибір обумовлений тим, що він є простим, зручним та легким, на відміну від аналогів. Також Inkscape є безкоштовним.

Розробка буде вестися за допомогою редактору вихідного коду Visual Studio Code, який є багатоплатформним, легким та потужним інструментом. Для контролю якості та стилю коду в редактор також будуть встановлені плагіни: SonarQube – для моніторингу та виправлення можливих похибок у коді; Prettier – для якісного форматування коду та виправлення можливих стилістичних помилок.

В якості веб-серверу буде використаний Apache 2.4.

## 2 ПРОЕКТУВАННЯ КОМПЛЕКСУ ТРЕНАЖЕРІВ

### 2.1 Структурно-функціональне моделювання

Проектування інформаційної системи – це досить трудомістка, складна та тривала робота, виконання і правильне застосування якої дозволяє підвищити якість інформаційної системи і збільшити її термін життя. Для успішної реалізації проекту об'єкт проектування повинен бути адекватно описаний, тобто повинні бути побудовані якісні моделі архітектури програмного забезпечення. Саме моделі є засобом візуалізації і документування архітектури системи [13].

Діаграма IDEF0, розроблена для прикладу функціонування одного інтерактивного тренажера, представлена на рисунку 2.1. Інший тренажер функціонує аналогічно. Діаграма використовується для опису бізнес-процесів [14], яка містить вхідні процеси (мова інтерфейсу, варіант), вихідні процеси (кількість балів), керівні процеси (алгоритм роботи тренажера, методичні вказівки) та механізми (студент, технічне та програмне забезпечення).

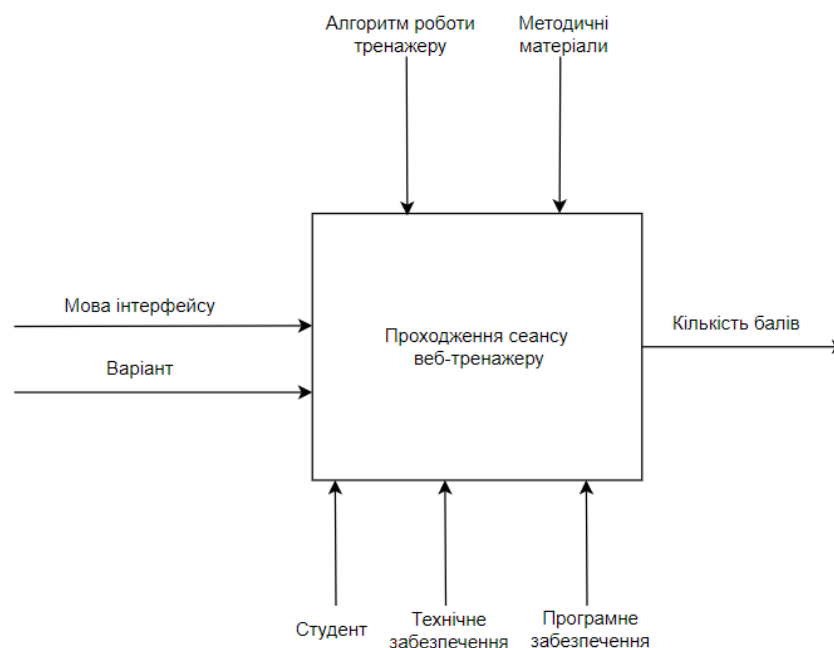


Рисунок 2.1 – Діаграма IDEF0

Процес «Проходження сеансу тренажеру» декомпонується на 4 процеси: «Перегляд інструкції користувача», «Перегляд теорії», «Проходження основного етапу тренажеру», «Отримання результатів».

Діаграму декомпозиції процесу «Проходження сеансу тренажеру» представлено на рисунку 2.2.

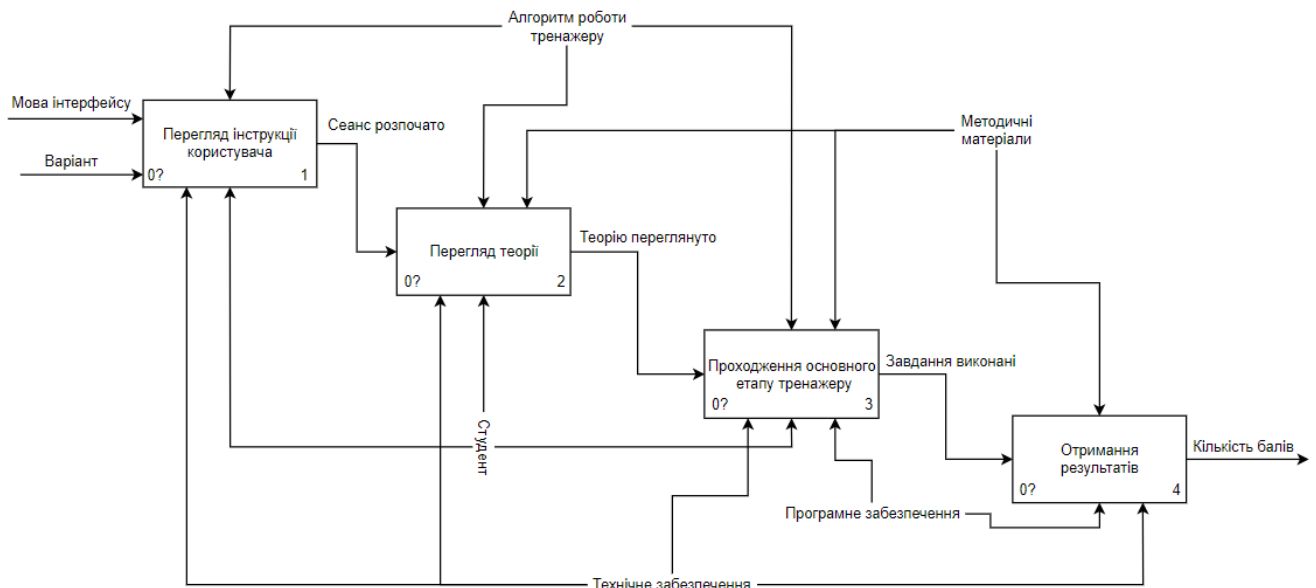


Рисунок 2.2 – Діаграма декомпозиції

Діаграма сценаріїв використання будується під час ознайомлення із технічним завданням та описує розроблювану систему на концептуальному рівні. Діаграма показує відносини між акторами та прецедентами (варіантами використання) та описує дії, які виконує користувач при роботі з системою [15]. Діаграма сценаріїв використання представлена на рисунку 2.3. Акторами системи є користувач, Apache, бібліотека `trainer.engine.js`, бібліотека `mustache.js`, файли конфігурації. Прецедентами є перегляд теорії, перегляд інструкції, вибір мови, вибір варіанту, проходження сценарію, підрахунок балів.

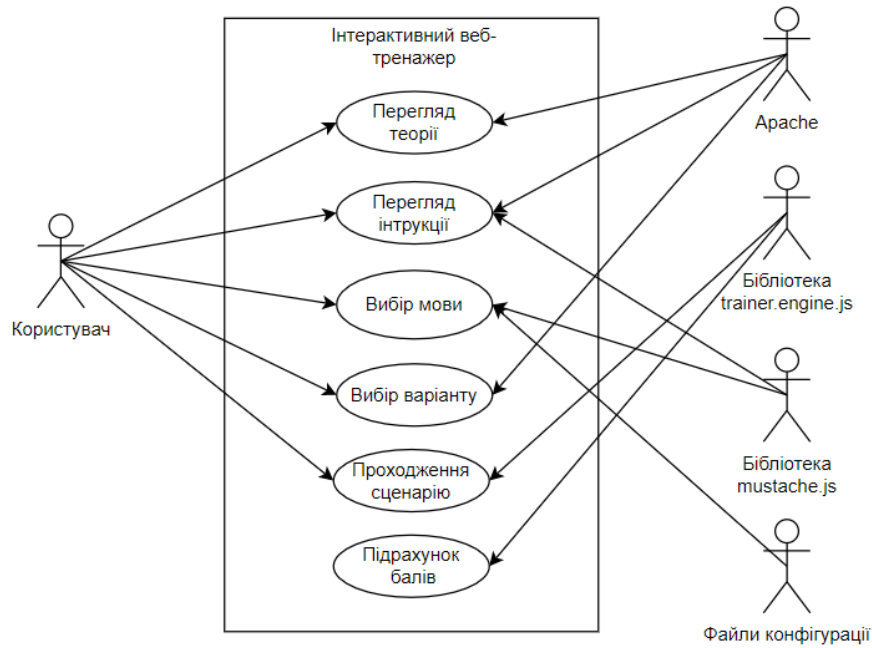


Рисунок 2.3 – Діаграма сценаріїв використання

Діаграма станів представлена на рисунку 2.4. Діаграма демонструє перехід об'єктів між станами. Чорне коло показує початковий стан об'єкту, а коло з чорним колом в ньому показує кінцевий стан об'єкту. Прямокутники описують стан, в якому знаходиться об'єкт, а стрілки описують перехід між станами [16].

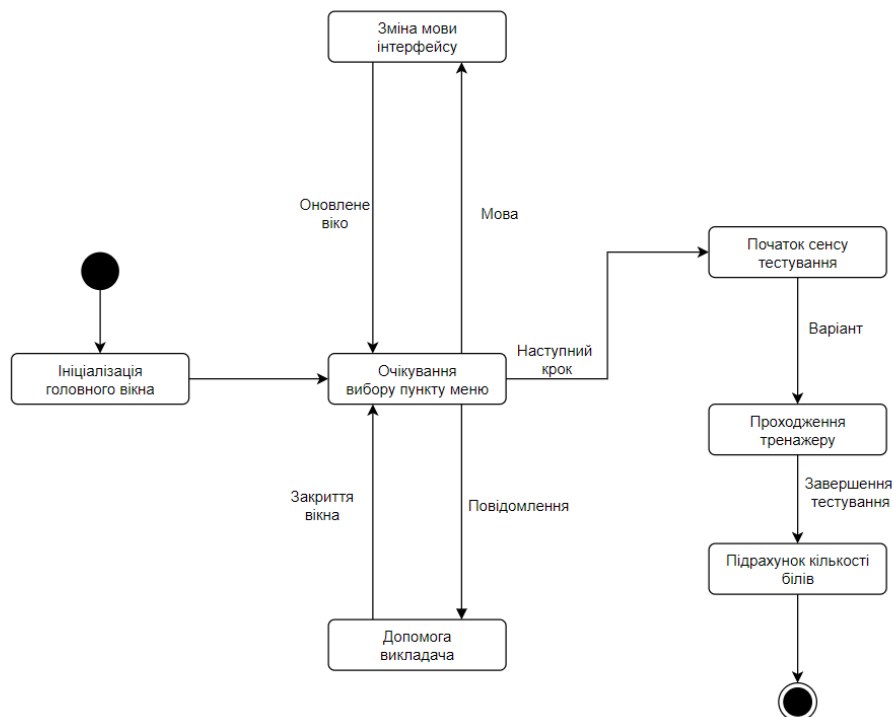


Рисунок 2.4 – Діаграма станів

Діаграма класів аналізу представлена на рисунку 2.5. Діаграма описує відношення між класами. Елементами діаграми є: граничні класи (панель вибору мови, головне вікно, панель навігації та інші), які є абстракціями панелей, діалогових вікон та форм; керівні класи (валідатор), які відповідають за взаємодію між об'єктами; класи сутностей (вибір варіанту, кроки виконання, підрахунок балів), які є абстракціями сутностей. Класи поєднуються за допомогою стрілок [17].

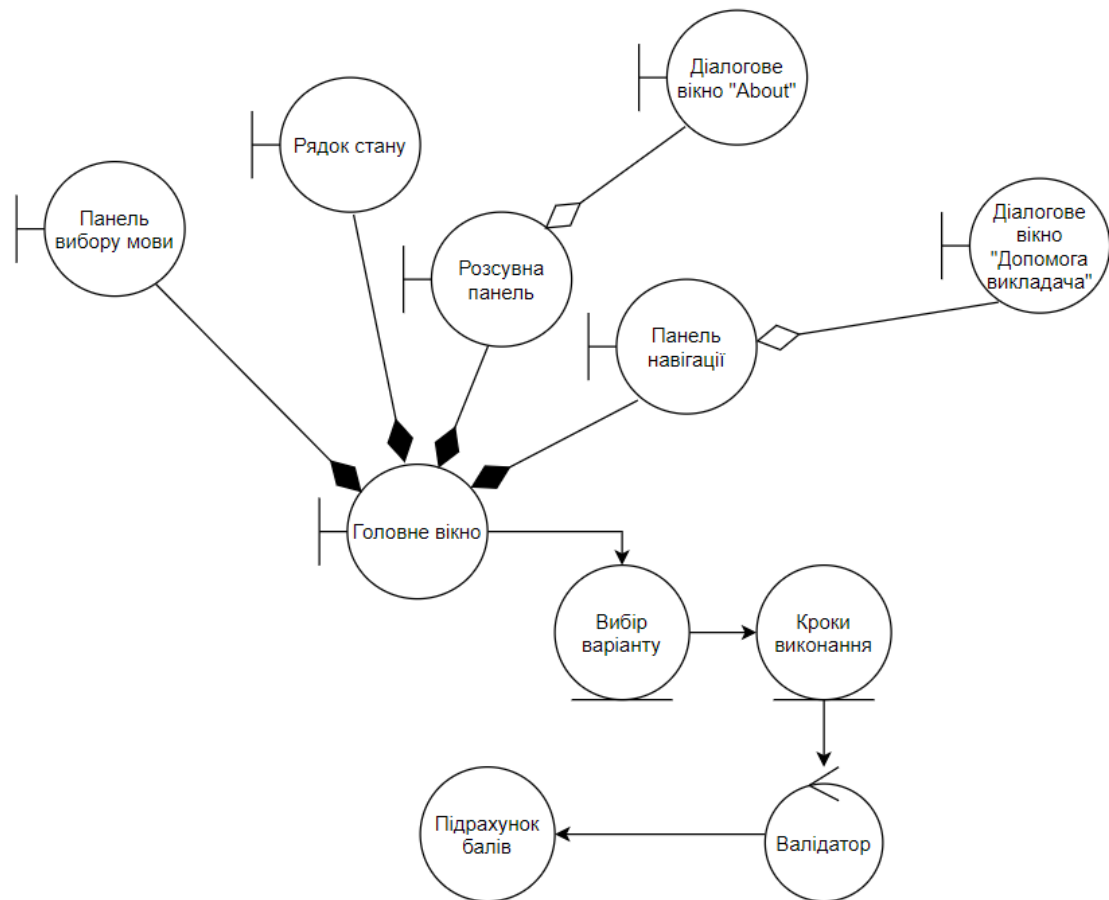


Рисунок 2.5 - Діаграма класів аналізу

Діаграма послідовності представлена на рисунку 2.6. Діаграма демонструє послідовність взаємодії між об'єктами інформаційної системи. Діаграма складається з: прямокутників (користувач, тренажер, валідатор, підрахунок балів) – це об'єкти інформаційної системи; вертикальних прямокутників – «лінії життя» об'єктів; стрілок, які показують взаємодії між об'єктами.



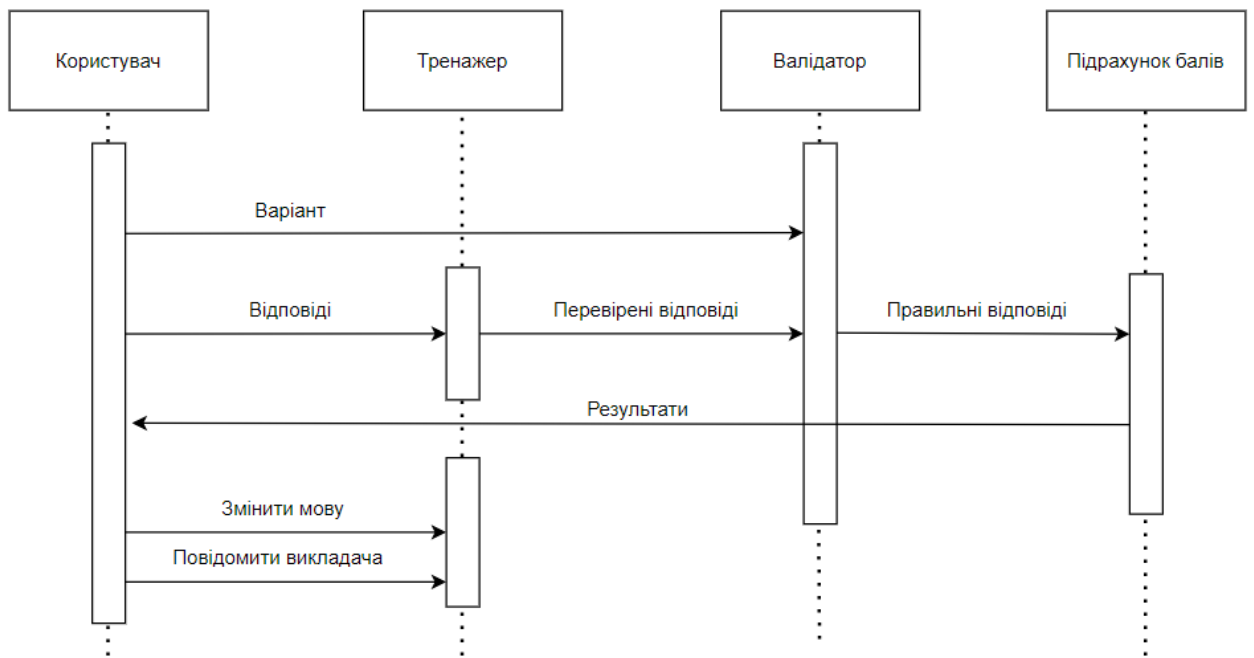


Рисунок 2.6 – Діаграма послідовності

Діаграма комунікації представлена на рисунку 2.7. Діаграма показує взаємодію між акторами та класами.

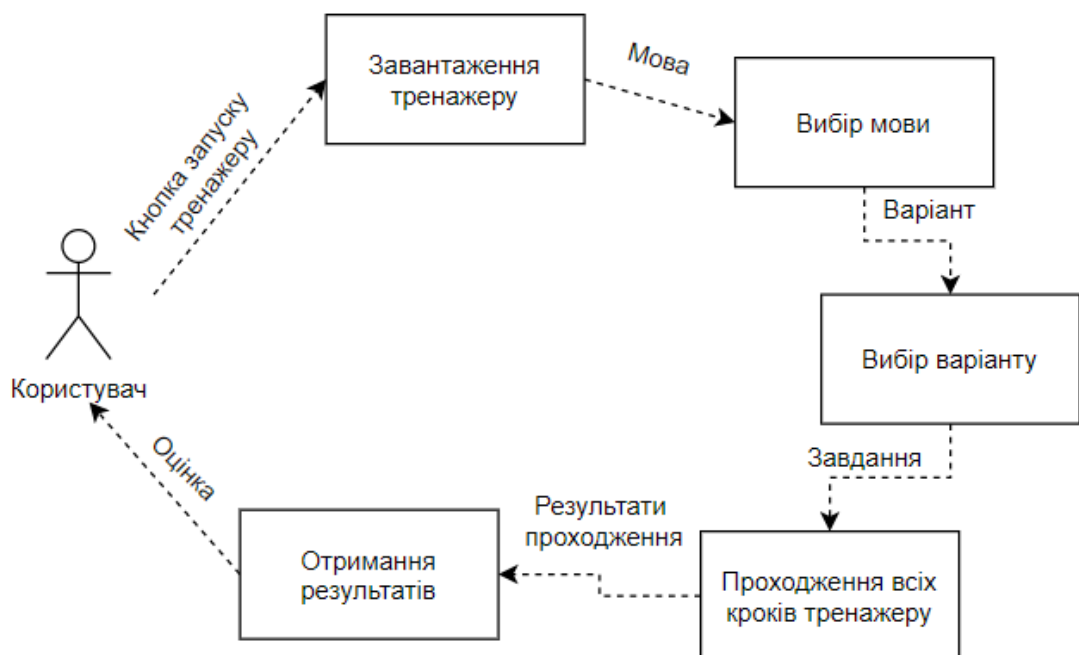


Рисунок 2.7 – Діаграма комунікації

Діаграма діяльності представлена на рисунку 2.8. Діаграма демонструє алгоритм виконання програми.

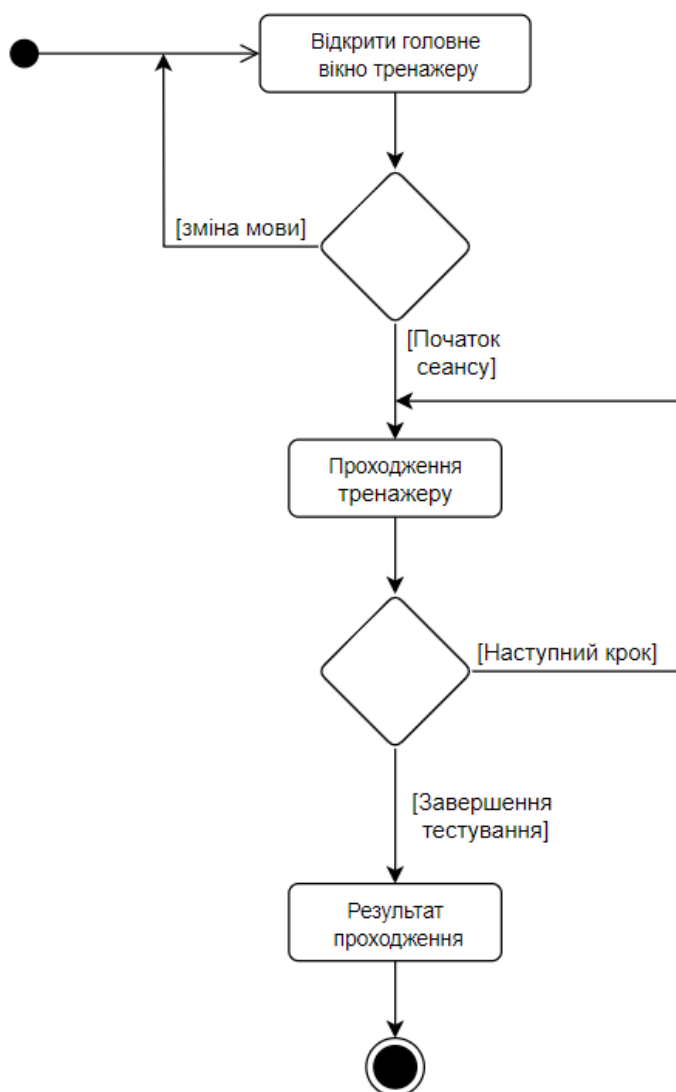


Рисунок 2.8 – Діаграма діяльності

Діаграма компонентів представлена на рисунку 2.9. Діаграма описує особливості фізичних рівнів системи та дозволяє спроектувати архітектуру системи, яка розробляється [18].

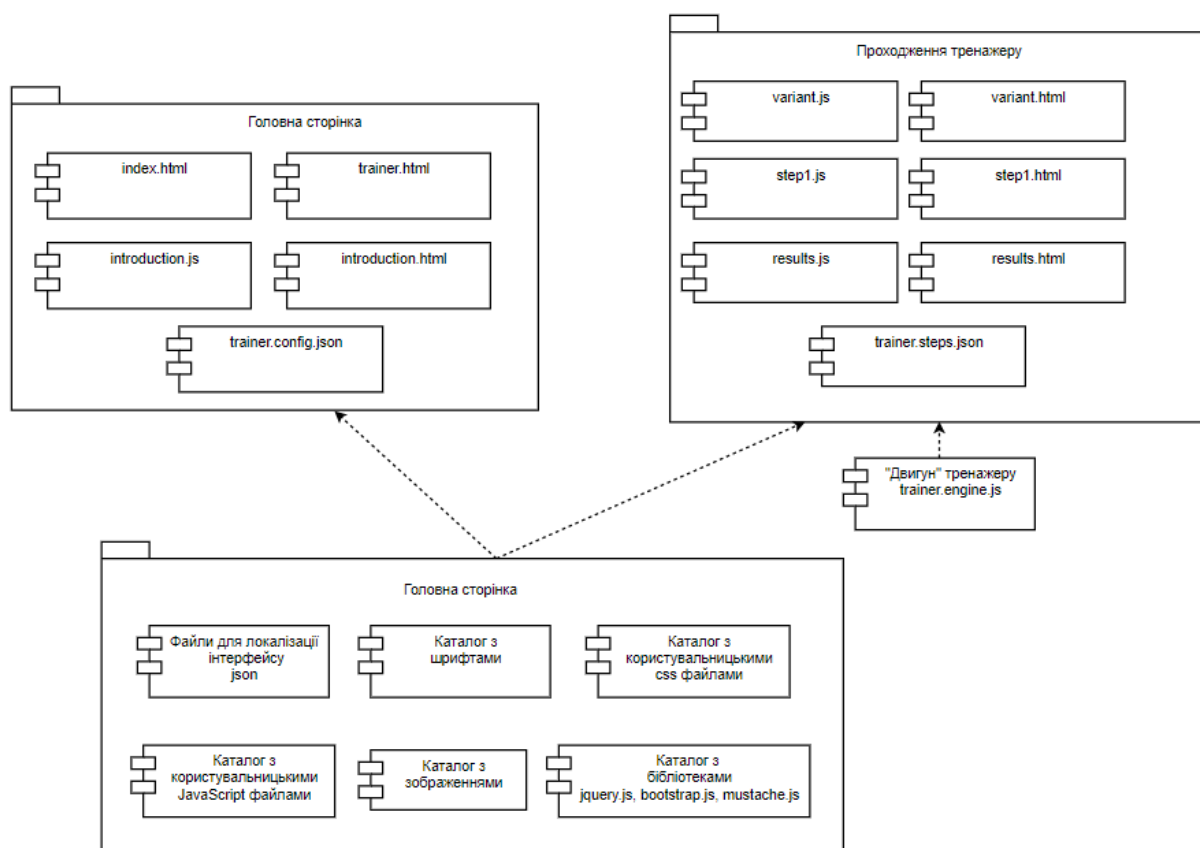


Рисунок 2.9 – Діаграма компонентів

Діаграма розгортання представлена на рисунку 2.10. Діаграма показує функціональні елементи системи, які використовуються при роботі програми.

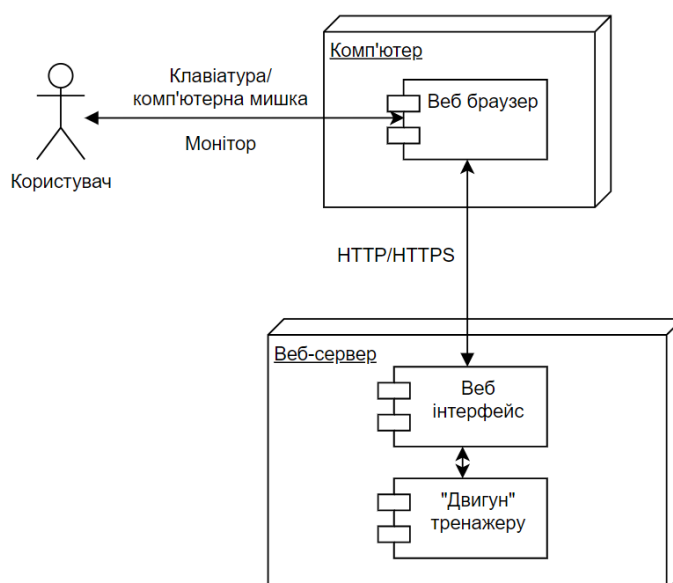


Рисунок 2.10 – Діаграма розгортання

## 2.2 Проектування моделі бази даних

Як база даних в проєкті використовуються JSON файли локалізації, які містять пари ключ/значення. JSON файлами керує система веб шаблонів `mustache.js`.

На рисунку 2.11 показана частина структури JSON файлу локалізації. Ключ `local` містить назву мови, на якій описано файл. Ключ `lang` містить перелік вкладених пар ключ/значення, де ключем є змінна середовища, а значенням – локалізований рядок. Наприклад ключ `TRAINER_NAME` містить назву тренажера. Ключ поміщається в HTML код сторінки та після її завантаження змінює своє значення згідно активному файлу локалізації.

```
"local": "English",  
"lang": {  
  "TRAINER_NAME": "Primitives",  
  "MAX_SCORE": "Max score",  
  "NEXT_STEP": "Next step",  
  "ENTER_TEXT": "Enter text",  
  "ENTER_VARIANT": "Enter variant",  
  "LENGTH": "Length",  
  "WIDTH": "Width",  
  "HEIGHT": "Height",
```

Рисунок 2.11 – Частина змісту файлу локалізації

На рисунку 2.12 показано зміст JSON файлу конфігурації тренажера. Ключ `LANGUAGES` містить список файлів локалізації, які використовуються для поточного тренажера. Ключ `DEFAULT_LANG` містить мову інтерфейсу за замовчуванням. Ключ `TRAINER_SCORE` містить максимальну кількість балів, яка нараховується за проходження усіх кроків тренажера. Ключі `TRAINER_AUTHOR` та `COURSE_AUTHOR` містять інформацію про авторів. Ключ `PRODUCTION` містить значення стану розробки.

```

{
  "LANGUAGES": {
    "trainer2_ua": "Українська",
    "trainer2_ru": "Русский",
    "trainer2_en": "English"
  },
  "DEFAULT_LANG": "trainer2_ua",
  "TRAINER_SCORE": 100,
  "TRAINER_AUTHOR": "Vadim Isaev",
  "COURSE_AUTHOR": "Vadim Isaev",
  "PRODUCTION": false
}

```

Рисунок 2.12 – Зміст файлу конфігурації тренажеру

На рисунку 2.13 показаний зміст JSON файлу конфігурації кроків тренажеру. Файл містить певну кількість блоків, кожен з яких описує один крок. Ключ filename містить назву HTML та JavaScript файлів кроку. Ключ score містить кількість балів, яка нараховується за проходження кроку. Ключ hasScript містить значення, яке показує чи містить крок JavaScript файл.

```

[
  {
    "filename" : "introduction",
    "score" : 0,
    "hasScript" : true
  },
  {
    "filename" : "instruction",
    "score" : 0,
    "hasScript" : true
  },
  {
    "filename" : "trainer2_step1",
    "score" : 25,
    "hasScript" : true
  },
  {
    "filename" : "trainer2_step2",
    "score" : 25,
    "hasScript" : true
  },
  {
    "filename" : "trainer2_step3",
    "score" : 25,
    "hasScript" : true
  },
  {
    "filename" : "trainer2_step4",
    "score" : 25,
    "hasScript" : true
  },
  {
    "filename" : "results",
    "score" : 0,
    "hasScript" : true
  }
]

```

Рисунок 2.13 – Зміст файлу trainer.steps.json

## 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ КОМПЛЕКСУ ТРЕНАЖЕРІВ

### 3.1 Архітектура програмного додатку

Комплекс містить два незалежні інтерактивні тренажери, об'єднані темою вивчення програми 3ds Max. Перший тренажер сприяє вивченню слухачами теми «Стандартні примітиви в 3ds Max», другий – теми «Дії з об'єктами». Побудовані і розроблені тренажери за однаковим принципом. Кожний тренажер містить по чотири кроки. Користувач обирає перед початком роботи один із 10 варіантів, кожний варіант відрізняється параметрами конкретного завдання.

Архітектура кожного тренажеру (рис. 3.1) складається із трьох рівнів: клієнтська частина, серверна частина та сервер бази даних. Користувачами системи є викладач та студент. На першому рівні системи відбувається взаємодія студента з інтерфейсом тренажеру. Студент може звернутися за допомогою до викладача та після звернення викладач повинен побачити повідомлення студента у своєму інтерфейсі. Після проходження тренажеру студентом кількість отриманих балів відправляються на сервер. На другому рівні сервер отримує від клієнта кількість балів та записує їх до бази даних. На третьому рівні база даних отримує від сервера запит на запис балів.

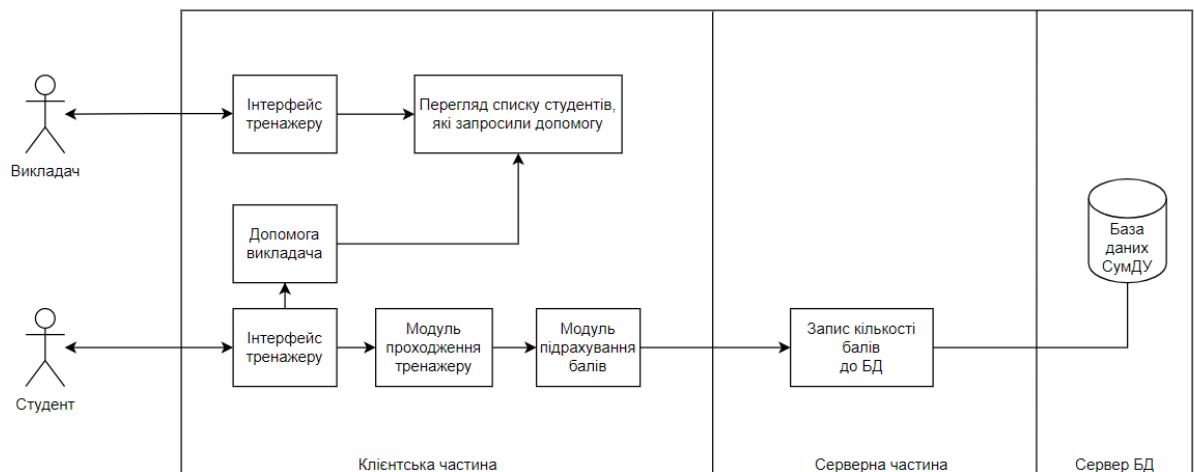


Рисунок 3.1 – Архітектура програмного додатку

## 3.2 Структура проекту

Кожний інтерактивний веб-тренажер було створено на основі JavaScript бібліотеки по створенню тренажерів. За допомогою цієї бібліотеки можна створювати прості багатокрокові веб-тренажери з використанням таких елементів як: кнопки, поля для вводу, графіки, списки, елементи перетаскування, чек та радіо бокси.

Першим кроком по розробці власного проекту є ознайомлення із бібліотекою для створення веб-тренажерів.

Другим кроком є створення інфраструктури проекту. Структура кореню проекту показана на рисунку 3.2.

В корені каталогу знаходяться 2 файли: `index.html` та `trainer.html`. Перший файл є стартовою точкою виконання проекту і містить HTML фрейм, зміст якого завантажується з файлу `trainer.html`. Зміст файлу `index.html` представлено на рисунку 3.3.

Локальний диск (C:) > Server > data

Имя	Дата изменения	Тип	Размер
^			
.vscode	14.04.2021 13:12	Папка с файлами	
img	16.04.2021 19:33	Папка с файлами	
javascripts	16.04.2021 19:33	Папка с файлами	
langs	16.04.2021 19:33	Папка с файлами	
trainer	16.04.2021 19:33	Папка с файлами	
index	03.04.2021 20:07	Chrome HTML Do...	1 КБ
trainer	03.04.2021 20:07	Chrome HTML Do...	6 КБ

Рисунок 3.2 – Структура проекту

```

<html lang="en">
  <head>
    <meta charset="UTF-8"/>
  </head>
  <body style="margin: 0; padding: 0; overflow: hidden">
    <iframe src="trainer.html" style="position: absolute; height: 100% height="100%" width="100%"></iframe>
  </body>
</html>

```

Рисунок 3.3 – Зміст файлу index.html

Папка .vscode містить файл налаштувань середовища розробки Visual Studio Code.

Папка img містить зображення, які використовуються для імітації інтерфейсу 3DS Max.

Папка javascripts (рис. 3.4) містить файли третіх сторін, такі як бібліотеки, шрифти, стилі та зображення. Папка css містить файли стилізації (animate.css, bootstrap.min.css, trainer.style.css). Папка fonts містить файли шрифтів, які використовуються в проєкті. Папка img містить зображення, які використовуються при формуванні інтерфейсу тренажеру. Папка js містить JavaScript файли (bootstrap.min.js, jquery.min.js, mustache.js, trainer.engine.js, additions\drag-drops.js, additions\html2canvas.js).

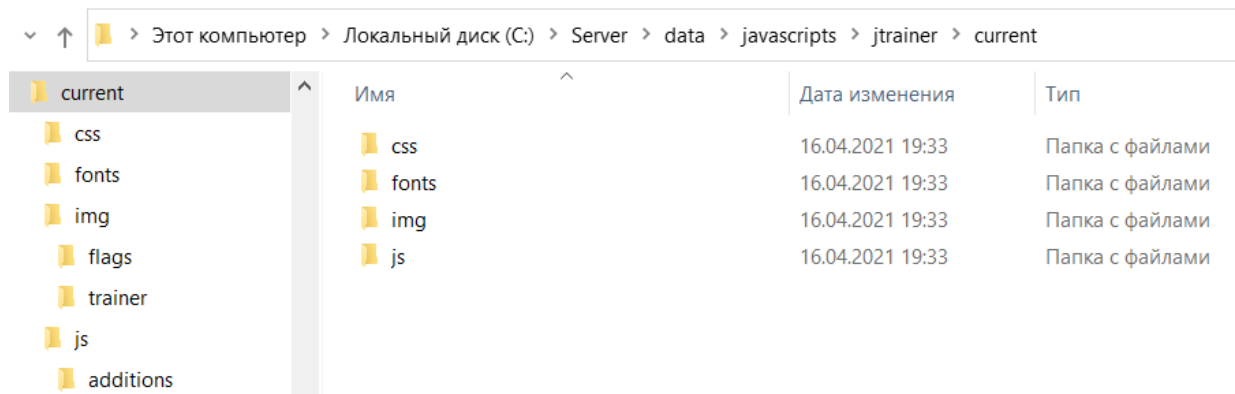


Рисунок 3.4 – Структура папки javascripts

Папка langs (рис. 3.5) містить JSON файли локалізації, які дозволяють використовувати тренажери трьома мовами – українською, англійською та російською.



Локальный диск (C:) &gt; Server &gt; data &gt; langs

Имя	Дата изменения	Тип	Размер
trainer1_en	13.03.2021 15:27	JSON File	5 КБ
trainer1_ru	13.03.2021 15:27	JSON File	7 КБ
trainer1_ua	13.03.2021 15:27	JSON File	7 КБ
trainer2_en	03.04.2021 12:39	JSON File	5 КБ
trainer2_ru	03.04.2021 12:39	JSON File	7 КБ
trainer2_ua	03.04.2021 12:39	JSON File	7 КБ

Рисунок 3.5 – Зміст папки langs

Папка trainer (рис. 3.6) містить файли, які використовуються для створення та управління кроками тренажеру. В корені папки містяться HTML файли кожного кроку тренажеру.

Папка css містить стилі, які застосовуються до HTML файлів кроків. Папка scripts містить скрипти управління кожним кроком тренажеру. Папка settings містить JSON файли trainer.config.json та trainer.steps.json, які містять рядки стану середовища.

Локальный диск (C:) &gt; Server &gt; data &gt; trainer

Имя	Дата изменения	Тип	Размер
css	16.04.2021 19:33	Папка с файлами	
scripts	19.04.2021 21:46	Папка с файлами	
settings	16.04.2021 19:33	Папка с файлами	
instruction	03.04.2021 20:07	Chrome HTML Do...	1 КБ
introduction	03.04.2021 20:07	Chrome HTML Do...	2 КБ
results	06.03.2021 19:32	Chrome HTML Do...	1 КБ
trainer1_step1	03.04.2021 20:07	Chrome HTML Do...	16 КБ
trainer1_step2	14.03.2021 11:43	Chrome HTML Do...	7 КБ
trainer1_step3	13.03.2021 15:27	Chrome HTML Do...	10 КБ
trainer1_step4	14.03.2021 11:43	Chrome HTML Do...	12 КБ
trainer2_step1	14.03.2021 11:43	Chrome HTML Do...	4 КБ
trainer2_step2	27.03.2021 13:26	Chrome HTML Do...	4 КБ
trainer2_step3	27.03.2021 13:26	Chrome HTML Do...	5 КБ
trainer2_step4	27.03.2021 11:57	Chrome HTML Do...	11 КБ

Рисунок 3.6 – Структура папки trainer

### 3.3 Програмна реалізація

Третім кроком у розробці проекту є підготовка зображень інтерфейсу 3DS Max та розмітка їх інтерактивних областей. Після створення знімків екрану інтерфейсу 3DSMax необхідно за допомогою векторного редактору Inkscape розмітити інтерактивні області кожного знімку. Приклад розмітки представлений на рисунку 3.7, код інтерактивної області знаходиться у вікні в правій частині рисунку.

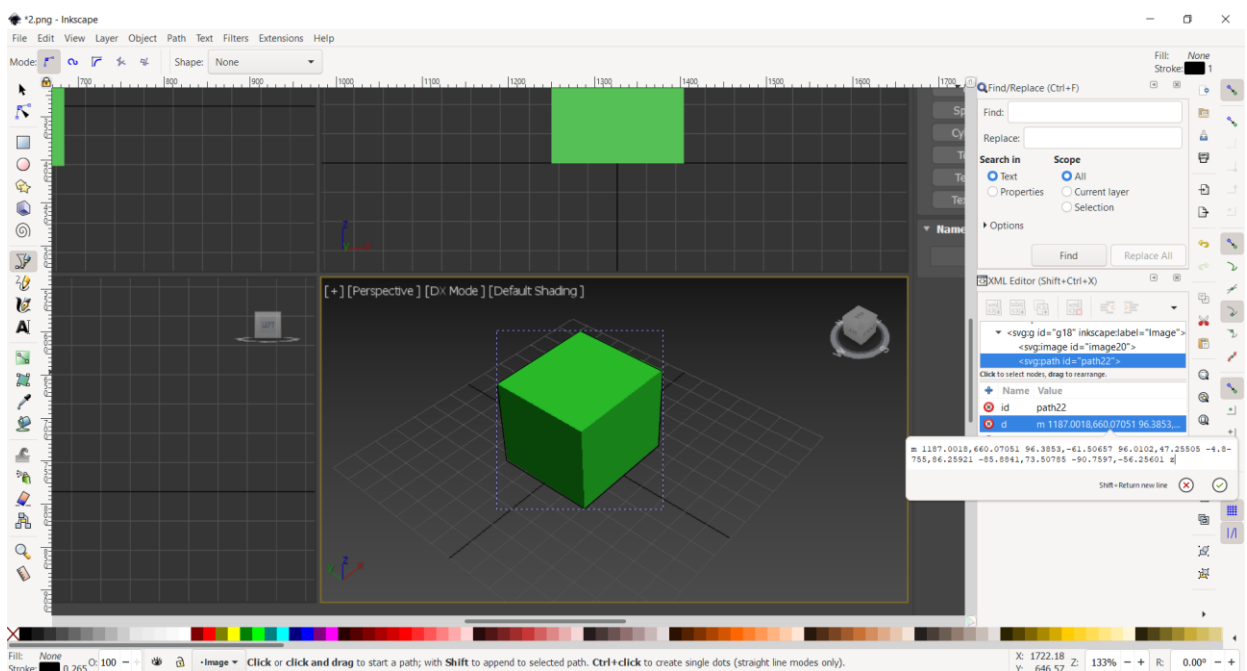


Рисунок 3.7 – Створення інтерактивної області навколо границь куба

Тепер отриманий код переноситься до HTML файлу кроку. На рисунку 3.8 представлений код області, який поміщений в атрибут d.

```
<a href="javascript:void(0)">
  <path id="10" class="part" d="m 1187.0018,660.07051 96.3853,-61.50657 96.0102,47.25505 -4.8755,86.25921 -88.8841,73.50785 -90.7597,-56.25601 z" />
</a>
```

Рисунок 3.8 – Приклад однієї розміченої області рисунку

Щоб помістити області для вводу на рисунок інтерфейсу, необхідно вручну підбирати значення атрибутів x та y для того, щоб створене поле для вводу стало

на місце поля для вводу інтерфейсу 3DS Max. На рисунку 3.9 представлений код елемента розміченого поля для вводу.

```
<foreignObject class="step2-inputs" x="953" y="976" width="82" height="160">
  {{{STEP2_INPUT_X}}}
</foreignObject>
```

Рисунок 3.9 – Приклад розміченої області для вводу

Наступним кроком є написання скриптів взаємодії користувача з тренажером та створення інтерфейсу користувача. Кожен крок тренажеру має свій JavaScript та HTML файл, де описується інтерфейс та логіка роботи кроку. Лістинг з кодом файлів наведений в додатку.

На рисунку 3.10 представлений код HTML файлу одного із кроків. Кожен HTML файл кроку містить текст завдання та може містити таблицю зі значеннями, які користувач буде повинен ввести у поля для вводу.

```
1 <link rel="stylesheet" href="/trainer/css/steps.css" type="text/css"/>
2 <div id="step2" class="jumbotron page1">
3   <h3>{{{STEP2_TITLE}}}</h3>
4   <p>{{{STEP2_TASK}}}</p>
5   <table class="table table-bordered text-center">
6     <thead><tr><th scope="col" class="text-center">X</th><th scope="col" class="text-center">Y</th><th scope="col" class="text-center">Z</th></tr></thead>
7     <tbody>
8       <tr><td id="x_value"></td><td id="y_value"></td><td id="z_value"></td></tr>
9     </tbody>
10  </table>
11  <div class="map">
12    <svg id="svgmap" viewBox="0 0 1920 1030">
13      <a href="javascript:void(0)">
14        <path id="1" class="part" d="m 310.90821,63.38177 -0.18752,31.315846 2.8128,2.62528 33.56609,0.18752 2.06272,-2.43776 0.56256,-31.128326 -2.62528,-3.18784 -33.37857,0.18752 z">
15      </a>
16      <a href="javascript:void(0)">
17        <path id="2" class="part" d="m 349.72486,63.94433 -0.18752,31.315845 2.8128,2.62528 33.56609,0.18752 2.06272,-2.43776 0.56256,-31.128326 -2.62528,-3.18784 -33.37857,0.18752 z">
18      </a>
19      <a href="javascript:void(0)">
20        <path id="3" class="part" d="m 391.54183,63.75681 -0.18752,31.315845 2.8128,2.62528 33.56609,0.18752 2.06272,-2.43776 0.56256,-31.128326 -2.62528,-3.18784 -33.37857,0.18752 z">
21      </a>
22      <a href="javascript:void(0)">
23        <path id="4" class="part" d="m 430.35848,64.31937 -0.18752,31.315844 2.8128,2.62528 33.56609,0.18752 2.06272,-2.43776 0.56256,-31.128324 -2.62528,-3.18784 -33.37857,0.18752 z">
24      </a>
25      <a href="javascript:void(0)">
26        <path id="5" class="part" d="m 476.86246,64.694426 -0.18752,31.315844 2.8128,2.62528 33.56609,0.18752 2.06272,-2.43776 0.56256,-31.128324 -2.62528,-3.18784 -33.37857,0.18752 z">
27      </a>
28      <a href="javascript:void(0)">
29        <path id="6" class="part" d="m 517.55431,64.131866 -0.18752,31.315843 2.8128,2.62528 33.56609,0.18752 2.06272,-2.43776 0.56256,-31.128323 -2.62528,-3.18784 -33.37857,0.18752 z">
30      </a>
31      <a href="javascript:void(0)">
32        <path id="7" class="part" d="m 558.80871,64.131867 -0.18752,31.31584 2.8128,2.62528 33.5661,0.18752 2.06272,-2.43776 0.56256,-31.12832 -2.62528,-3.18784 -33.37858,0.18752 z">
33      </a>
34      <a href="javascript:void(0)">
35        <path id="8" class="part" d="m 597.62537,64.694427 -0.18752,31.31584 2.8128,2.62528 33.5661,0.18752 2.06272,-2.43776 0.56256,-31.12832 -2.62528,-3.18784 -33.37858,0.18752 z">
36      </a>
37      <a href="javascript:void(0)">
38        <path id="10" class="part" d="m 1187.5359,660.33149 96,-62.32044 96,46.67403 -5.0387,87.51382 -86.1878,72.66298 -90.4309,-57.01657 z">
39      </a>
40    </svg>
41    <foreignObject class="step2-inputs" x="953" y="976" width="82" height="160">
42      {{{STEP2_INPUT_X}}}
43    </foreignObject>
44    <foreignObject class="step2-inputs" x="1055" y="976" width="82" height="160">
45      {{{STEP2_INPUT_Y}}}
46    </foreignObject>
47    <foreignObject class="step2-inputs" x="1156" y="976" width="82" height="160">
48      {{{STEP2_INPUT_Z}}}
49    </foreignObject>
50  </div>
51  
52  <div id="chart" style="height: 20px;">{{{DIAG}}}</div>
53  <button class="btn btn-success pull-right check">{{{CHECK}}}</button>
54 </div>
```

Рисунок 3.10 – Приклад HTML файлу другого кроку другого тренажеру

Текст у кроках представлений у вигляді конструкції `{{ TEXT }}`.

Після завантаження файлу у браузері бібліотека `mustache.js` замінює конструкції з фігурними скобками згідно з розташованою у ній змінній на текст із JSON файлу локалізації, який відповідає поточній мові інтерфейсу користувача.

Кожен крок містить зображення інтерфейсу 3DS Max із розміченими інтерактивними областями, яке поміщається в HTML тег `svg`. Тег `svg` зберігає координати розмічених областей (тег `path`, параметр `d`) та координати областей для вводу (тег `foreignObject`).

За допомогою бібліотеки для створення тренажерів та бібліотеки `mustache.js` в тіло тегів `foreignObject` вбудовується код областей для вводу у форматі `{{{ INPUT_NAME }}}`.

На рисунку 3.11 представлений код JavaScript файлу одного із кроків. Скрипт налаштовує поведінку кроку згідно з варіантом користувача та встановлює події натиснення на інтерактивні області та в деяких кроках на кнопку «Перевірити».

Якщо в кроці передбачені поля для вводу, то їх генерацією займається скрипт.

```

trainer2_step1.js x
1   var VStep1;
2   const imgPath = "img/trainer2-step1/";
3
4   var trainer2_step1 = function () {
5     this.postDispatch = function () {
6       $("#endTrainer").removeClass("disabled btn-default");
7       $("button.check").css("display", "none");
8       var options = [4, 3, 5, 9, 10, 2, 8, 6, 1, 7];
9       var order = [options[userVariant - 1]];
10      var imgInfo = {
11        path: [imgPath + "2-" + userVariant.toString() + ".png"],
12        order: [options[userVariant - 1]],
13      };
14      VStep1 = new Validator();
15      VStep1.addSvgStep(order, imgInfo, null, () => {
16        $(".div.validation-alert-success").fadeIn();
17        Scorer.addScore(25);
18        Rotator.enableNextButton();
19      });
20      VStep1.setStrictMode(true).enableStepFinishAlert(true).setIgnoreCase(false);
21    };
22
23    this.mustache = function () {
24      return {
25        FIGURE_NAME: I18N.getConstants()[getFigureName()],
26      };
27    };
28
29    function getFigureName() {
30      switch (userVariant) {
31        case "1": return "BOX";
32        case "2": return "CONE";
33        case "3": return "SPHERE";
34        case "4": return "CYLINDER";
35        case "5": return "TUBE";
36        case "6": return "TORUS";
37        case "7": return "PYRAMID";
38        case "8": return "TEAPOT";
39        case "9": return "PLANE";
40        case "10": return "TEXT_PLUS";
41      }
42    }
43  };

```

Рисунок 3.11 – Приклад JavaScript файлу першого кроку другого тренажеру

Також був розширений функціонал бібліотеки по створенню тренажерів, зокрема були додані методи для включення кроку, який містить роботу із svg зображеннями до загальної системи керування кроками бібліотеки.

Наприклад було додано метод `addSvgStep` (рис. 3.12), який створює логіку натиснення на інтерактивну область.

```

this.addSvgStep = function (order, imgInfo, partsBorders, callback) {
  this.addSvgScore(order);
  $(".part").click(function () {
    if (order[0] == $(this).attr("id")) {
      $(this).css("fill", "green");
      if (order[0] == imgInfo.order[0]) {
        $("#mainImg").attr("src", imgInfo.path[0]);
        imgInfo.order.shift();
        imgInfo.path.shift();
      }
      if (partsBorders) {
        $(".part").css("fill", "blue");
        $(".part:gt(0)").hide();
        $(".part:lt(" + partsBorders.max[0] + ")").show();
        $(".part:lt(" + partsBorders.min[0] + ")").hide();
        partsBorders.min.shift();
        partsBorders.max.shift();
      }
      order.shift();
      if (order.length == 0) {
        $(".part").hide();
        callback();
      }
    } else {
      $(this).css("fill", "red");
      attempts--;
      if (attempts == 0) {
        $(".part").hide();
        $(".div.validation-alert-danger").fadeIn();
        Rotator.enableNextButton();
      }
    }
  });
}
}

```

Рисунок 3.12 – Код методу addSvgStep для створення кроку із використанням svg зображень

### 3.4 Перевірки працездатності тренажерів

Останнім кроком розробки є тестування готового тренажеру та підготовка проекту для публікації на сервер.

Розроблені тренажери були розміщені в системі дистанційної освіти СумДУ (рис.3.13-3.14).

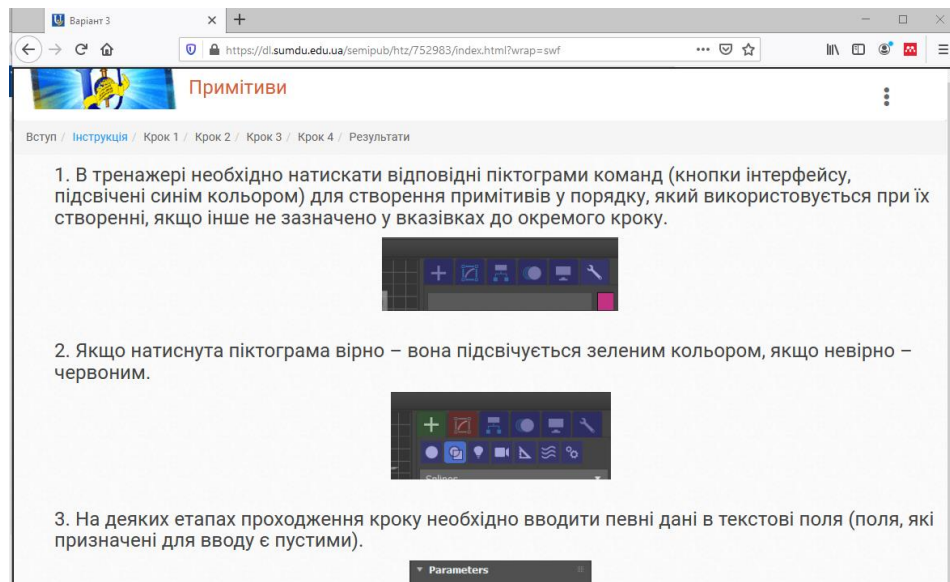


Рисунок 3.13 – Тестування роботи тренажера

Тестування проводилося для обох тренажерів, для різних варіантів та мов локалізації.

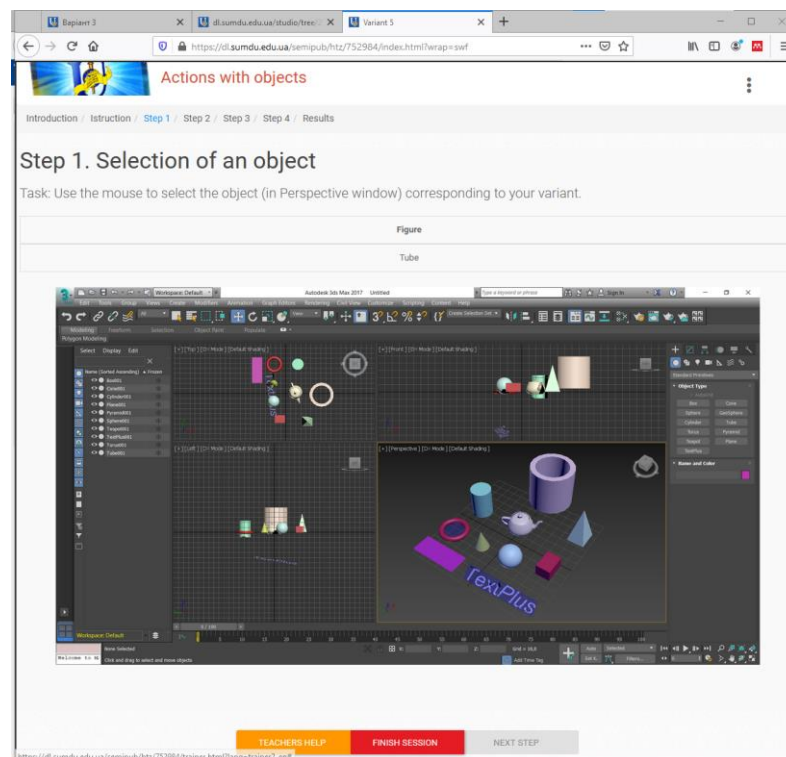


Рисунок 3.14 – Тестування роботи тренажера

Перевірка показала їх повну працездатність. Помилки при тестуванні не виявлено.

### 3.5 Використання тренажерів

Робота з кожним із тренажерів проводиться однаково, та починається після відкриття адреси, за якою знаходиться сайт. Покажемо її на прикладі тренажеру «Дії з об'єктами».

Після завантаження з'являється головна сторінка тренажеру (рис. 3.15). На головній сторінці користувач може змінити мову та обрати варіант для проходження тренажеру.

Інтерфейс сторінки адаптований для системи дистанційної освіти СумДУ та відповідає її вимогам.

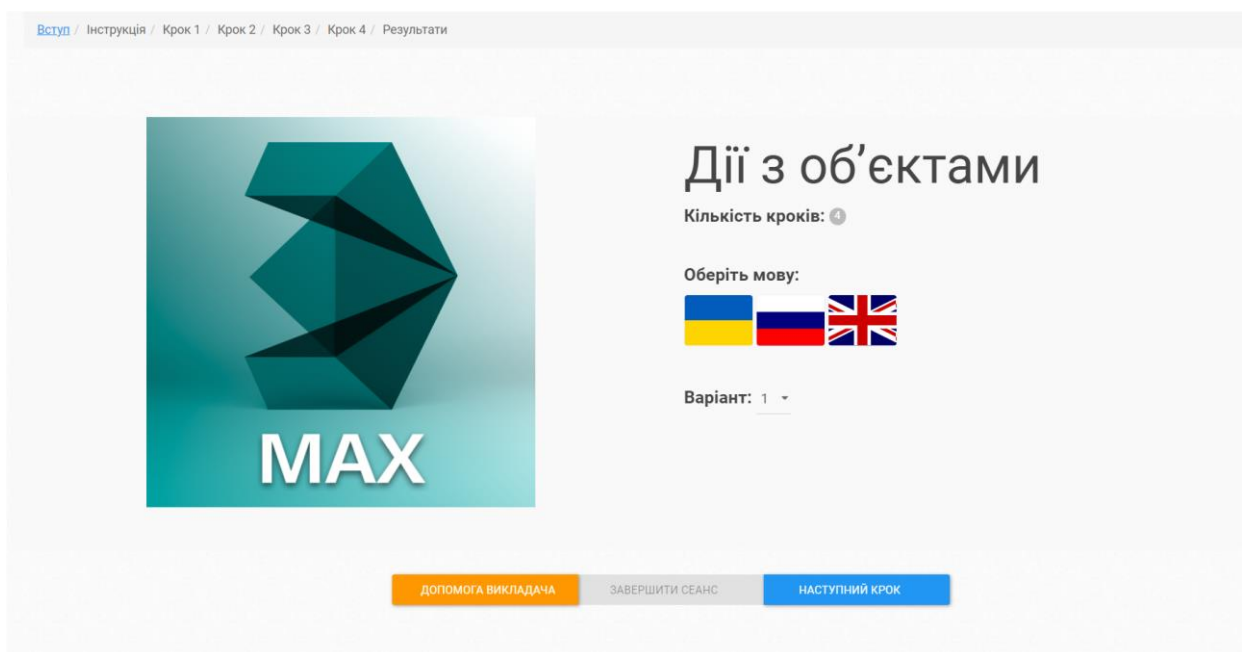


Рисунок 3.15 – Вигляд головного вікна сторінки тренажеру

Протягом всього процесу проходження тренажеру користувач має можливість звернутися за допомогою до викладача, натиснувши кнопку «Допомога викладача».



Після натискання відкриється діалогове вікно (рис. 3.16), де користувач може описати проблему, з якою він зіткнувся.

Після натиснення кнопки «Відправити» текст та знімок екрану поточного стану вікна користувача відправляються до сервера.

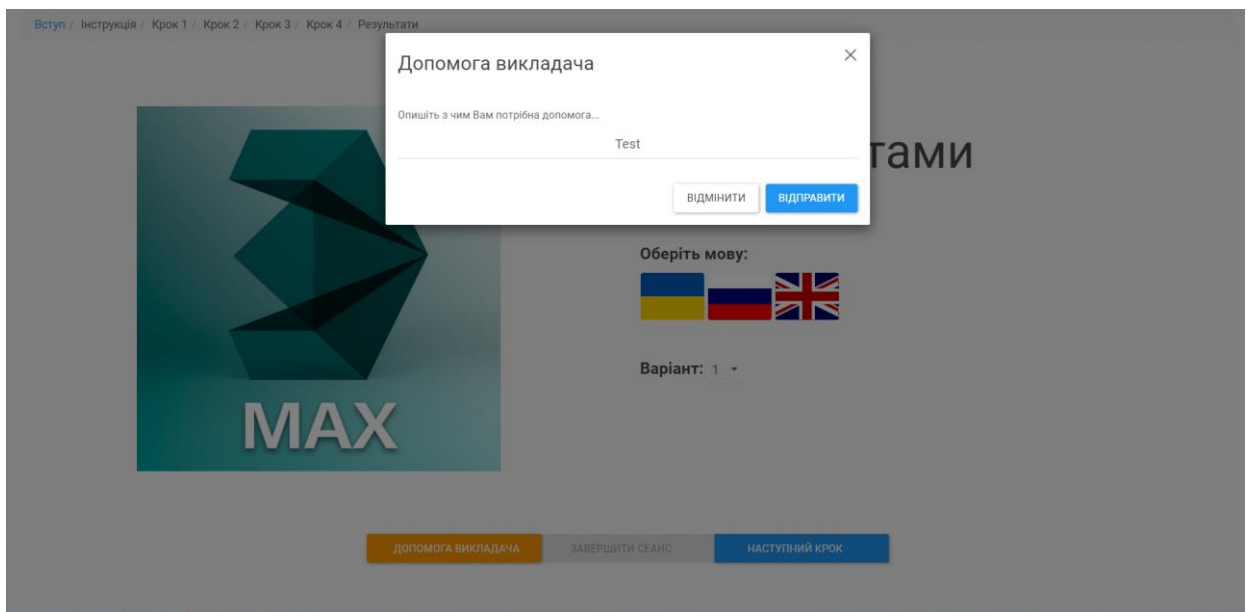


Рисунок 3.16 – Діалогове вікно про з запитом про надання допомоги щодо проходження тренажеру

По завершенню вибору усіх налаштувань на початковому екрані, користувач повинен натиснути кнопку «Наступний крок», після чого зміст зміниться на вікно з інструкцією користувача (рис 3.17).

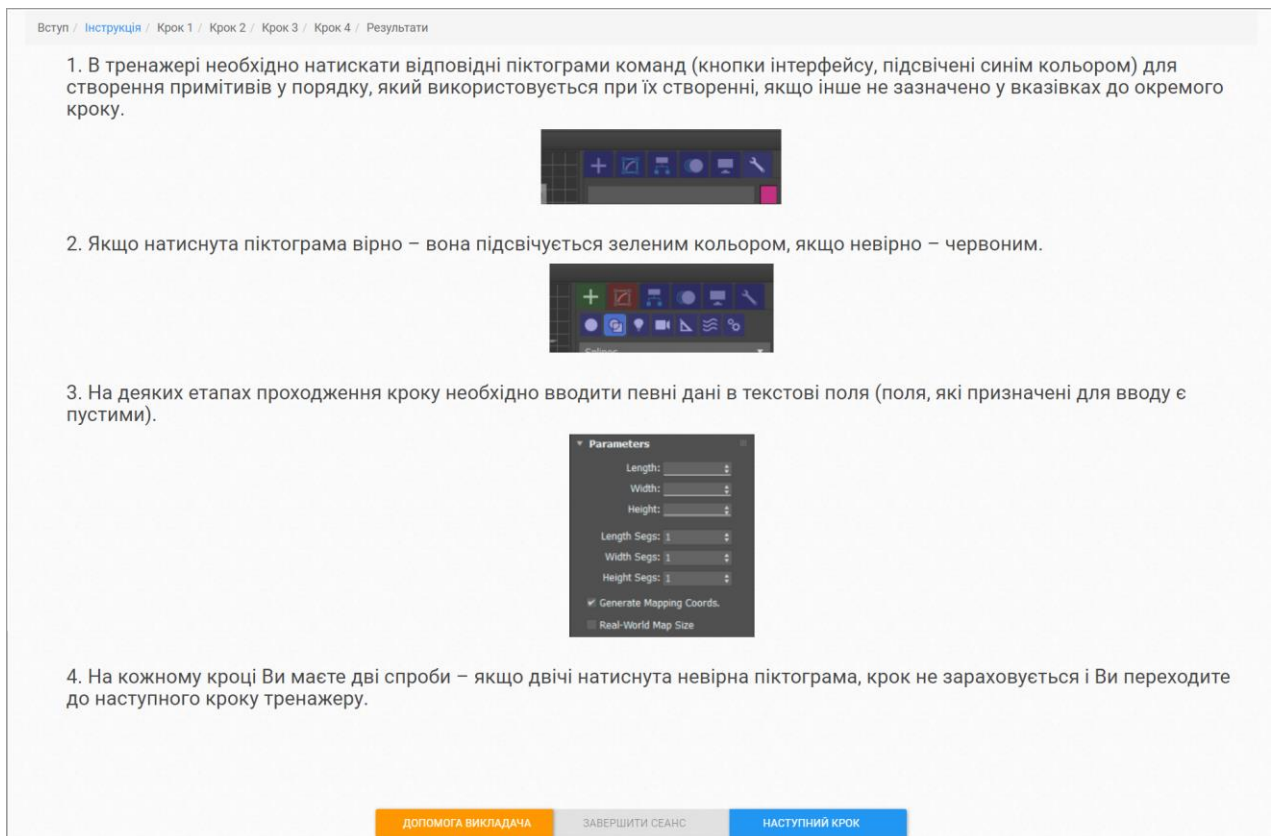


Рисунок 3.17 – Діалогове вікно з інструкцією користувача

Після ознайомлення з інструкцією, починається основний етап проходження тренажеру, який містить чотири кроки. На кожному кроці користувачу необхідно ознайомитися із завданням та виконати його через взаємодію з інтерфейсом 3DS Max.

Перший крок тренажеру показано на рисунку 3.18. Користувачу потрібно обрати на сцені фігуру типу «Вох». Для полегшення вибору користувача елемент сцени, на котрий можна натиснути, підсвічується блакитним кольором та змінює свій відтінок після наведення на нього курсору миші.

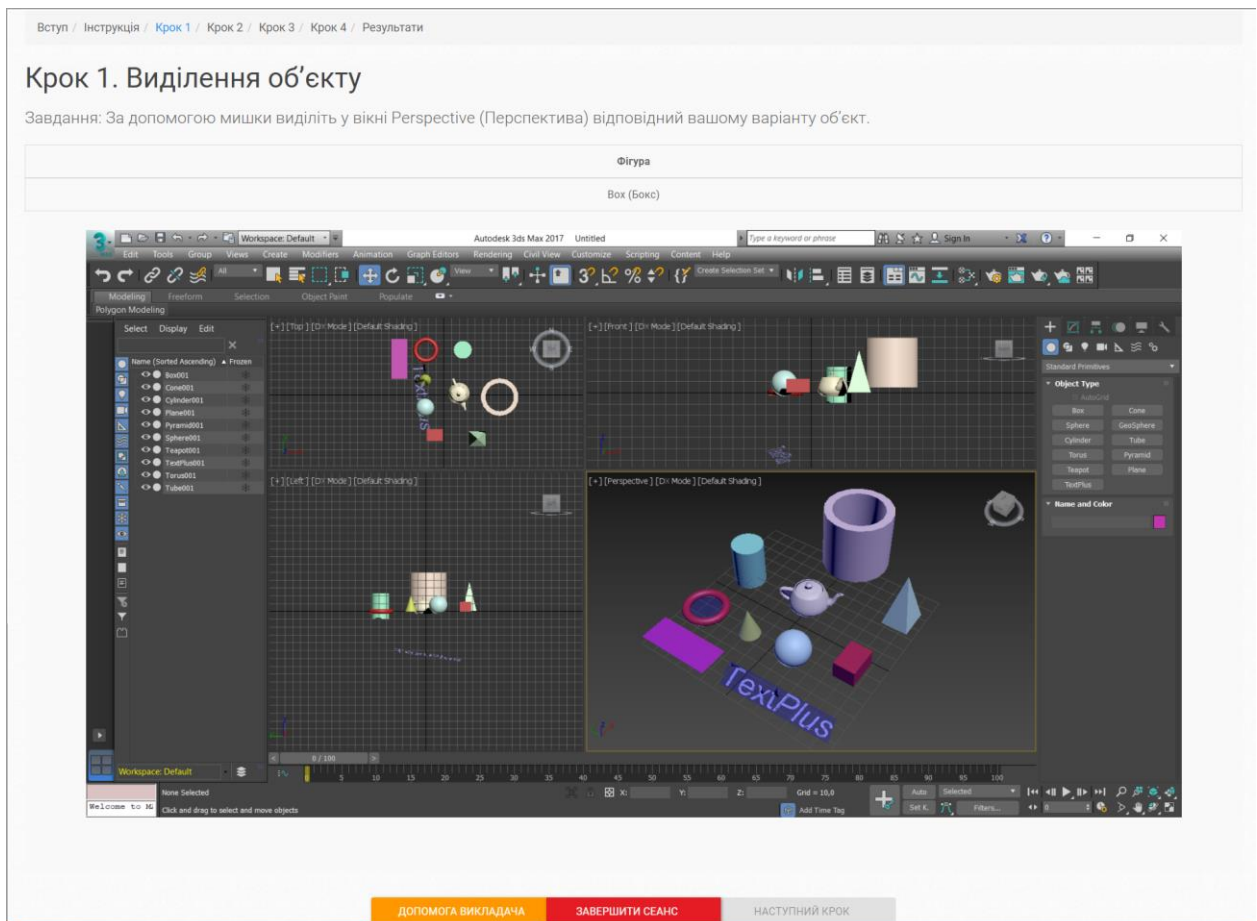


Рисунок 3.18 – Частина початкового вікна першого кроку проходження тренажеру

Для того, щоб виконати перший крок, користувачу необхідно натиснути на червоний паралелепіпед, після чого з'явиться повідомлення про успішне проходження кроку та активується кнопка «Наступний крок».

Зображення інтерфейсу змінює свій вигляд після того, як користувач обирає правильні варіанти відповідей. Наприклад, на рисунку 3.19 показано, що після того, як користувач обрав червоний паралелепіпед, попередній рисунок змінився на рисунок з обраним паралелепіпедом.

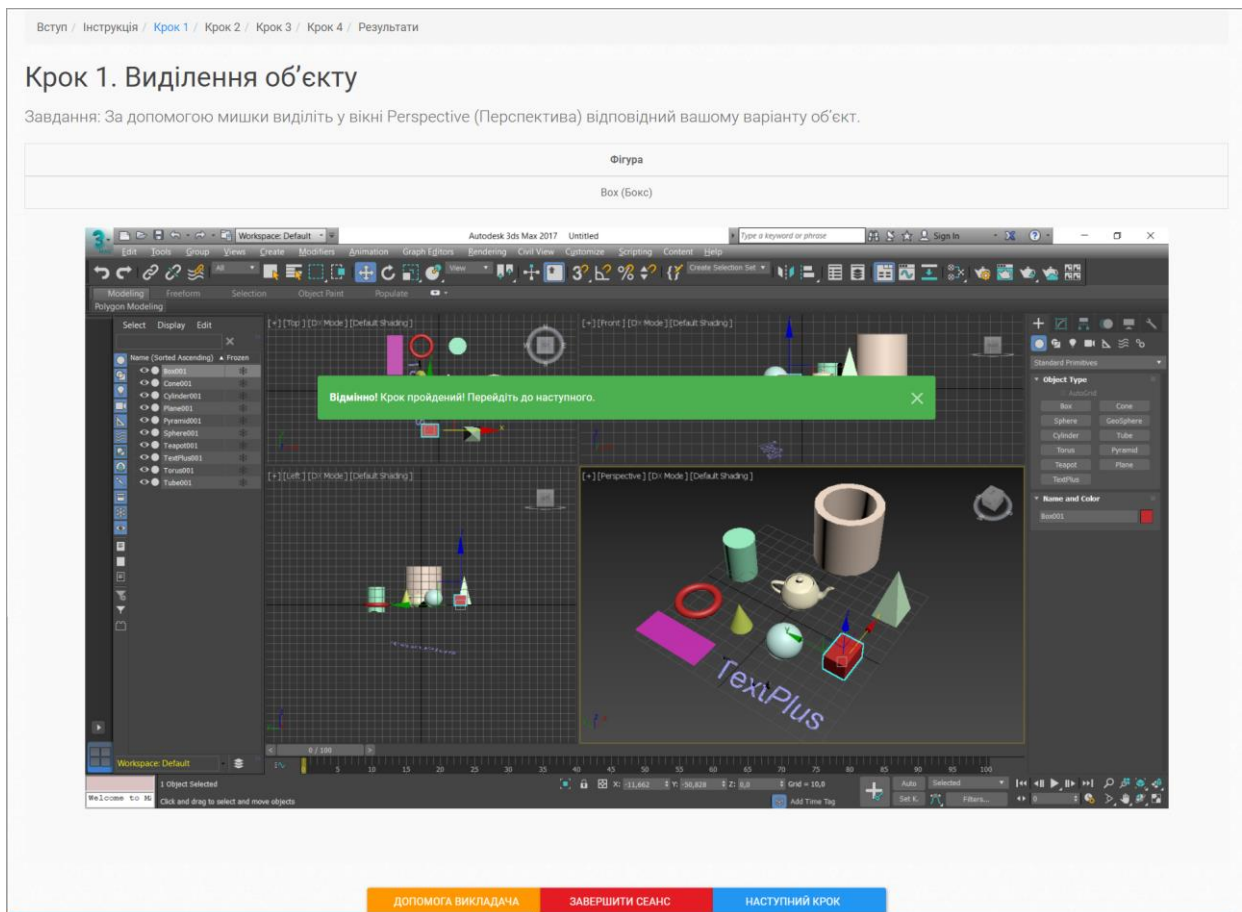


Рисунок 3.19 – Видяг вікна після успішного проходження першого кроку тренажеру

Деякі кроки вимагають від користувача вводити числові значення у поля для вводу. На рисунку 3.20 показано стан вікна тренажеру після вводу числових значень у поля (нижня частина екрану) та натиснення кнопки перевірення.

Якщо після натискання кнопки «Перевірити» поле для вводу містить правильне значення, то його нижня границя підсвічується зеленим кольором, якщо значення невірне або поле пусте, то підсвічується червоним кольором.

Також на рисунку 3.20 можна побачити, що кількість спроб, які залишилися у користувача, показуються у вигляді тексту на кнопці «Перевірити».

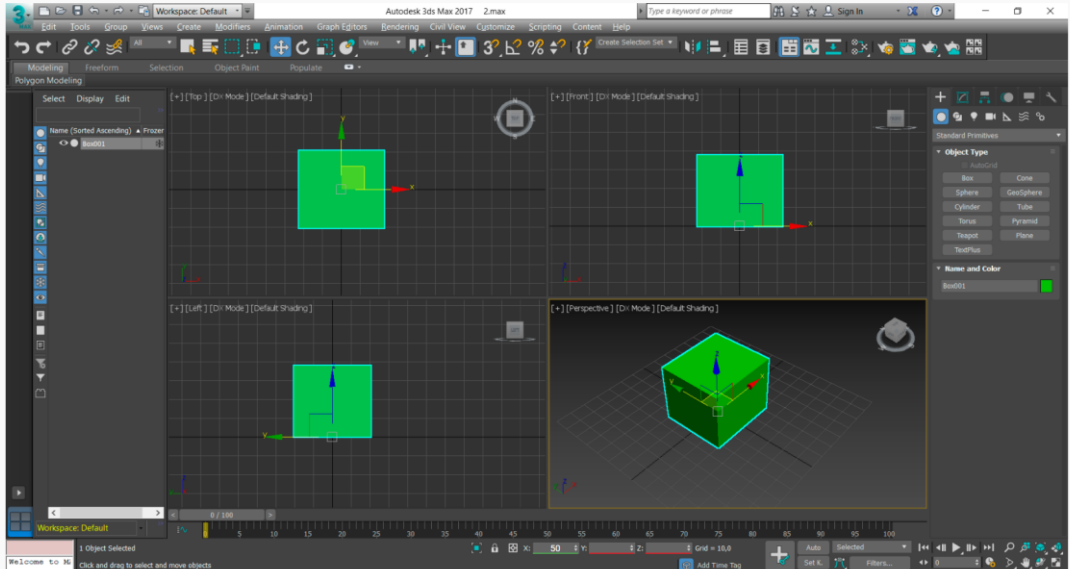
Кнопка «Перевірити» потрібна для перевірки значень, введених в поля для вводу.

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

## Крок 2. Переміщення об'єктів

Завдання: Перемістити об'єкт Box в точку із заданими координатами згідно варіанту. Для виконання завдання слід обрати відповідну команду на головній панелі інструментів Main Toolbar. Далі виділити об'єкт для переміщення, потім в області Transform Type-In у відповідних полях по чергово ввести значення координат точки X, Y, Z.

X	Y	Z
50	25	-50



ПЕРЕВІРИТИ (ЩЕ СПРОБ - 2)

ДОПОМОГА ВИКЛАДАЧА ЗАВЕРШИТИ СЕАНС НАСТУПНИЙ КРОК

Рисунок 3.20 – Вигляд вікна з полями для вводу

На рисунку 3.21 показано стан вікна, коли користувач перевищив ліміт дозволених спроб на проходження кроку. Спроба знімається кожний раз коли користувач натискає на елемент інтерфейсу 3DS Max, який не призначений для виконання дій, наведених в завданні. Також спроба знімається після натискання кнопки «Перевірити» поля для вводу містять некоректні значення.

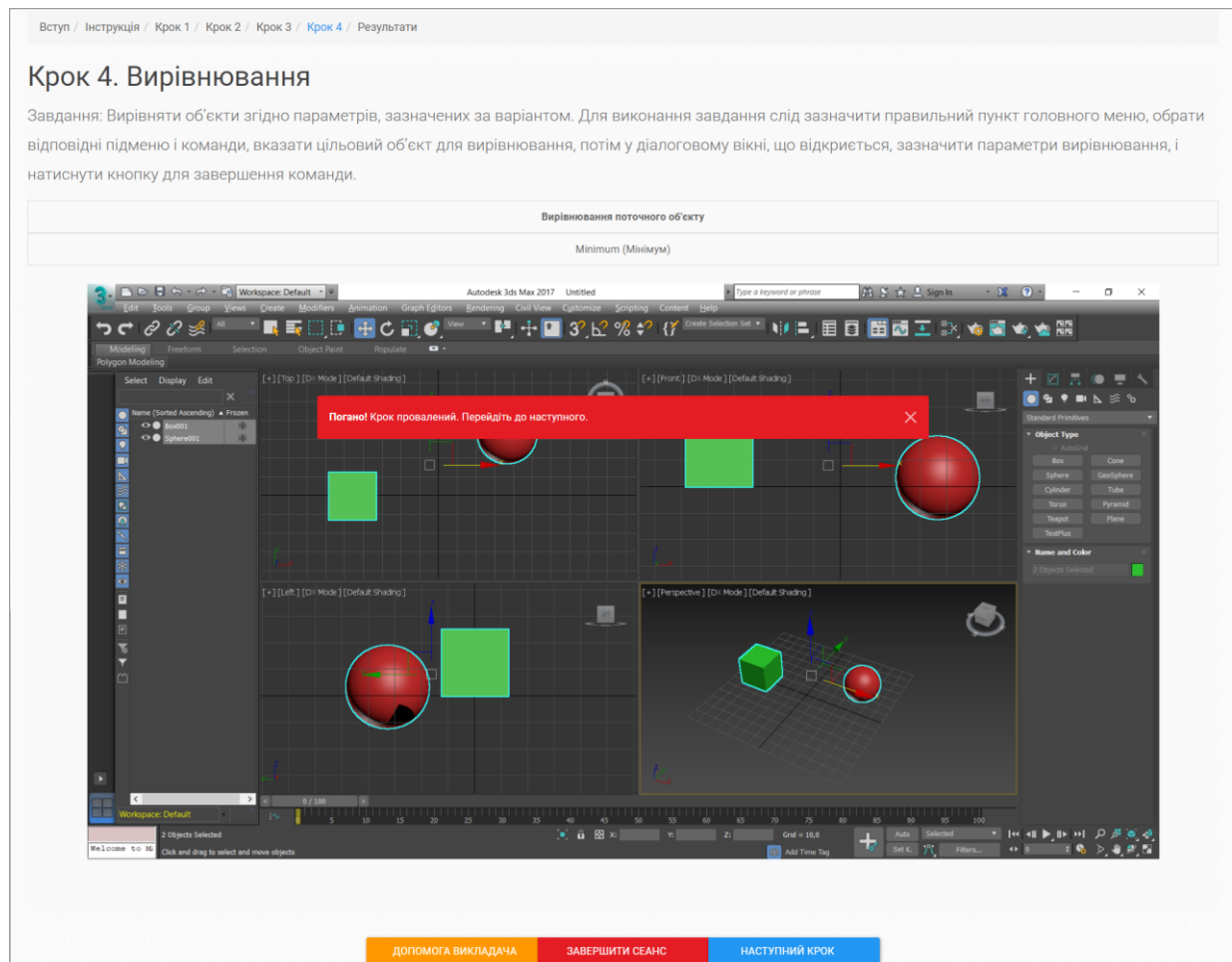


Рисунок 3.21 – Вигляд вікна при досягненні ліміта спроб на проходження кроку

Після завершення проходження усіх кроків тренажеру користувач переходить на вікно результатів (рис. 3.22). Вікно містить дані про час, затрачений на проходження сеансу тренажеру та кількість балів у відсотках. Далі необхідно натиснути кнопку «Завершити сеанс» для відправлення результатів до сервера.

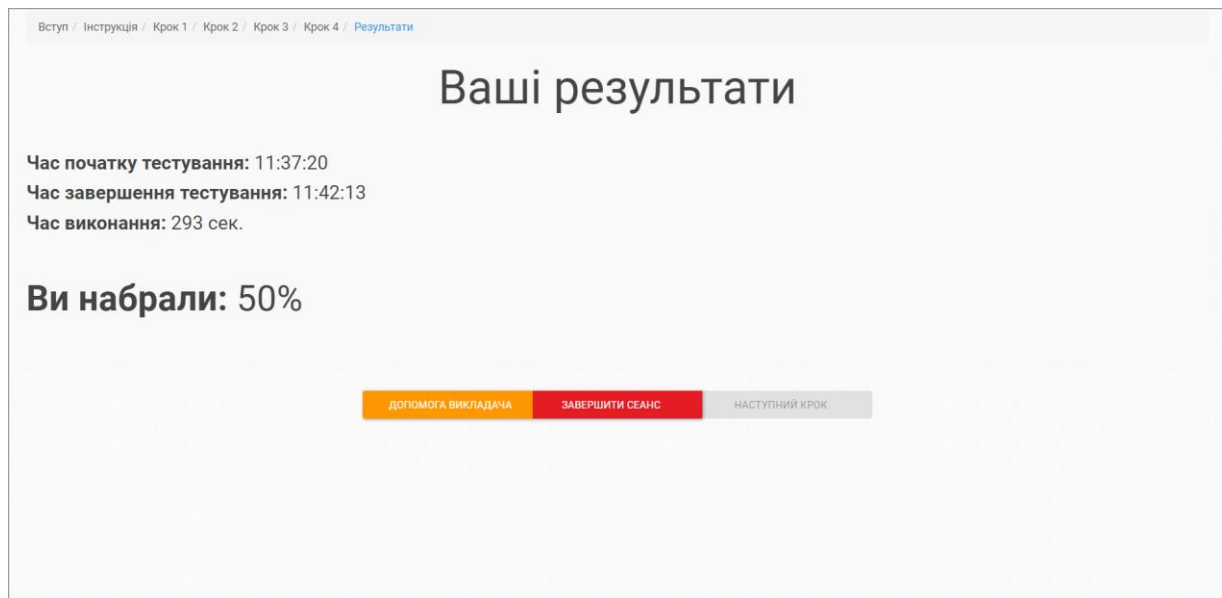


Рисунок 3.22 – Вигляд вікна коли користувач досягнув ліміта спроб на проходження кроку

Таким чином, можна зробити висновок, що поставлена мета дослідження виконана – розроблено комплекс тренажерів для вивчення тем курсу «Комп’ютерні технології дизайну» системи дистанційної освіти СумДУ. Результати роботи з розробки комплексу тренажерів були представлені на конференції ІМА-2021 [19]

Практичне значення розробки підтверджується актом впровадження (додаток Г).

## ВИСНОВКИ

Метою даної роботи є розробка віртуальних навчальних систем – тренажерів для дистанційного курсу. Кожний тренажер призначений для закріплення навичок з однієї теми.

В роботі детально досліджена предметна область, проведено аналіз подібних віртуальних тренажерів, виявлені їх основні проблеми та переваги, на основі чого сформульовано постановку задачі та визначені вимоги до функціональних характеристик тренажерів, що розробляються. Також проаналізовано існуючі технології, які можна використати при створенні веб-додатків подібного роду.

Проведено структурно-функціональне моделювання, представлені етапи проектування віртуального навчального тренажера, які стосуються моделювання його структури.

На етапі практичної реалізації розроблено два інтерактивних веб-тренажери для вивчення програмного продукту Autodesk 3DS Max. Розроблено інтерфейс обидвох тренажерів, його локалізовано трьома мовами.

Для кожного тренажеру створено 4 кроки завдань у 10 варіантах проходження. Для реалізації кожного кроку і кожного варіанту підготовлені відповідні зображення інтерфейсу програми, розмічено на них інтерактивні області та реалізовано обробку дій користувача при взаємодії з ними. Для цього було розширено функціонал методів для роботи із svg зображеннями.

Тренажери були інтегровані до системи сайту дистанційного навчання СумДУ. Кожен тренажер був протестований перед розгорненням на сервері.

Практичне значення роботи полягає в тому, що впровадження розроблених тренажерів в дистанційному курсі «Комп'ютерні технології дизайну» дозволить студентам закріпити навички роботи з програмою 3DS Max.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Особенности использования автоматизированного контроля знаний студентов: С.И. Дорошенко, В. С. Никольская, 2008. 79 с.
2. Безмежні можливості 3д графіки: об'ємні герої, рухомі зображення, комп'ютерна анімація. URL: <https://koloro.ua/ua/blog/3d-tehnologii/bezgranichnye-vozmozhnosti-3d-grafiki-obemnye-geroi-dvizhuwiesya-izobrazheniya-kompyuternaya-animaciya.html> (дата звернення: 25.04.2021).
3. Лучшие программы для 3d-моделирования. URL: <https://nastroyvse.ru/programs/rating/luchshie-programmy-dlya-3d-modelirovaniya.html> (дата звернення: 25.04.2021).
4. Застосування навчальних комп'ютерних інтерактивних тренажерів здобувачами вищої освіти інженерних спеціальностей в умовах інформаційно-освітнього середовища: Доценко Н. А. Миколаїв: Миколаївський національний аграрний університет, 2018. 119 с.
5. 10 javascript библиотек для svg анимации - URL: <https://postovoy.net/10-javascript-bibliotek-dlya-svg-animacii.html> (дата звернення: 25.04.2021).
6. HTML svg element - URL: <https://developer.mozilla.org/en-US/docs/Web/SVG/Element/svg> (дата звернення: 25.04.2021).
7. Export high-quality, optimized SVG - URL: <https://helpx.adobe.com/illustrator/how-to/export-svg.html> (дата звернення: 25.04.2021).
8. Tools to Visualize and Edit SVG Paths (Kinda!) – URL: <https://css-tricks.com/tools-visualize-edit-svg-paths-kinda/> (дата звернення: 25.04.2021).
9. Inkscape: creating and using paths - URL: <https://fedoramagazine.org/inkscape-creating-and-editing-paths/> (дата звернення: 25.04.2021).
10. Тест «Autodesk 3DS Max. Уровень 1. Основы 3D - моделирования» - URL: <https://www.specialist.ru/test/655> (дата звернення: 25.04.2021).
11. Как хорошо вы знаете Европу? – URL: <https://motovskikh.ru/europe/> (дата звернення: 25.04.2021).

12. Інтерактивний додаток для вивчення побудови геометричних примітивів з курсу "Комп'ютерна графіка" [Текст]: робота на здобуття кваліфікаційного рівня бакалавр; спец.: 122 - комп'ютерні науки, освітньо-професійна програма «Інформаційні технології проектування» / С.С. Дуванов; наук. кер. І.В. Баранова. – Суми: СумДУ, 2020. – 109 с.
13. Проектування інформаційних систем: навч. посібн. / Авраменко В.С., Авраменко А.С., 2017. 213 с.
14. Методологія IDEF0 – URL: [https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6\\_2](https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_2) (дата звернення: 28.04.2021).
15. Основы UML — диаграммы использования (use-case) - URL: <https://pro-prof.com/archives/2594> (дата звернення: 26.04.2021).
16. Діаграма станів – URL: <http://flash.retejo.info/cxefpagxo/uml/diagrama-staniv> (дата звернення: 28.04.2021).
17. Проектування інформаційних системи на основі уніфікованої мови моделювання – URL: <https://sites.google.com/site/analizvimogdopz/lekcii/uml> (дата звернення: 28.04.2021).
18. Uml діаграма компонентів опис. Основні діаграми мови UML – URL: <https://zhumor.ru/uk/computer/uml-diagramma-komponentov-opisanie-osnovnyye-diagrammy-yazyka-uml-primer.html> (дата звернення: 28.04.2021).
19. Інтерактивний веб-тренажер для підтримки вивчення 3ds Max. / Ісаєв В.В., Баранова І.В. // Інформатика, математика, автоматика (ІМА-2021) - Матеріали та програма Міжн. науково-техн. конф. студентів та молодих учених, м.Суми–Нур-Султан, 19–23 квітня 2021 р. / Відп. за вип. О.О.Дрозденко. – Суми, Сумський держ.ун-т, 2021. – С.91.
20. Постановка цілей по SMART – приклади, критерії – URL: <https://pdatu.edu.ua/images/vihovna-robota/psiholog/ps10.pdf> (дата звернення: 28.04.2021).
21. Що таке діаграма Ганта та як нею правильно користуватися? – URL: <https://nachasi.com/2020/09/03/gantt-chart/> (дата звернення: 28.04.2021).

## **ДОДАТОК А**

### **ТЕХНІЧНЕ ЗАВДАННЯ**

#### **1. Призначення й мета створення інтерактивних веб-додатків**

##### **1.1 Призначення інтерактивного веб-додатку**

Інтерактивні веб-додатки (тренажери) призначені для перевірки теоретичних та практичних знань студентів з побудови та редагування геометричних примітивів в середовищі Autodesk 3DS Max.

##### **1.2 Мета створення інтерактивного веб-додатку**

Підвищення якості дистанційного навчання, автоматизований контроль знань студентів.

##### **1.3 Цільова аудиторія**

Цільовою аудиторією є студенти.

#### **2 Вимоги до інтерактивних веб-додатків**

##### **2.1 Вимоги до інтерактивного веб-додатку в цілому**

###### **2.1.1 Вимоги до структури й функціонування інтерактивного веб-додатку**

Веб-додаток повинен мати вигляд односторінкової веб-сторінки, написаної стандартними засобами веб-розробки (HTML, CSS, JavaScript) з можливим використанням бібліотек третіх сторін. Основний зміст веб-додатку повинен бути поміщений у фрейм, що дозволить переходити між етапами веб-додатку без оновлення сторінки. Після завершення виконання всіх кроків тренажеру користувач повинен побачити час, витрачений на проходження тренажеру та кількість набраних балів.

### **2.1.2 Вимоги до персоналу**

Оскільки додаток не вимагає постійної підтримки, тому нема потреби залучати персонал. Після інтеграції додатку до сервера СумДУ веб-тренажер не потребує подальшої підтримки.

### **2.1.3 Вимоги до збереження інформації**

Вся інформація, з якою працює додаток повинна зберігатися сервері, а саме JSON файли локалізації та зображення інтерфейсу.

### **2.1.4 Вимоги до розмежування доступу**

Веб-додаток не повинен мати розмежування по типу доступу, такі як адміністратор або користувач, тому кожен користувач тренажеру вважається відвідувачем.

## **2.2 Структура веб-додатку**

### **2.2.1 Загальна інформація про структуру веб-додатку**

Додаток являє собою односторінкову веб-сторінку, з вбудованим в неї якої засобами JavaScript фреймом. Перехід між етапами веб-додатку повинно здійснюватися за допомогою оновлення вмісту фрейма. Першим розділом веб-додатку повинен бути «Вступ», який містить назву тренажеру, блок вибору мови веб-інтерфейсу, панель вибору варіанту проходження веб-додатку та логотип тренажеру. Другим розділом додатку є «Інструкція», що містить вказівки щодо роботи з інтерфейсом веб-додатку. Наступними чотирма розділами є етапи виконання тренажера «Крок 1», «Крок 2», «Крок 3», «Крок 4». Кожен розділ містить завдання, яке повинен виконати студент, та інтерактивне зображення вікна Autodesk 3DS Max. Проходження тренажеру має бути лінійним. Останнім розділом є «Результати», де користувач зможе дізнатися процент правильних відповідей та час проходження сеансу. Користувач може потрапити на розділ «Результати» передчасно (пройшовши не всі кроки), натиснувши кнопку «Завершити сеанс». Також користувач може запросити допомогу у викладача, натиснувши кнопку «Допомога викладача».

## 2.2.2 Навігація

Панель навігації в кожному розділі повинна мати вигляд рядка, який знаходиться у верхній частині вікна та містить назву усіх розділів, присутніх у тренажері. Панель навігації повинна показувати користувачу на якому етапі він знаходиться.

Перехід між кроками тренажеру здійснюється за допомогою кнопки «Наступний крок».

## 2.2.3 Наповнення веб-додатку (контент)

Оскільки основний зміст веб-додатку знаходиться у фреймі, то зміна кроків тренажеру повинна здійснюватися шляхом заміщення попереднього вмісту фрейма на новий.

## 2.2.4 Дизайн та структура додатку

Стильове оформлення додатку повинно відповідати палітрі кольорів сайту дистанційного навчання СумДУ.

Розділ «Вступ» (рис. 1) повинен містити панель навігації, логотип та назву тренажеру, інформацію про кількість кроків, значки у вигляді прапорів для вибору мови інтерфейсу та список з вибором варіанту.

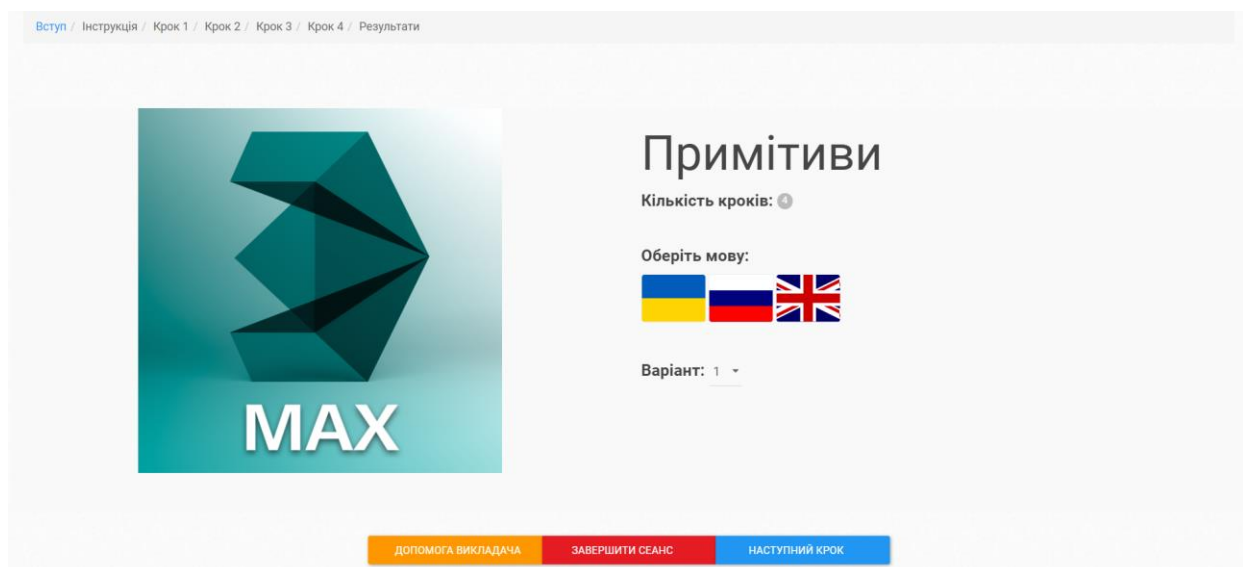


Рисунок 1 – Вікно «Вступ»

Розділ «Інструкція» (рис. 2) повинен містити текст та зображення, які допоможуть користувачу зрозуміти процес проходження тренажеру.

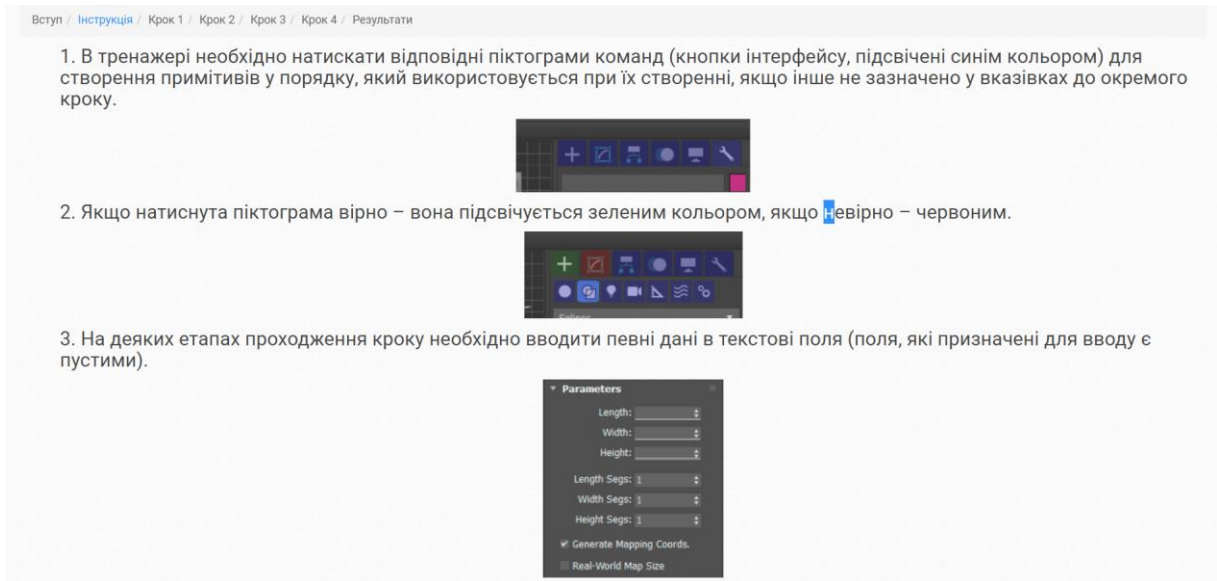


Рисунок 2 – Частина вікна розділу «Інструкція»

Розділи із кроками (рис. 3) повинні містити блок з текстом завдання та інтерактивне зображення, яке симулює роботу з 3DS Max. Розділ «Результати» (рис. 4) повинен містити інформацію про результати проходження сеансу.

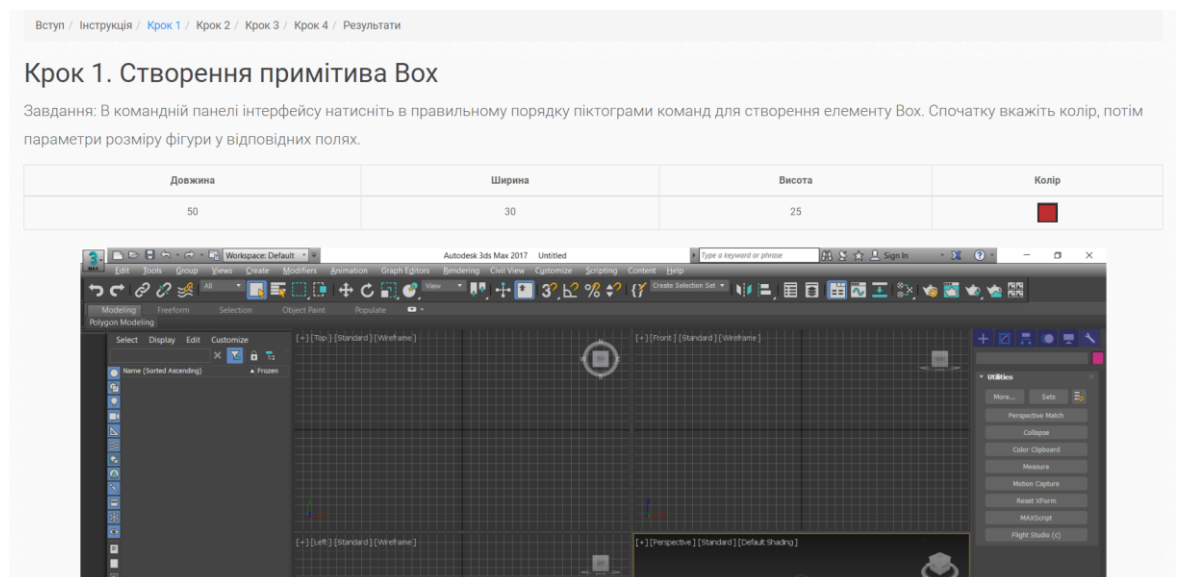


Рисунок 3 – Частина вікна «Крок 1»

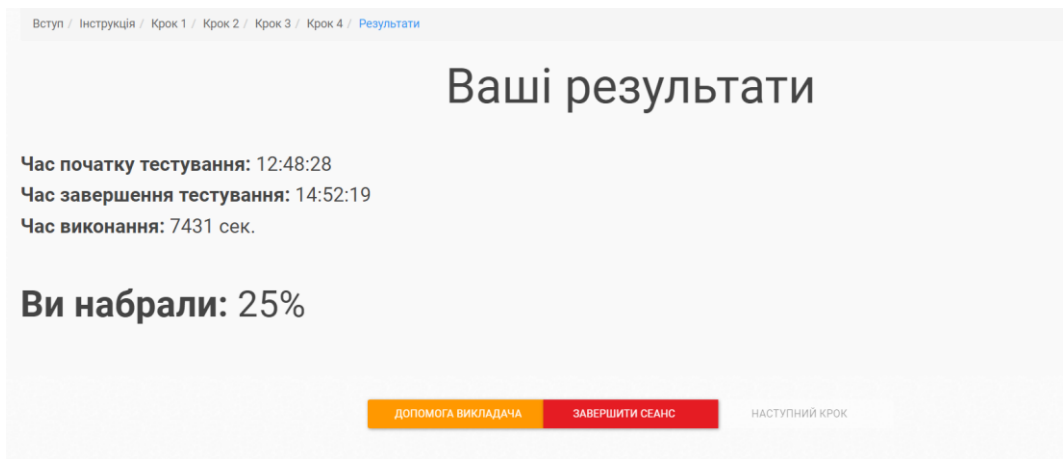


Рисунок 4 – Вікно «Результати»

Кожен розділ повинен містити кнопки: «Допомога викладача», «Завершити сеанс», «Наступний крок», а також кнопка «Перевірити» (крім першого, другого та останнього розділу). Стиль оформлення додатку повинен бути однаковим для усіх кроків.

### 2.2.5 Система навігації (карта веб-додатку)

Карта додатку представлена на рисунку 5.

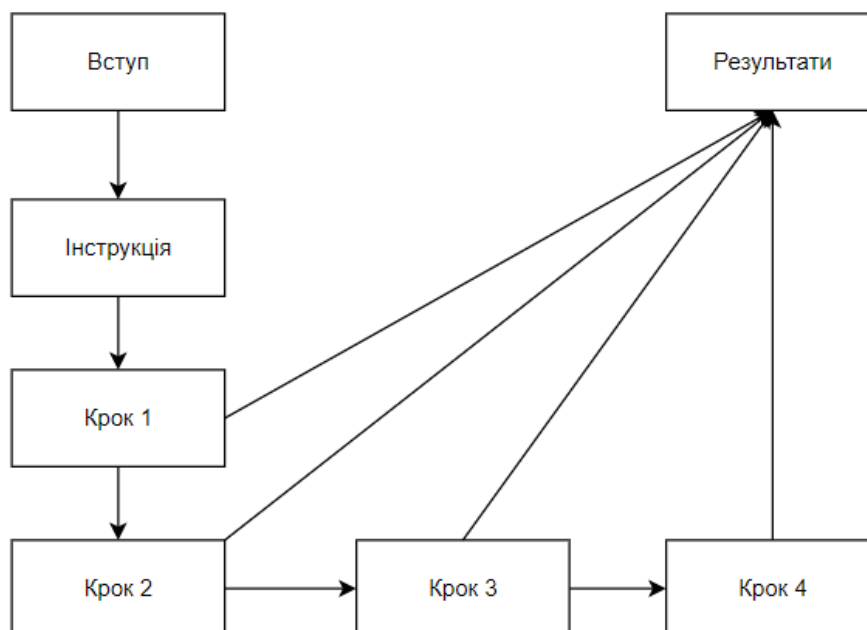


Рисунок 5 – Карта веб-додатку

## **2.3 Вимоги до функціонування системи**

### **2.3.1 Потреби користувача**

Перед початком проходження тренажеру користувач повинен мати можливість обрати в розділі «Вступ» мову локалізації інтерфейсу: українську, російську або англійську мову.

Кожен крок повинен мати кнопки навігації для переходу між кроками.

Деякі кроки, які потребують перевірки даних повинні містити кнопку «Перевірити», яка перевіряє коректність введених користувачем даних. Після натиснення кнопки неправильні відповіді завдань повинні підсвічуватися червоним кольором, а правильні – зеленим.

Після проходження всіх кроків користувач повинен мати можливість побачити свій результат. Вікно з результатами повинне показувати користувачу такі дані: час початку та час завершення проходження тренажеру (години/хвилини/секунди), загальний час виконання (у секундах), кількість набраних балів (у відсотках).

### **2.3.2 Функціональні вимоги**

На основі потреб користувача були визначені такі функціональні вимоги:

- Зміна мови інтерфейсу;
- Вибір варіанту проходження тренажеру;
- Запит допомоги у викладача;
- Передчасне завершення проходження тренажеру та отримання результатів

## **2.4 Вимоги до видів забезпечення**

### **2.4.1 Вимоги до лінгвістичного забезпечення**

Інтерфейс тренажеру повинен бути локалізований трьома мовами: українська, російська та англійська.

### **2.4.2 Вимоги до програмного забезпечення**

Клієнтська частина повинна мати наступне програмне забезпечення:



- Будь-який сучасний браузер;
- Підтримка JavaScript.

### 3 Склад і зміст робіт зі створення веб-додатку

Порядок етапів створення веб-додатку наведено в таблиці 1.

Таблиця 1 – Етапи створення веб-тренажеру

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Ознайомлення з існуючим шаблоном	2 дні
2	Вибір технології для впровадження симуляції роботи з 3DS Max	2 дні
3	Корегування макету веб-сторінки	3 дні
4	Впровадження можливості вибору варіанта	1 день
5	Підготовка інтерактивних зображень	1 день
6	Розмітка інтерактивних областей зображення та формування на їх основі HTML документів	2 дні
7	Впровадження основного функціоналу: інтерактивність та зміна зображень, варіативність	5 днів
8	Корегування механізму нарахування балів	1 день
9	Створення блоку з інструкцією	1 день
10	Локалізація інтерфейсу	1 день
10	Тестування системи та виправлення помилок	3 дні
11	Виправлення недоліків інтерфейсу	2 дні
12	Оформлення документації	7 днів
	<b>Загальна тривалість робіт (з урахуванням резервного строку на налагодження й виправлення помилок) і строк закінчення проекту</b>	31

#### **4 Вимоги до складу й змісту робіт із введення веб-додатку в експлуатацію**

Перед перенесенням веб додатку на сервер необхідно не забути вимкнути режим розробника, якщо такий присутній. Інтеграція з сервером СумДУ повинна відбуватися згідно з вимогами адміністратора дистанційного сайту СумДУ.

## ДОДАТОК Б

### ПЛАНУВАННЯ РОБІТ

**Деталізація мети проекту методом SMART.** Мета проекту – розробити інтерактивний веб-тренажер, для вивчення функціоналу програми Autodesk 3DS Max шляхом імітації його інтерфейсу. Продукт створюється для подальшої інтеграції до дистанційного курсу дисципліни «Комп’ютерна графіка». Постановка мети методом SMART надає можливість спрогнозувати досягнення мети [20]. Результат деталізації методом SMART представлений у табл. Б.1.

Таблиця Б.1 – Деталізація мети методом SMART

Specific (конкретна)	Розробка інтерактивного веб-тренажеру для дистанційного курсу за допомогою якого можна автоматизувати навчання, проводити тестування та оцінювання студентів.
Measurable (вимірювана)	Інтегрування тренажеру до системи тренажерів університету.
Achievable (досяжна)	Створення проектами засоби середовища розробки VS Code, браузера Google Chrome та веб-серверу Apache 2.4.
Relevant (реалістична)	Для успішного виконання проекту необхідні знання веб-розробки, зокрема HTML, CSS, JavaScript. Також додатково потрібні знання фреймворків Bootstrap, jQuery та mustache.js.
Time-framed (обмежена у часі)	Робота повинна бути виконана згідно термінів календарного плану

**Планування змісту структури робіт.** Для планування змісту структури робіт проекту використовується WBS-діаграма – ієрархічна схема, яка розбиває проект на дрібні задачі та відображає їх відношення одна до одної. WBS-діаграма представлена на рисунку Б.1.

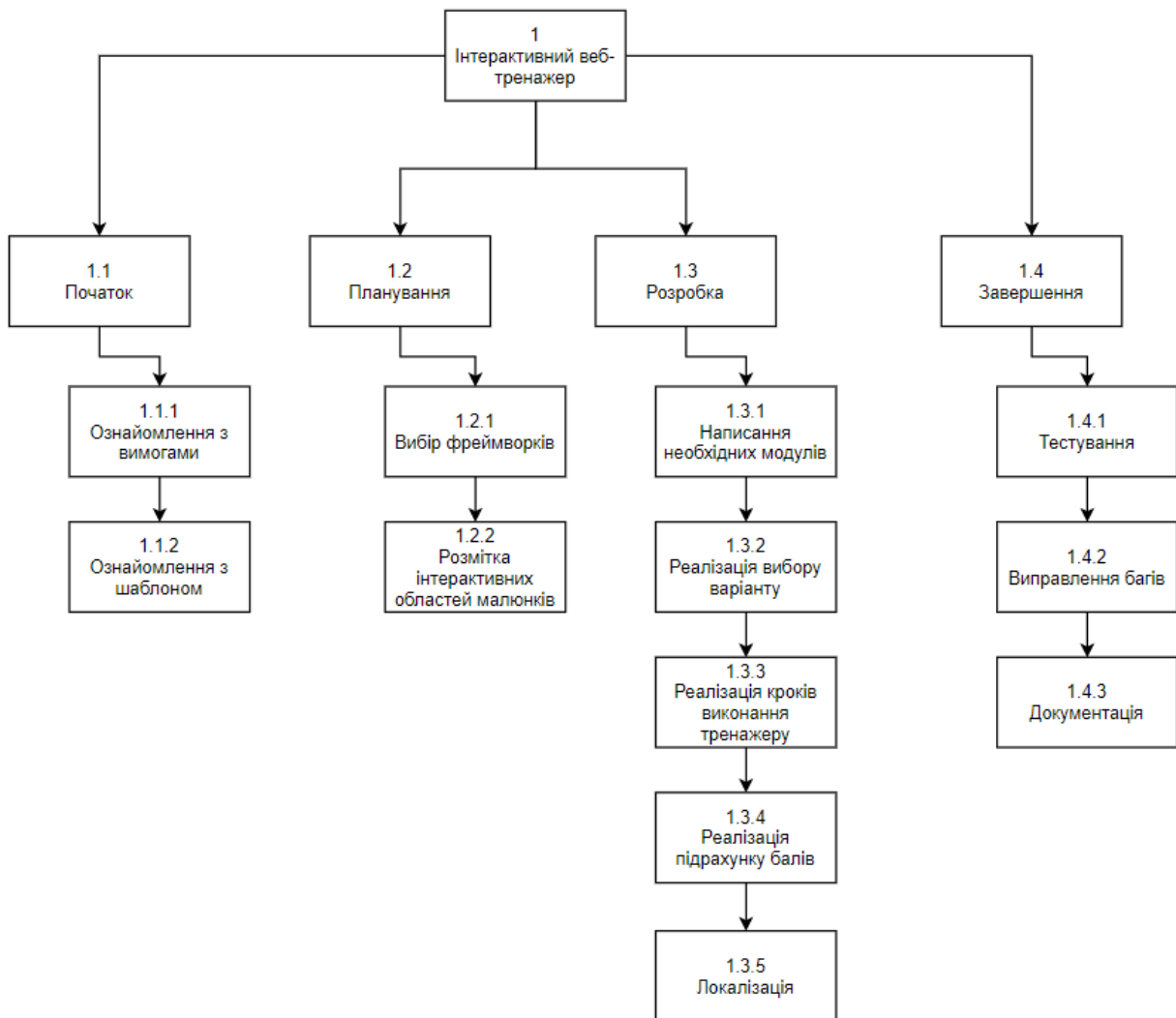


Рисунок Б.1 – WBS-діаграма робіт проекту

Для планування структури організації учасників проекту та їх взаємовідношень один із одним використовується OBS діаграма. OBS діаграма разом із діаграмою Ганта представлена на рисунку Б.2.

Список учасників проекту наведено в табл. Б.2.

Таблиця Б.2 – Учасники проекту

Роль	Ім'я	Проектна Роль
Керівник проекту	Баранова І. В	Формування технічного завдання та підготовка шаблонів кроків тренажерів
Розробник	Ісаєв В. В.	Оформлення документації, створення інтерфейсу, написання коду
Тестувальник	Ісаєв В. В.	Тестування функціоналу

**Діаграма Ганта.** Діаграма використовується для графічного відображення списку робіт (рис. Б.2) згідно календарного плану [21]. За день вважається 3 години. Загальна кількість днів від початку проекту до завершення складає 50 днів. Діаграма Ганта зображена на рисунку Б.3.

Task Name	Duration	Start	Finish
<b>Створення комплексу інтерактивних тренажерів для дистанційного курсу з вивчення 3DS Max</b>	<b>50 days</b>	<b>05.04.21 08:00</b>	<b>24.05.21 17:00</b>
<b>Оформлення документації</b>	<b>8 days</b>	<b>05.04.21 08:00</b>	<b>12.04.21 17:00</b>
Розробка ТЗ	2 days	05.04.21 08:00	06.04.21 17:00
Планування змісту структури робіт	1 day	07.04.21 08:00	07.04.21 17:00
Розробка UML діаграм	4 days	08.04.21 08:00	11.04.21 17:00
Оцінка ризиків	1 day	12.04.21 08:00	12.04.21 17:00
<b>Підготовка ресурсів</b>	<b>3 days</b>	<b>13.04.21 08:00</b>	<b>15.04.21 17:00</b>
Встановлення необхідного програмного забезпечення	1 day	13.04.21 08:00	13.04.21 17:00
Створення скріншотів інтерфейсу 3DS Max	1 day	14.04.21 08:00	14.04.21 17:00
Розмітка інтерактивних областей	1 day	15.04.21 08:00	15.04.21 17:00
<b>Створення інтерфейсу веб-додатку</b>	<b>9,5 days</b>	<b>16.04.21 08:00</b>	<b>25.04.21 12:00</b>
Проектування	2 days	16.04.21 08:00	17.04.21 17:00
Створення макету	0,5 days	18.04.21 08:00	18.04.21 12:00
Інтеграція інтерактивних областей до HTML файлів	1 day	18.04.21 13:00	19.04.21 12:00
Формування стилів	1 day	19.04.21 13:00	20.04.21 12:00
Верстання	5 days	20.04.21 13:00	25.04.21 12:00
<b>Написання основної логіки</b>	<b>19,5 days</b>	<b>25.04.21 13:00</b>	<b>14.05.21 17:00</b>
Налаштування веб-серверу	0,5 days	25.04.21 13:00	25.04.21 17:00
Ознайомлення з кодом шаблону	5 days	26.04.21 08:00	30.04.21 17:00
Написання коду модулів	14 days	01.05.21 08:00	14.05.21 17:00
<b>Тестування</b>	<b>8 days</b>	<b>15.05.21 08:00</b>	<b>22.05.21 17:00</b>
Пошук несправностей	1 day	15.05.21 08:00	15.05.21 17:00
Виправлення недоліків інтерфейсу	3 days	16.05.21 08:00	18.05.21 17:00
Виправлення недоліків коду	4 days	19.05.21 08:00	22.05.21 17:00
<b>Розгортання</b>	<b>2 days</b>	<b>23.05.21 08:00</b>	<b>24.05.21 17:00</b>
Розміщення веб-додатку на хостингу дистанційного курсу	1 day	23.05.21 08:00	23.05.21 17:00
Завершення проекту	1 day	24.05.21 08:00	24.05.21 17:00

Рисунок Б.2 – Список робіт діаграми Ганта

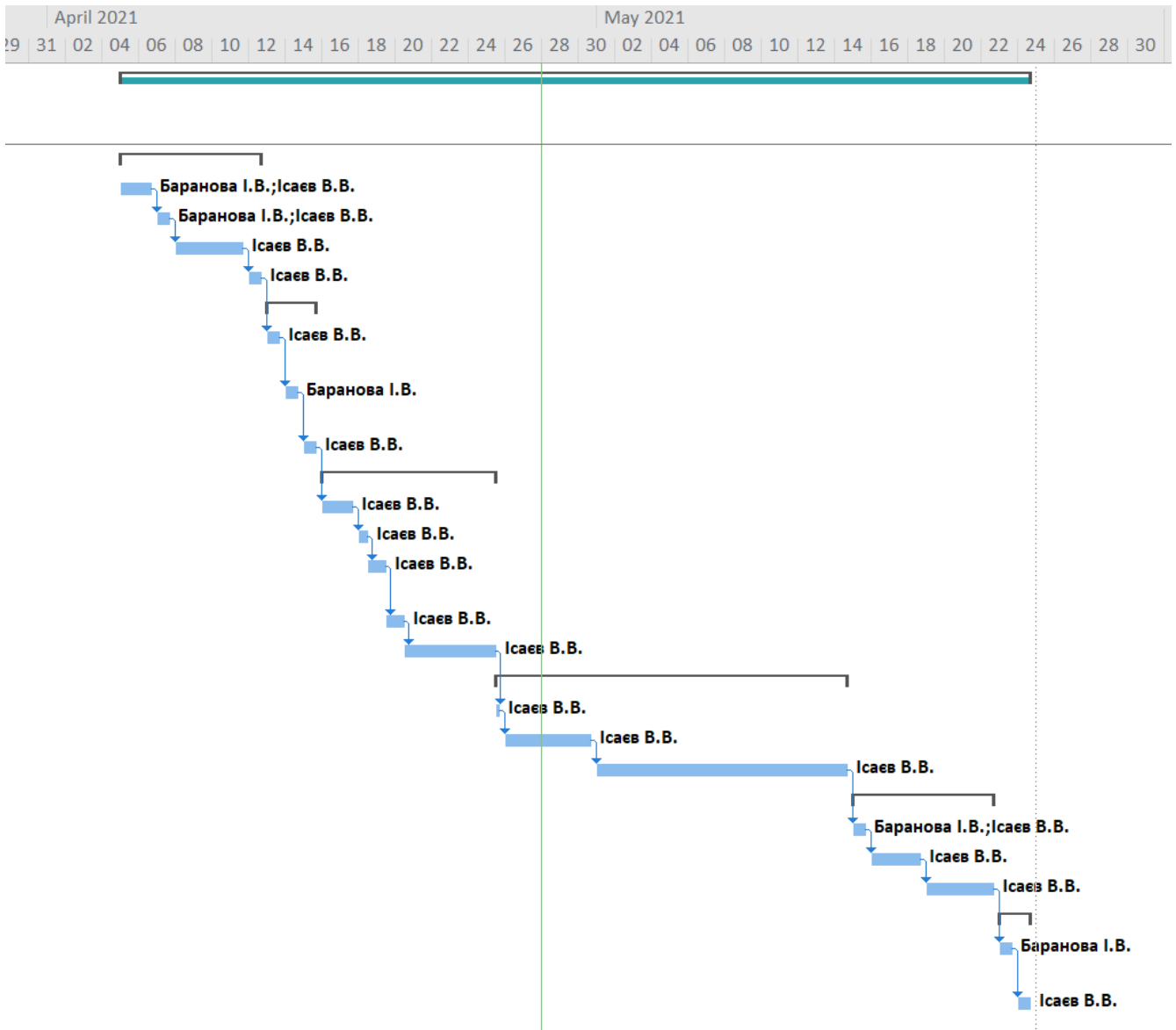


Рисунок Б.3 – Діаграма Ганта

**Аналіз ризиків.** Аналіз ризиків виконується для передбачення незапланованих ситуацій, які можуть виникати під час створення проекту. Визначаються причини виникнення ризиків та знаходяться способи їх подолання.

Класифікуємо ризики за показниками ймовірності виникнення (табл. Б.3).

Таблиця Б.3 – Аналіз ризиків

Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	План	Тип стратегії реагування
Відкритий	Обрана для розробки технологія перестає відповідати необхідним вимогам на пізньому етапі проекту	Середня	Високий	Попередній аналіз обраної технології згідно вимогам у технічному завданні	Пом'якшення
Відкритий	Виникнення помилок у кодї шаблону проекту	Низька	Високий	Перерозподіл календарного плану та виділення часу на виправлення помилок шаблону	Пом'якшення
Відкритий	Проблеми під час розгортання проекту на сервері	Низька	Середній	Консультація із адміністрацією серверу та виправлення помилок коду при їх наявності	Попередження
Відкритий	Неправильне розуміння задач, поставлених в технічному завданні	Висока	Високий	Детальна консультація із керівником проекту щодо суперечливих місць проекту	Пом'якшення
Відкритий	Недотримання термінів календарного плану	Середня	Високий	Перерозподіл календарного плану	Пом'якшення

## ДОДАТОК В

### ЛІСТИНГ КОДУ

#### Файл `trainer.config.json`

```
{
  "LANGUAGES": {
    "trainer2_ua": "Українська",
    "trainer2_ru": "Русский",
    "trainer2_en": "English"
  },
  "DEFAULT_LANG": "trainer2_ua",
  "TRAINER_SCORE": 100,
  "TRAINER_AUTHOR": "Vadim Isaev",
  "COURSE_AUTHOR": "Vadim Isaev",
  "PRODUCTION": false
}
```

#### Файл `trainer.steps.json`

```
[
  {
    "filename"      : "introduction",
    "score"         : 0,
    "hasScript"    : true
  },
  {
    "filename"      : "instruction",
    "score"         : 0,
    "hasScript"    : true
  },
  {
    "filename"      : "trainer2_step1",
    "score"         : 25,
    "hasScript"    : true
  },
  {
    "filename"      : "trainer2_step2",
    "score"         : 25,
    "hasScript"    : true
  },
  {
    "filename"      : "trainer2_step3",
    "score"         : 25,
    "hasScript"    : true
  },
  {
    "filename"      : "trainer2_step4",
    "score"         : 25,
    "hasScript"    : true
  },
  {
    "filename"      : "results",
    "score"         : 0,
    "hasScript"    : true
  }
]
```



```

Файл steps.css
.map {
  position: relative;
  text-align: center;
  width: 100%;
}

svg {
  position: absolute;
  top: 0;
  left: 0;
  height: 100%;
  width: 100%;
  display: inline !important;
}

.part {
  opacity: 0.2;
  fill: blue;
}

.part:hover {
  transition: opacity 0.2s ease;
  opacity: 0.5;
  fill: blue;
}

foreignObject input[type="text"] {
  font-size: 14px;
  color: white;
  height: 20px;
}

#mainImg {
  max-width: 100%;
  max-height: 100%;
  width: 90%;
  height: auto;
}

foreignObject {
  visibility: hidden;
}

.instruction-text {
  padding-left: 50px;
  padding-bottom: 110px;
}

```

## Файл introduction.html

```

<div class="row jumbotron " style="margin-top: 80px;">
  <div class="col-md-6 animated zoomIn">
    
  </div>
  <div class="col-md-6 animated fadeInDown">
    <h1>{{TRAINER_NAME}}</h1>
    <p>
      <h5><strong>{{STEP_COUNT}}:</strong> <span
class="badge">{{STEPS_COUNT}}</span></h5>
    <br/>

```

```

<h5><strong>{{CHOOSE_LANG}}:</strong></h5>
  <div class="flag-icon flag-icon-ua" title="ua" id="ua"></div>
  <div class="flag-icon flag-icon-ru" title="ru" id="ru"></div>
  <div class="flag-icon flag-icon-en" title="en" id="en"></div>
<br/><br/>
<h5><strong>{{VARIANT}}:</strong>
  <select id="variant-select" class="form-control" style="width: 40px;
padding: 5px;">
    <option selected value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
    <option value="6">6</option>
    <option value="7">7</option>
    <option value="8">8</option>
    <option value="9">9</option>
    <option value="10">10</option>
  </select>
</h5>
</p>
</div>
</div>

```

### Файл instruction.js

```

var userVariant;
var introduction = function () {
  this.postDispatch = function () {
    $("#endTrainer").addClass("disabled btn-default");
    $("div.flag-icon").each(function () {
      $(this).attr(
        "onclick",
        "window.location.href = '?lang=trainer2_' + $(this).attr(\"id\")" // need
to change trainerNUMBER_ according to current trainer
      );
    });
  };
  this.mustache = function () {
    return {
      STEPS_COUNT: Rotator.getStepsCount() - 3,
    };
  };
};

```

### Файл instruction.html

```

<link rel="stylesheet" href="/trainer/css/steps.css" type="text/css" />
<div class="instruction-text">
  <h4>{{INSTRUCTION1}}</h4>
  <div class="text-center">
    
  </div><br>
  <h4>{{INSTRUCTION2}}</h4>
  <div class="text-center">
    
  </div><br>
  <h4>{{INSTRUCTION3}}</h4>
  <div class="text-center">
    
  </div><br>

```

```
<h4>{{INSTRUCTION4}}</h4>
</div>
```

### Файл instruction.js

```
var userVariant = document.getElementById("variant-select").value;
var instruction = function () {
  this.postDispatch = function () {
    Rotator.enableNextButton();
  };
  this.mustache = function () {
    window.parent.document.title = I18N.getConstants()["VARIANT"] + " " +
userVariant;
    return {};
  };
};
```

### Файл helpers.js

```
function addSvgStep (order, imgInfo, partsBorders, vstep, callback) {
  $(".part").click(function () {
    if (order[0] == $(this).attr("id")) {
      $(this).css("fill", "green");
      if (order[0] == imgInfo.order[0]) {
        $("#mainImg").attr("src", imgInfo.path[0]);
        imgInfo.order.shift();
        imgInfo.path.shift();
      }
      if (partsBorders) {
        $(".part").css("fill", "blue");
        $(".part:gt(0)").hide();
        $(".part:lt(" + partsBorders.max[0] + ")").show();
        $(".part:lt(" + partsBorders.min[0] + ")").hide();
        partsBorders.min.shift();
        partsBorders.max.shift();
      }
      order.shift();
      if (order.length == 0) {
        $(".part").hide();
        callback();
      }
    } else {
      $(this).css("fill", "red");
      var attempts = vstep.getAttempts() - 1;
      if (attempts == 0) {
        $(".part").hide();
        $(".div.validation-alert-danger").fadeIn();
      }
    }
  });
}
```

```

        Rotator.enableNextButton();
    } else {
        vstep.setAttempts(attempts);
    }
}
});
}

```

### Файл `trainer1_step1.html`

```

<link rel="stylesheet" href="/trainer/css/steps.css" type="text/css" />
<script src="/trainer/scripts/helpers.js"></script>
<div id="step1" class="jumbotron page1">
    <div style="margin-left: 5px">
        <h3>{{STEP1_TITLE}}</h3>
        <p>{{STEP1_TASK}}</p>
        <table class="table table-bordered text-center">
            <thead>
                <tr>
                    <th scope="col" class="text-center">{{STEP1_TABL_HEADER1}}</th>
                    <th scope="col" class="text-center">{{STEP1_TABL_HEADER2}}</th>
                    <th scope="col" class="text-center">{{STEP1_TABL_HEADER3}}</th>
                    <th scope="col" class="text-center">{{STEP1_TABL_HEADER4}}</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td id="boxlength"></td>
                    <td id="boxwidth"></td>
                    <td id="boxheight"></td>
                    <td><img id="boxcolor" alt=""></td>
                </tr>
            </tbody>
        </table>
    </div>
    <div class="map">
        <svg id="svgmap" viewBox="0 0 1920 1030">
            <a href="javascript:void(0)">

```

```

        <title>Create</title>
        <path id="1" class="part" d="m 1670.7971,184.54387
0.3751,-32.65894 36.7666,0 -0.1875,32.82883 z">
    </a>
    <a href="javascript:void(0)">
        <title>Modify</title>
        <path id="2" class="part" d="m 1711.2574,185.10047
0.3751,-32.65894 h 36.7666 l -0.1875,32.82883 z">
    </a>
    <a href="javascript:void(0)">
        <title>Hierarchy</title>
        <path id="3" class="part" d="m 1751.6626,185.37047
0.3751,-32.65894 h 36.7666 l -0.1875,32.82883 z">
    </a>
    <a href="javascript:void(0)">
        <title>Motion</title>
        <path id="4" class="part" d="m 1791.7069,185.43676
0.3751,-32.65894 h 36.7666 l -0.1875,32.82883 z">
    </a>
    <a href="javascript:void(0)">
        <title>Display</title>
        <path id="5" class="part" d="m 1831.3533,185.30417
0.3751,-32.65894 h 36.7666 l -0.1875,32.82883 z">
    </a>
    <a href="javascript:void(0)">
        <title>Utilities</title>
        <path id="6" class="part" d="m 1871.066,185.63566 0.3751,-
32.65894 h 36.7666 l -0.1875,32.82883 z">
    </a>
    <a href="javascript:void(0)">
        <title>Geometry</title>
        <path id="7" class="part" d="m 1676.2505,190.59035 -
0.148,28.42258 27.3012,0.13831 -0.2218,-28.63005 z">
    </a>
    <a href="javascript:void(0)">
        <title>Shapes</title>
        <path id="8" class="part" d="m 1706.5124,190.58968 -
0.148,28.42258 27.3012,0.13831 -0.2218,-28.63005 z">
    </a>
    <a href="javascript:void(0)">
        <title>Lights</title>

```

```

        <path id="9" class="part" d="m 1736.5255,190.79715 -
0.148,28.42258 27.3012,0.13831 -0.2218,-28.63005 z">
    </a>
    <a href="javascript:void(0)">
        <title>Cameras</title>
        <path id="10" class="part"
            d="m 1766.6078,191.00461 -0.148,28.42258
27.3012,0.13831 -0.2218,-28.63005 z">
    </a>
    <a href="javascript:void(0)">
        <title>Helpers</title>
        <path id="11" class="part"
            d="m 1796.6404,190.94684 -0.148,28.42258
27.3012,0.13831 -0.2218,-28.63005 z">
    </a>
    <a href="javascript:void(0)">
        <title>Space Warps</title>
        <path id="12" class="part"
            d="m 1826.8724,190.88055 -0.148,28.42258
27.3012,0.13831 -0.2218,-28.63005 z">
    </a>
    <a href="javascript:void(0)">
        <title>Systems</title>
        <path id="13" class="part"
            d="m 1856.3752,191.01314 -0.148,28.42258
27.3012,0.13831 -0.2218,-28.63005 z">
    </a>
    <a href="javascript:void(0)">
        <title>Box</title>
        <path id="14" class="part"
            d="m 1692.8371,305.09509 -0.1875,21.37728
91.0409,0.18752 -0.1875,-21.47104 z">
    </a>
    <a href="javascript:void(0)">
        <title>Sphere</title>
        <path id="15" class="part"
            d="m 1693.0715,331.06636 -0.1875,21.37728
91.0409,0.18752 -0.1875,-21.47104 z">
    </a>
    <a href="javascript:void(0)">
        <title>Cylinder</title>

```







```

                d="m      864.05952,405.08558      -0.1956,29.24205
29.04645,0.0978 0.0978,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="32" class="part"
            d="m      895.74615,404.69438      -0.1956,29.24205
29.04645,0.0978 0.0978,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="33" class="part"
            d="m      927.62962,404.49878      -0.1956,29.24205
29.04646,0.0978 0.0978,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="34" class="part"
            d="m      959.70786,404.49878      -0.1956,29.24205
29.04646,0.0978 0.0978,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="35" class="part" d="m 991.88389,404.79217 -
0.1956,29.24205 29.04651,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="36" class="part" d="m 1023.5709,404.69438 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="37" class="part" d="m 1055.747,404.88998 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="38" class="part" d="m 1088.3388,404.48857 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="39" class="part" d="m 1120.5149,404.78196 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="40" class="part" d="m 1152.2019,404.68417 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">

```

```

</a>
<a href="javascript:void(0)">
  <path id="41" class="part" d="m 1184.378,404.87977 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
</a>
<a href="javascript:void(0)">
  <path id="42" class="part"
d="m 703.93191,437.23248 -0.1956,29.24205
29.04646,0.0978 0.0978,-29.24205 z">
</a>
<a href="javascript:void(0)">
  <path id="43" class="part"
d="m 735.81539,437.03688 -0.1956,29.24205
29.04646,0.0978 0.0978,-29.24205 z">
</a>
<a href="javascript:void(0)">
  <path id="44" class="part"
d="m 767.89363,437.03688 -0.1956,29.24205
29.04646,0.0978 0.0978,-29.24205 z">
</a>
<a href="javascript:void(0)">
  <path id="45" class="part"
d="m 800.06966,437.33027 -0.1956,29.24205
29.04645,0.0978 0.0978,-29.24205 z">
</a>
<a href="javascript:void(0)">
  <path id="46" class="part" d="m 831.7567,437.23248 -
0.1956,29.24205 29.04648,0.0978 0.098,-29.24205 z">
</a>
<a href="javascript:void(0)">
  <path id="47" class="part" d="m 863.93278,437.42808 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
</a>
<a href="javascript:void(0)">
  <path id="48" class="part" d="m 895.61938,437.03688 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
</a>
<a href="javascript:void(0)">
  <path id="49" class="part" d="m 927.50288,436.84128 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
</a>

```

```

    <a href="javascript:void(0)">
      <path id="50" class="part" d="m 959.58108,436.84128 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
      <path id="51" class="part" d="m 991.75708,437.13467 -
0.1956,29.24205 29.04652,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
      <path id="52" class="part" d="m 1023.4441,437.03688 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
      <path id="53" class="part" d="m 1055.6202,437.23248 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
      <path id="54" class="part" d="m 1088.212,436.83107 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
      <path id="55" class="part" d="m 1120.3881,437.12446 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
      <path id="56" class="part" d="m 1152.0751,437.02667 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
      <path id="57" class="part" d="m 1184.2512,437.22227 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
      <path id="58" class="part" d="m 703.9976,468.75816 -
0.1956,29.24205 29.04646,0.0978 0.0978,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
      <path id="59" class="part"
      d="m 735.88108,468.56256 -0.1956,29.24205
29.04646,0.0978 0.0978,-29.24205 z">
    </a>

```

```

<a href="javascript:void(0)">
  <path id="60" class="part"
        d="m      767.95932,468.56256      -0.1956,29.24205
29.04646,0.0978 0.0978,-29.24205 z">
  </a>
<a href="javascript:void(0)">
  <path id="61" class="part"
        d="m      800.13535,468.85595      -0.1956,29.24205
29.04644,0.0978 0.0978,-29.24205 z">
  </a>
<a href="javascript:void(0)">
  <path id="62" class="part" d="m 831.82238,468.75816 -
0.1956,29.24205 29.04648,0.0978 0.098,-29.24205 z">
  </a>
<a href="javascript:void(0)">
  <path id="63" class="part" d="m 863.99846,468.95376 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
  </a>
<a href="javascript:void(0)">
  <path id="64" class="part" d="m 895.68506,468.56256 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
  </a>
<a href="javascript:void(0)">
  <path id="65" class="part" d="m 927.56856,468.36696 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
  </a>
<a href="javascript:void(0)">
  <path id="66" class="part" d="m 959.64676,468.36696 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
  </a>
<a href="javascript:void(0)">
  <path id="67" class="part" d="m 991.82276,468.66035 -
0.1956,29.24205 29.04654,0.0978 0.098,-29.24205 z">
  </a>
<a href="javascript:void(0)">
  <path id="68" class="part" d="m 1023.5098,468.56256 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
  </a>
<a href="javascript:void(0)">
  <path id="69" class="part" d="m 1055.6859,468.75816 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">

```

```

</a>
<a href="javascript:void(0)">
    <path id="70" class="part" d="m 1088.2777,468.35675 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
</a>
<a href="javascript:void(0)">
    <path id="71" class="part" d="m 1120.4538,468.65014 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
</a>
<a href="javascript:void(0)">
    <path id="72" class="part" d="m 1152.1408,468.55235 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
</a>
<a href="javascript:void(0)">
    <path id="73" class="part" d="m 1184.3169,468.74795 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
</a>
<a href="javascript:void(0)">
    <path id="74" class="part" d="m 703.8998,500.44521 -
0.1956,29.24205 29.04646,0.0978 0.0978,-29.24205 z">
</a>
<a href="javascript:void(0)">
    <path id="75" class="part"
        d="m 735.78328,500.24961 -0.1956,29.24205
29.04646,0.0978 0.0978,-29.24205 z">
</a>
<a href="javascript:void(0)">
    <path id="76" class="part"
        d="m 767.86152,500.24961 -0.1956,29.24205
29.04646,0.0978 0.0978,-29.24205 z">
</a>
<a href="javascript:void(0)">
    <path id="77" class="part" d="m 800.03755,500.543 -
0.1956,29.24205 29.04644,0.0978 0.0978,-29.24205 z">
</a>
<a href="javascript:void(0)">
    <path id="78" class="part" d="m 831.72458,500.44521 -
0.1956,29.24205 29.04648,0.0978 0.098,-29.24205 z">
</a>
<a href="javascript:void(0)">

```

```

        <path id="79" class="part" d="m 863.90066,500.64081 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="80" class="part" d="m 895.58726,500.24961 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="81" class="part" d="m 927.47076,500.05401 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="82" class="part" d="m 959.54896,500.05401 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="83" class="part" d="m 991.72496,500.3474 -
0.1956,29.24205 29.04654,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="84" class="part" d="m 1023.412,500.24961 -
0.1956,29.24205 29.0466,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="85" class="part" d="m 1055.5882,500.44521 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="86" class="part" d="m 1088.18,500.0438 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="87" class="part" d="m 1120.3561,500.33719 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">
        <path id="88" class="part" d="m 1152.0431,500.2394 -
0.1956,29.24205 29.0465,0.0978 0.098,-29.24205 z">
    </a>
    <a href="javascript:void(0)">

```

```

        <path id="89" class="part" d="m 1184.2192,500.435 -
0.1956,29.24205 29.0464,0.0978 0.098,-29.24205 z">
        </a>
        <foreignObject class="step1-inputs" x="1792" y="665" width="85"
height="160">
                {{{STEP1_INPUT_LENGTH}}}
        </foreignObject>
        <foreignObject class="step1-inputs" x="1792" y="690" width="85"
height="160">
                {{{STEP1_INPUT_WIDTH}}}
        </foreignObject>
        <foreignObject class="step1-inputs" x="1792" y="715" width="85"
height="160">
                {{{STEP1_INPUT_HEIGHT}}}
        </foreignObject>
</svg>

</div>
<div id="chart" style="height: 20px;">{{{DIAG}}}</div>
<button style="margin-right: 75px;" class="btn btn-success pull-right
check">{{{CHECK}}}</button>
</div>

```

### Файл trainer1\_step1.js

```

var VStep1;
const imgPath = "img/trainer1-step1/";

var trainer1_step1 = function () {
  this.postDispatch = function () {
    $("navbar navbar-default").css("display", "none");
    $(".input").attr("autocomplete", "off");
    $("#endTrainer").removeClass("disabled btn-default");
    $(".part:gt(5)").hide();
    $(".part:gt(24)").css("opacity", "0");
    var options = {
      length: [50, 20, 15, 13, 25, 10, 55, 20, 10, 15],
      width: [30, 80, 23, 24, 35, 10, 44, 30, 15, 45],
      height: [25, 10, 60, 10, 45, 10, 62, 20, 90, 50],
      color: [75, 76, 77, 79, 80, 81, 82, 85, 86, 88],
    };
    var colorAreaId = options.color[userVariant - 1];

```

```

var order = [1, 7, 14, 25, colorAreaId];
var imgInfo = {
  path: [imgPath + "2.png", imgPath + "3.png", imgPath + "4.png", imgPath + "5.png",
imgPath + "6-" + userVariant + ".png"],
  order: order.slice(),
};
var partIdsBorders = {
  min: [6, 13, 24, 25],
  max: [13, 24, 25, 89],
};
setTableValues(options);
VStep1 = new Validator();
VStep1.addValidator($('input[name="step1-input-length"]'), options.length[userVariant
- 1]);
VStep1.addValidator($('input[name="step1-input-width"]'), options.width[userVariant -
1]);
VStep1.addValidator($('input[name="step1-input-height"]'), options.height[userVariant
- 1]);
VStep1.setStrictMode(true).enableStepFinishAlert(true).setIgnoreCase(false);
addSvgStep(order, imgInfo, partIdsBorders, VStep1, () => {
  $(".step1-inputs").css("visibility", "visible");
});
$("button.check").click(function () {
  const saveAttempts = VStep1.getAttempts();
  VStep1.setAttemptsOnCheckButton($(this));
  VStep1.validate();
  if (saveAttempts == VStep1.getAttempts() && VStep1.getAttempts() > 0) {
    $(".step1-inputs").css("visibility", "hidden");
    $("#mainImg").attr("src", "img/trainer1-step1/7-" + userVariant + ".png");
    $("button.check").off("click");
  }
});
};

this.mustache = function () {
  return {
    STEP1_INPUT_LENGTH: new TextInput("step1-input-length")
      .placeholder("")
      .render(),
    STEP1_INPUT_WIDTH: new TextInput("step1-input-width")
      .placeholder("")

```



```

        .render(),
    STEP1_INPUT_HEIGHT: new TextInput("step1-input-height")
        .placeholder("")
        .render(),
    };
};

function setTableValues(values) {
    $("#boxlength").html(values.length[userVariant - 1]);
    $("#boxwidth").html(values.width[userVariant - 1]);
    $("#boxheight").html(values.height[userVariant - 1]);
    $("#boxcolor").attr("src", "img/trainer1-step1/colors/" + userVariant + ".png");
    $("#variant").text($("#variant").text() + userVariant);
}
};

```

### Файл trainer2\_step3.html

```

<link rel="stylesheet" href="/trainer/css/steps.css" type="text/css" />
<script src="/trainer/scripts/helpers.js"></script>
<div id="step3" class="jumbotron page1">
    <div style="margin-left: 5px">
        <h3>{{STEP3_TITLE}}</h3>
        <p>{{STEP3_TASK}}</p>
    </div>
    <div class="map">
        <svg id="svgmap" viewBox="0 0 1920 1030">
            <a href="javascript:void(0)">
                <title>Edit</title>
                <path id="1" class="part"
                    d="M 50.964783,30.207353 51.259178,54.18359
107.20566,53.928523 107.05839,30.143586 Z">
            </a>
            <a href="javascript:void(0)">
                <title>Tools</title>
                <path id="2" class="part"
                    d="m 107.14729,30.145486 0.32435,23.978036 61.63863,-
0.255085 -0.16227,-23.786722 z">
            </a>
            <a href="javascript:void(0)">
                <title>Group</title>
                <path id="3" class="part"

```

```

                                d="m 169.2278,30.084518 0.33081,23.903171 62.86703,-
0.254289 -0.1655,-23.712455 z">
    </a>
    <a href="javascript:void(0)">
      <title>Views</title>
      <path id="4" class="part"
                                d="m 232.26014,30.020945 0.33081,23.903171 62.86703,-
0.254289 -0.1655,-23.712455 z">
    </a>
    <a href="javascript:void(0)">
      <title>Create</title>
      <path id="5" class="part"
                                d="m 295.32126,29.98556 0.36661,23.846649 69.66911,-
0.253688 -0.18341,-23.656384 z">
    </a>
    <a href="javascript:void(0)">
      <title>Modifiers</title>
      <path id="6" class="part"
                                d="m 365.25553,30.002309 0.45202,23.722503 85.90007,-
0.252367 -0.22614,-23.533228 z">
    </a>
    <a href="javascript:void(0)">
      <title>Animation</title>
      <path id="7" class="part"
                                d="m 451.41934,29.976015 0.50715,23.648715 96.37771,-
0.251582 -0.25372,-23.460029 z">
    </a>
    <a href="javascript:void(0)">
      <title>Graph Editors</title>
      <path id="8" class="part"
                                d="m 548.10532,29.966254 0.59265,23.542167 112.62574,-
0.250448 -0.29649,-23.354332 z">
    </a>
    <a href="javascript:void(0)">
      <path id="9" class="part"
                                d="m 98.512418,61.180452 0.18748,35.253765 35.628802,-
0.37504 -0.0938,-34.972485 z">
    </a>
    <a href="javascript:void(0)">
      <path id="10" class="part"

```

```

                                d="m 137.23065,60.915259 0.18748,35.253765 35.6288,-
0.37504 -0.0938,-34.972485 z">
    </a>
    <a href="javascript:void(0)">
        <path id="11" class="part"
                                d="m 178.07043,61.180453 0.18748,35.253765 35.6288,-
0.37504 -0.0938,-34.972485 z">
    </a>
    <a href="javascript:void(0)">
        <path id="12" class="part"
                                d="m 311.46269,61.445645 0.18748,35.253766 35.6288,-
0.37504 -0.0938,-34.972486 z">
    </a>
    <a href="javascript:void(0)">
        <path id="13" class="part"
                                d="m 352.35014,61.13153 0.18748,35.253766 35.6288,-0.37504
-0.0938,-34.972486 z">
    </a>
    <a href="javascript:void(0)">
        <path id="14" class="part"
                                d="m 391.52725,61.602703 0.18748,35.253767 35.6288,-
0.37504 -0.0938,-34.972487 z">
    </a>
    <a href="javascript:void(0)">
        <path id="15" class="part"
                                d="m 432.4147,61.288588 0.18748,35.253767 35.6288,-0.37504
-0.0938,-34.972487 z">
    </a>
    <a href="javascript:void(0)">
        <path id="16" class="part" d="m 68.083122,153.90986
0.0913,22.18712 H 465.2296 l -0.18262,-21.97408 z">
    </a>
    <a href="javascript:void(0)">
        <path id="17" class="part" d="m 68.174422,176.09698
0.0913,22.18712 H 465.32089 l -0.18262,-21.97408 z">
    </a>
    <a href="javascript:void(0)">
        <path id="18" class="part" d="m 68.357046,198.07106
0.0913,22.18712 H 465.50351 l -0.18262,-21.97408 z">
    </a>
    <a href="javascript:void(0)">

```

```

        <title>OK</title>
        <path id="19" class="part"
            d="m 374.47239,541.13452 2.19132,-2.73915 93.99626,-0.074
2.96987,2.53945 0.12913,22.08035 -2.84075,2.7977 -93.46485,0.0215 -3.24964,-2.19512 z">
        </a>
        <a href="javascript:void(0)">
            <title>Group...</title>
            <path id="20" class="part" d="M 165.96844,55.093257 H 374.63415 V
80.573888 L 165.6241,80.918221 Z">
        </a>
        <a href="javascript:void(0)">
            <title>Assembly</title>
            <path id="21" class="part" d="m 166.4883,278.73703 h 208.66571 v
25.48063 l -209.01005,0.34433 z">
        </a>
        <a href="javascript:void(0)">
            <title>OK</title>
            <path id="22" class="part"
                d="m 931.69327,580.65578 -0.0938,-21.84609 2.53152,-
2.43776 94.13511,-0.0938 2.4377,3.00032 0.1876,21.56481 -2.6253,3.00032 -93.94759,0.18752
z">
        </a>
    </svg>
    
</div>
<div id="chart" style="height: 20px;">{{{DIAG}}}</div>
<button style="margin-right: 75px;" class="btn btn-success pull-right
check">{{{CHECK}}}</button>
</div>
function setTableValues(values) {
    $("#boxlength").html(values.length[userVariant - 1]);
    $("#boxwidth").html(values.width[userVariant - 1]);
    $("#boxheight").html(values.height[userVariant - 1]);
    $("#boxcolor").attr("src", "img/trainer1-step1/colors/" + userVariant + ".png");
    $("#variant").text($("#variant").text() + userVariant);
}
};

```

### Файл trainer2\_step3.js

```

var VStep3;
const imgPath = "img/trainer2-step3/";

```

```

var trainer2_step3 = function () {
  this.preDispatch = function () {
    $("#step2").empty();
  };
  this.postDispatch = function () {
    $("button.check").css("display", "none");
    $(".part:gt(14)").hide();
    var order = [13, 16, 17, 18, 19, 3, 20, 22];
    var imgInfo = {
      path: getImgPathArray(),
      order: order.slice(),
    };
    var partIdsBorders = {
      min: [15, 16, 17, 18, 0, 19, 21],
      max: [19, 19, 19, 19, 15, 21, 22],
    };
    VStep3 = new Validator();
    VStep3.setStrictMode(true).enableStepFinishAlert(true).setIgnoreCase(false);
    addSvgStep(order, imgInfo, partIdsBorders, VStep3, () => {
      $("div.validation-alert-success").fadeIn();
      Scorer.addScore(25);
      Rotator.enableNextButton();
    });
  };

  this.mustache = function () {
    return {};
  };

  function getImgPathArray() {
    var array = [];
    for (var i = 2; i <= 9; i++) {
      array[i - 2] = imgPath + i.toString() + ".png";
    }
    return array;
  }
};

```

**ДОДАТОК Г**  
**АКТ ВПРОВАДЖЕННЯ**