

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: «Інтерактивний додаток для вивчення текстування
об'єктів з курсу "Комп'ютерна графіка" »

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

Виконавець роботи: студент групи ІТ-62 Перепелиця Роман Олександрович

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**

_____ «__» _____ 2020 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-2020

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
«__»_____ 2020 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Перепелиця Роман Олександровича

1 Тема роботи *Інтерактивний додаток для текстурування об'єктів з предмету «Комп'ютерна графіка»*

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

затверджені наказом по університету від «14» травня 2020 р. № 0576-III

2 Строк подання студентом роботи «1» червня 2020 р.

3 Вхідні дані до роботи *технічне завдання для створення інтерактивного додатку та лекційний матеріал по темі текстурування об'єктів.*

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) *аналіз предметної області, проектування та розробка інтерактивного додатку.*

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) *актуальність роботи, аналіз програмних продуктів-аналогів, мета та задача дипломного проекту, аналіз технологій, етапи розробки та створення інтерактивного додатку.*

6. Консультанти розділів роботи:

| Розділ | Консультант | Підпис, дата | |
|--------|-------------|----------------|------------------|
| | | Завдання видав | Завдання прийняв |
| | | | |

7.Дата видачі завдання _____ 01.10.2019 _____

КАЛЕНДАРНИЙ ПЛАН

| № п/п | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|-------|-------------------------------------|-------------------------------|----------|
| 1 | Огляд останніх досліджень | 10.11.2019- 15.12.2019 | |
| 2 | Аналіз аналогів | 15.12.2019- 30.12.2020 | |
| 3 | Розробка каркасу | 01.01.2020- 19.03.2020 | |
| 4 | Створення скриптів | 20.03.2020- 04.04.2020 | |
| 5 | Розробка дизайну | 05.04.2020- 10.04.2020 | |
| 6 | Наповнення контентом | 10.04.2020- 15.04.2020 | |
| 7 | Оформлення документації | 15.04.2020- 31.05.2020 | |

Студент _____
(підпис)

Перепелиця Р. О.

Керівник роботи _____
(підпис)

к.т.н., доц. Баранова І.В.

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Інтерактивний додаток для вивчення текстуровання об'єктів з курсу "Комп'ютерна графіка"».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 17 найменувань, додатків. Загальний обсяг роботи – 82 сторінки, у тому числі 36 сторінок основного тексту, 2 сторінки списку використаних джерел, 43 сторінки додатків.

Кваліфікаційну роботу бакалавра присвячено створенню інтерактивного додатку для вивчення текстуровання об'єктів в комп'ютерній графіці для дистанційного навчання Сумського державного університету.

В роботі проаналізовано існуючі альтернативні проекти для практичного вивчення комп'ютерної графіки та розроблено вимоги до майбутнього проекту.

У розділі моделювання наведено структурно-функціональне моделювання роботи додатку та діаграма варіантів використання.

Представлено поетапно практичну реалізацію проекту: створено структуру тренажеру, візуалізовано графічний інтерфейс та створені функції додатку, також представлено кінцевий результат проекту.

Результатом є інтерактивний додаток для дистанційного навчання Сумського державного університету.

Ключові слова: ІНТЕРАКТИВНИЙ ДОДАТОК, ТЕКСТУРУВАННЯ ОБ'ЄКТІВ, КОМП'ЮТЕРНА ГРАФІКА, ДИСТАНЦІЙНЕ НАВЧАННЯ.

ЗМІСТ

| | |
|--|----|
| Вступ..... | 6 |
| 1 Аналіз предметної області | 8 |
| 1.1 Огляд останніх досліджень і публікацій | 8 |
| 1.2 Аналіз програмних продуктів – аналогів | 10 |
| 1.3 Постановка задачі | 13 |
| 2 Моделювання роботи інтерактивного додатку..... | 15 |
| 2.1 Побудова контекстної діаграми IDEF0 | 15 |
| 2.2 UML моделювання інтерактивного додатку | 16 |
| 3 Розробка інтерактивного веб-додатку..... | 20 |
| 3.1 Структура програмного додатку..... | 20 |
| 3.2 Програмна реалізація додатку | 21 |
| 3.3 Використання програмного додатку | 29 |
| Висновки..... | 36 |
| Список використаних джерел..... | 37 |
| Додаток А. Технічне завдання..... | 39 |
| Додаток Б. Планування робіт | 45 |
| Додаток В. Код додатку | 55 |

ВСТУП

Всі технології направлені на поліпшення життя зручність та безпеку усіх сфер життя. При цьому знання в сфері технології цінувалися та будуть тільки більше цінуватися при працевлаштуванні. Саме тому технології зробили свій виток до дистанційного навчання – більше не потрібно їхати в інше місто, іншу країну для того щоб отримати знання та досвід.

Майже все можна отримати дистанційно, матеріали, лекції, практичні завдання, тощо. І саме зараз, у тяжкий час, коли хвороба змусила людей піти на карантин, мінімізувати особистий контакт між людьми, дистанційне навчання розкриває свої зручності та переваги на відміну від традиційного навчання.

На жаль, не всі системи користувачів відповідають сучасним вимогам до програмних продуктів. Комп'ютерна графіка в наш час зробила дуже великий крок, розширивши свій вигляд та можливості, і разом с цим потребує все більше і більше затрат ресурсів комп'ютерів користувачів та розробників.

Також сучасні програмні продукти дуже часто потребують великих часових затрат для встановлення на машину та завантаження усіх потрібних бібліотек.

Метою даного проекту є створення інтерактивного додатку для системи дистанційної освіти Сумського державного університету. Додаток допоможе студентам виконувати практичні завдання з дисципліни «Комп'ютерна графіка», не встановлюючи ніякого програмного продукту, та одразу отримати оцінку результату за виконання практичного завдання.

Для досягнення мети необхідно під час практики вирішити такі задачі:

- дослідити актуальність питання розробки та використання інтерактивних тренажерів та веб-додатків;
- обрати засоби та методи реалізації поставленої задачі;

- провести планування робіт;
- розробити структуру додатку, інтуїтивно зрозумілий веб-інтерфейс;
- створити контент додатку;
- реалізувати функції роботи веб-тренажеру.

З використанням розробленого тренажера викладач зможе якісніше контролювати роботу студентів та вчасно допомагати їм в проблемних питаннях.

Проект спростить співпрацю між студентом та викладачем, а також пришвидшить набуття студентом вмінь та навичок роботи з комп'ютерною графікою.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Технології дистанційного навчання дуже швидко розвиваються і постійно оновлюються вбираючи до себе все зручніші та продуктивніші розробки та додатки.

В двадцять першому столітті ефективною системою підготовки фахівців високого кваліфікаційного рівня стало дистанційне навчання [1].

Закордонні експерти вважають, що альтернативна освіта стала порятунком у сучасному світі для отримання знань в цілому. Через величезну кількість населення, а також через величезний запит у знаннях, навіть самі заможні країни не можуть витримувати фінансові навантаження при денній формі навчання. Саме тому навчання по альтернативним технологіям з кожним роком тільки збільшується відповідно навчанню за денною формою.

В майбутньому дистанційна освіта має намір надавати людині знання в будь-якій частині світу. Такий процес дуже спростить навчання, адже виключить фізичне переміщення через країни та міста. Натомість надасть зручні та мобільні ідеї для користування знаннями світу.

Якщо порівняти традиційну та дистанційну форми навчання – то у останньої виділяють певні особливості [2], такі як:

- гнучкість – студенти альтернативної форми навчання працюють у зручному темпі й у зручний для себе час;
- паралельність – студенти можуть поєднувати навчання з іншою діяльністю;

- віддаленість не є перешкодою для навчання;
- масовість – кількість дистанційних студентів форми не має обмеження;
- використання нових сучасних технологій та систем телекомунікації.

При користуванні дистанційним навчанням важливе значення надається розробці та програмному виконанні інтерактивних тренажерів [3]. Такий засіб навчання дасть змогу підвищити якість та розуміння матеріалу студентами, виробити практичні та теоретичні навички під час вивчення матеріалу.

Інтерактивний тренажер дає можливість реалізувати теорію та практичний матеріал в різних формах подання, надавати супроводжуючі підказки, надавати графічні анімації та звуковий супровід. Також такий спосіб подання допомагає контролювати отримання знань та допомагає отримувати знання на рівних рівнях засвоєння матеріалу [3].

Використовуючи такі інтерактивні додатки можна з легкістю розібратися в складних програмах та швидше опанувати інформацію, використовуючи різні навчальні матеріали: бібліотеки, каталоги, тощо.

За допомогою тренажеру можна враховувати:

- рівень підготовки;
- рівень вмотивованості;
- індивідуальні особливості;
- нормування обсягу.

Хоча технології зробили величезний ривок в розвитку, на даний момент не було створено інтерактивного додатку для навчання з дисципліни «Комп'ютерна графіка», і саме цей крок є дуже важливим у спрощенні та автоматизації навчання в дистанційному режимі.

Головною метою є пришвидшення та спрощення навчання студентів, та зменшення навантаження на систему користувача [4, 5]. Водночас додаток дає

змогу викладачу контролювати процес навчання студента та швидко виправляти помилки студента корегуючи та направляючи його.

Навчальний інтерактивний додаток має змогу надавати наступні можливості:

- виконання практичного завдання онлайн, без задіяння потужних ресурсів системи;
- зручний спосіб контролю між вчителем і студентом.

Таким чином, сучасний інтерактивний додаток зі стильним та зручним оформленням дозволяє спростити навчання студентам та поліпшити роботу викладачу під час перевірки завдань.

1.2 Аналіз програмних продуктів – аналогів

Створення інтерактивного додатку обов'язково повинно супроводжуватися чіткою постановкою завдання та аналізом вже існуючих додатків для вирішення проблеми. Аналіз вже готових рішень від інших розробників, аналіз цільової аудиторії та виявлення помилок інших продуктів є дуже важливим кроком при створенні власного додатку.

Під час пошуку аналогічних програм виявлено, що у відкритому доступі додатків, що відповідають темі роботи – текстурування в комп'ютерній графіці – знайдено не було. Тому аналіз було проведено для декількох інтерактивних додатків, що спеціалізуються в інших областях.

Деякі додатки частково відповідають меті навчити програмуванню на OpenGL, наприклад, веб-додаток у вигляді тесту для перевірки знань [6].

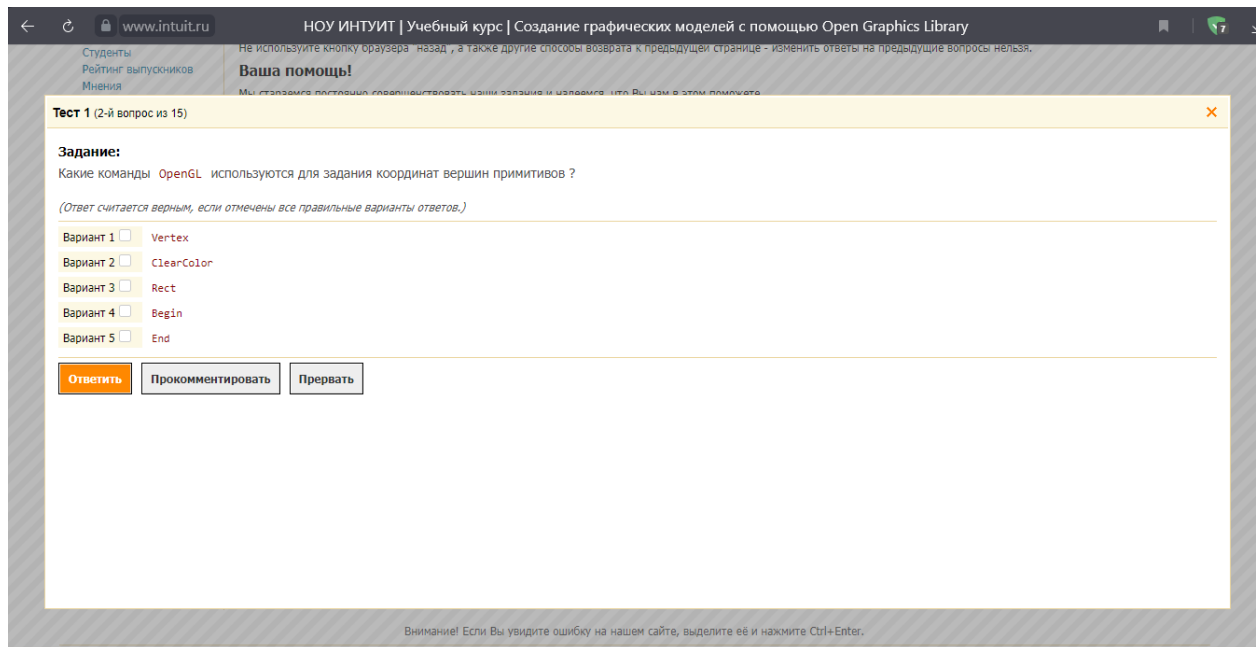


Рисунок 1.1 – Сторінка сайту intuit.ru

Цей додаток не відповідає поставленій задачі, оскільки він не дозволяє закріпити набуття практичних навичок з предметної області, лише перевіряє знання теоретичних матеріалів.

Інший приклад – сайт для вивчення рушія, заснованого на технології WebGL та відповідно, використання мови програмування Java Script kickjs.org [7].

Дуже зручно, що в такому додатку можна вводити команди та одразу бачити результат її виконання (рис. 1.2). На жаль, цей приклад також не відповідає темі роботи – вивчення саме теми текстурування об'єктів за допомогою бібліотеки OpenGL.

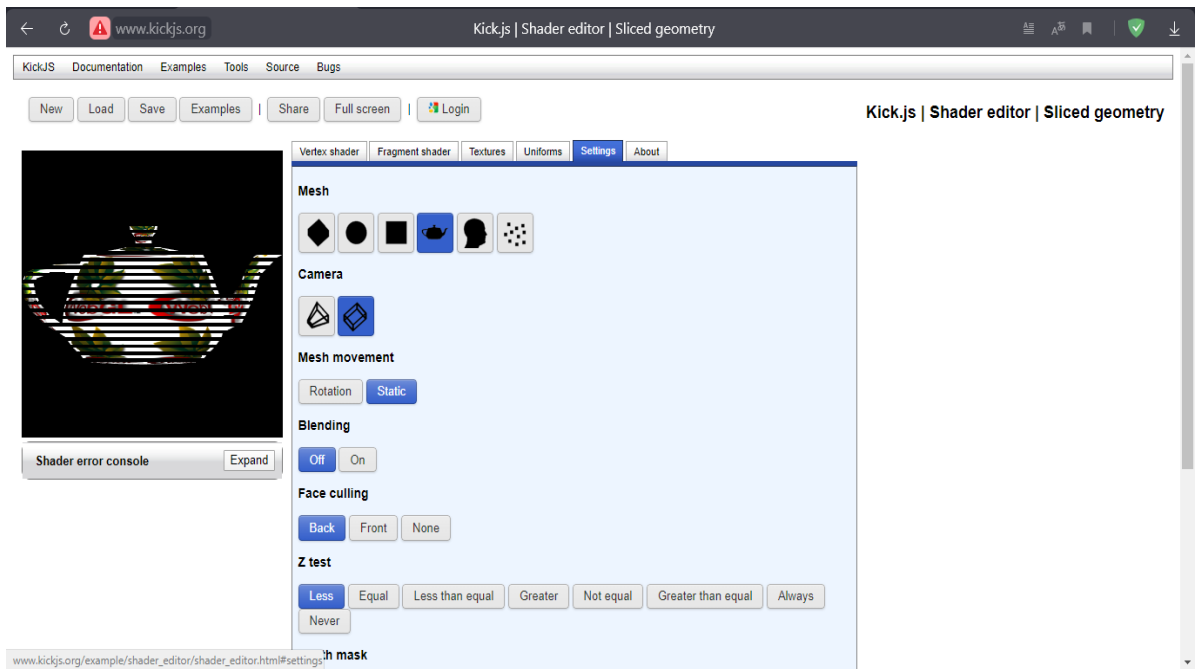


Рисунок 1.2 – Сторінка сайту kickjs.org

Проаналізувавши знайдені веб-додатки, було визначено їх недоліки та невідповідності до мети нашого інтерактивного додатку. Сформульовані переваги та недоліки проаналізованих програмних додатків наведено у таблиці 1.1.

Таблиця 1.1 – Аналіз розглянутих аналогів

| Критерії | kickjs.org | intuit.ru |
|------------------------------------|------------|-----------|
| Невідповідність поставленій задачі | + | + |
| Невідповідний контент | + | - |
| Можливість зворотного зв'язку | - | + |
| Вільний доступ до контенту | + | - |
| Зручна панель навігації | - | - |
| Вдалий підбір кольорів | - | - |
| Багатомовність | - | + |
| Оцінювання результату | - | + |

Недоліки:

- невідповідність поставленій задачі;
- невідповідний контент;
- відсутність вільного доступу;

До переваг аналогів можна віднести:

- оцінювання;
- відсутність реклами.

Таким чином, висновок про необхідність створення власного продукту.

1.3 Постановка задачі

Мета даної роботи полягає у створенні інтерактивного додатку по курсу «Комп'ютерна графіка» для сайту дистанційного навчання Сумського державного університету.

Для досягнення мети необхідно вирішити такі задачі:

- Розглянути сучасний стан питання розробки і підтримки інтерактивних тренажерів та їх використання у дистанційній освіті;
- Проаналізувати існуючі аналогічні продукти, визначити їх переваги та недоліки;
- Сформулювати постановку задачі, функціональні вимоги до проєкту;
- Обрати засоби та методи реалізації поставленої задачі;
- Провести моделювання роботи додатку з використанням методології IDEF0 та UML;
- Розробити структуру додатку, необхідні компоненти і модулі;
- Розробити інтерфейс додатку;

- Розробити та наповнити контент додатку – теоретичний матеріал, питання для тренажеру;
- Реалізувати механізм випадкового порядку виведення відповідей у завданні;
- Реалізувати механізм обчислення результату виконання завдань тренажеру;
- Перевірити працездатність інтерактивного додатку.

Тобто необхідно створити веб-додаток і роботи OpenGL в браузері.

Додаток повинен виконувати такі функції:

- вивід теоретичної частини,
- вивід завдань практичної частини;
- відображення підказок по роботі;
- виконання користувачем завдань тренажеру;
- вивід результатів для викладача.

Повністю вимоги до проекту наведені у технічному завданні (додаток А).

При виборі засобів реалізації завдання було обрано стандартизовану мову розмітки HTML, яка надає змогу розмістити елементів додатку на сторінці при цьому інформація та спосіб її представлення залежить від розробника. [8].

Для зміни зовнішнього вигляду об'єктів обрано набір параметрів форматування CSS [9].

При розробці проекту використано JavaScript - динамічну мову програмування, яка підтримує прототип створення об'єктів, може функціонувати як процедурна, так і об'єктно орієнтовна мова, також надає динамічні можливості для компіляції [10].

Проведене планування робіт над проектом, розроблений календарний план виконання проекту та оцінені можливі ризики. Повна інформація по плануванню наведена у додатку Б.

2 МОДЕЛЮВАННЯ РОБОТИ ІНТЕРАКТИВНОГО ДОДАТКУ

2.1 Побудова контекстної діаграми IDEF0

Контекстна діаграма дозволяє коротко і швидко описати призначення і межі системи [11], виявити і усунути колективні розбіжності в їх розумінні, показати і домовитися про її масштаб. Контекстна діаграма для інтерактивного додатку представлена на рисунку 2.1.

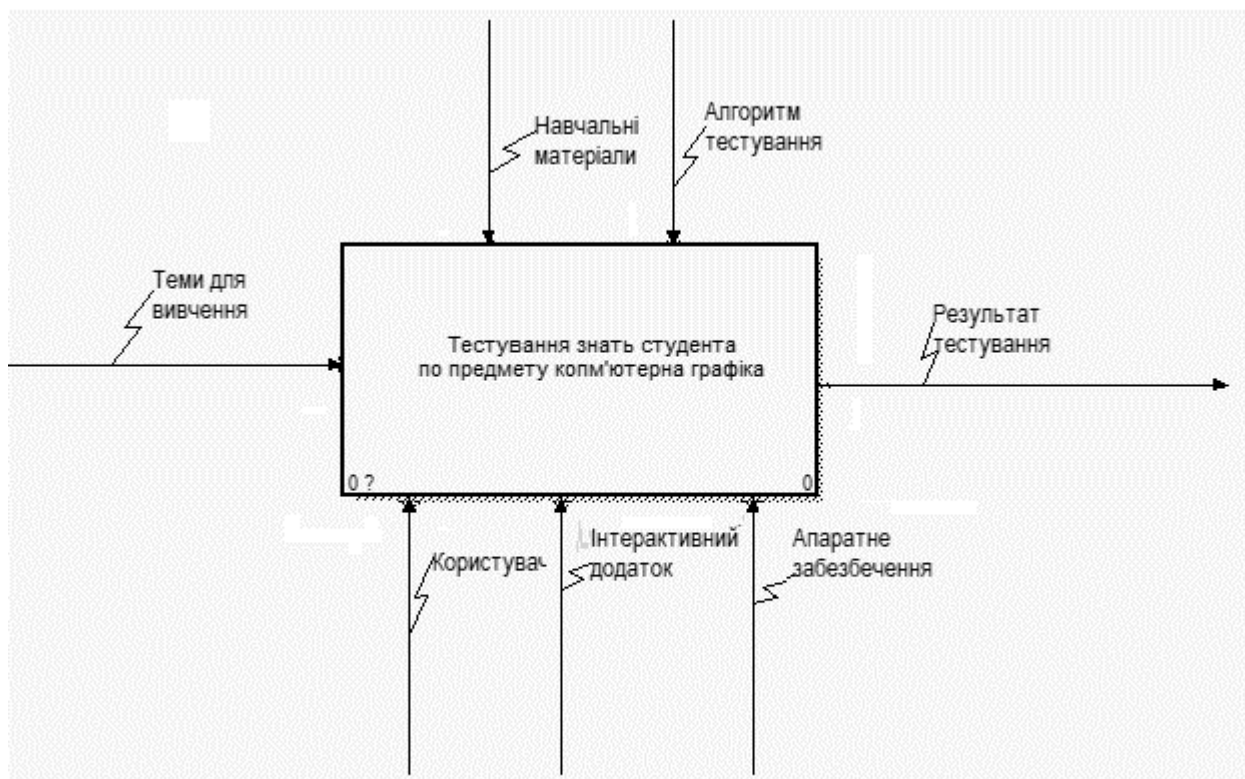


Рисунок 2.1 – Контекстна діаграма моделювання роботи веб-додатку для курсу «Комп'ютерна графіка»

Діаграма декомпозиції першого рівня дозволяє показати, із яких дрібних дій складається робота інтерактивного додатку (рис.2.2). В даній діаграмі були виділені наступні дії:

- Вивчення теоретичного матеріалу – ознайомлення з теоретичним матеріалом для можливості подальшого проходження практичного завдання по заданій темі;
- Проходження практичного завдання – виконання практичних завдань.

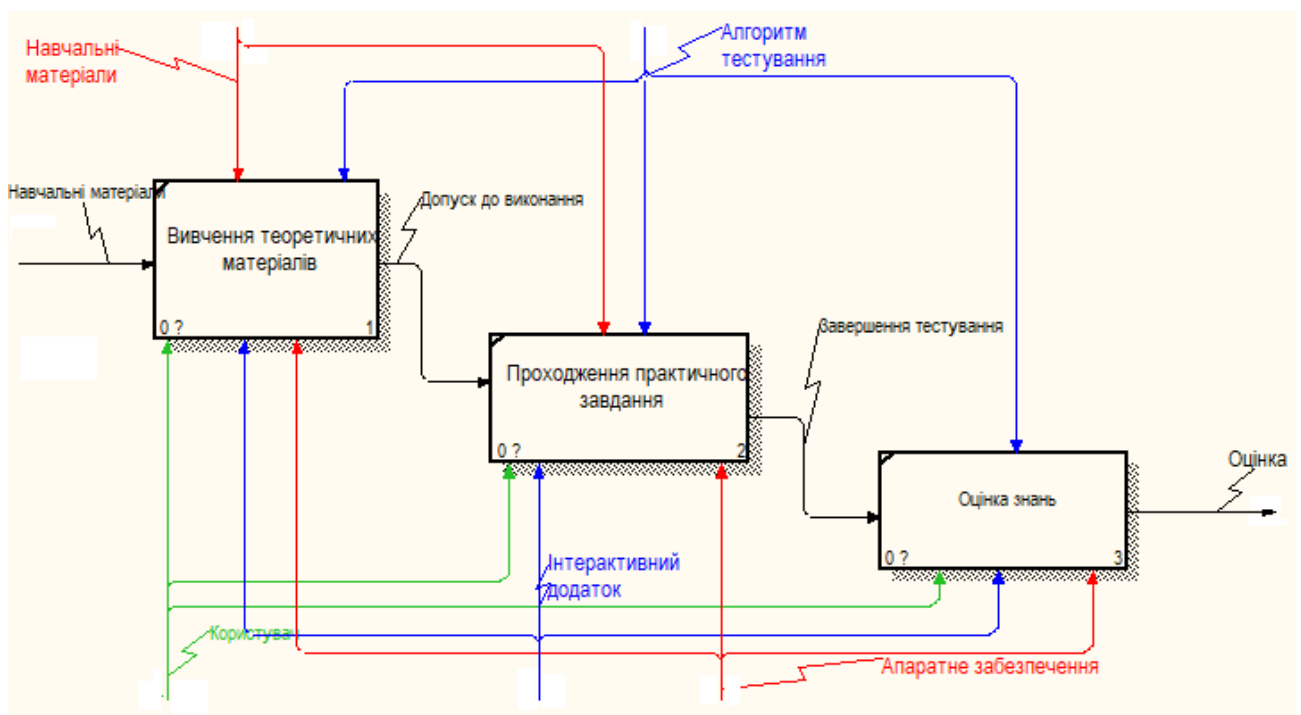


Рисунок 2.2 - Діаграма декомпозиції

2.2 UML моделювання інтерактивного додатку

Діаграма варіантів використання [12] створюється для демонстрації функцій, які доступні актору при використанні програмного продукту.

Основним актором є користувач, він же студент.

Актор має змогу:

- переглядати теоретичну частину;
- переглядати демонстраційного прикладу;
- проходити практичну частину завдання;
- отримати результат;

Іншим актором є сайт дистанційного навчання:

- отримати результат;

Також актором є сторонні бібліотеки та bootstrap, які надають свої функції для користування.

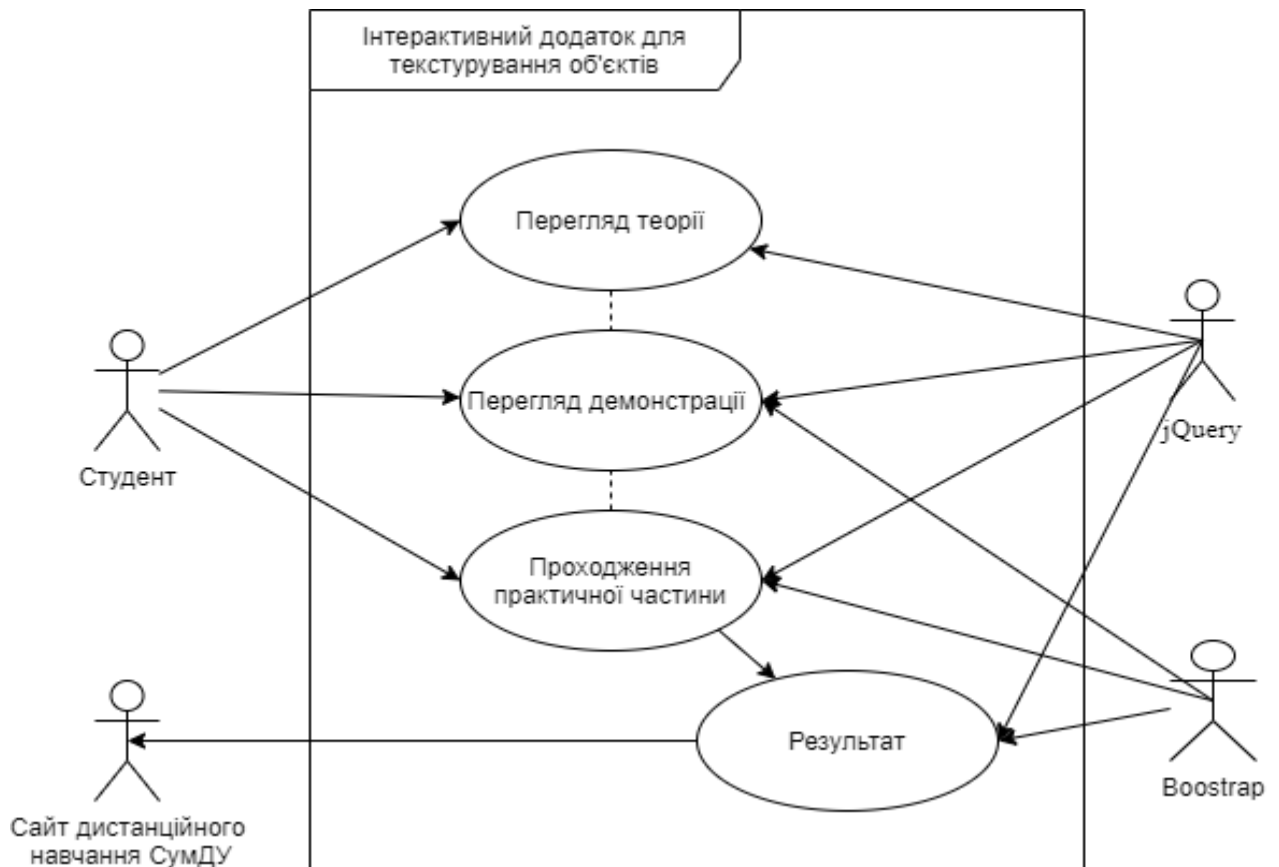


Рисунок 2.3 - Діаграма варіантів використання

Демонструючи компоненти, які формують систему, було створено діаграму компонентів [13]. Діаграма демонструє зв'язок компонентів системи між собою для роботи інтерактивного додатку (рис.2.4).

Додаток зорганізується за допомогою каскадних стилів, які знаходяться у файлі `title.css`.

Початковим є файл `index.html`, який відкриває головну сторінку додатку. Наступним йде файл `Theory.html`, який надає теоретичний матеріал по темі.

Після ознайомлення з теорією та переходом до наступної вкладки відкривається файл `S1.html`, в якому показано виконання програмної частини додатку за допомогою імітації програми Visual Studio [14]. Для функціонування були використані наступні файли: `panel.js` для скриптового функціонування допоміжної панелі, а також імітації компілятора, та бібліотеку JQuery [15].

Далі відкривається файл `Theory2`, який надає теоретичний матеріал до наступної частини, та після йдуть файли `S2` і `S3`, які повторюють функції файлу `S1`, але зі своїм контентом.

Після ознайомлення та переходу до наступної частини користувачу відкривається файл `Practick.html`. Даний файл для свого функціонування використовує файли `js.js`, в якому знаходиться набір адаптованих скриптів, та файл `data.js`, в якому знаходяться запитання та відповіді на запитання. Також файл використовує бібліотеку bootstrap [16] та віддалені файли `bootstrap.css` та `bootstrap.js`, які слугують для покращення зовнішнього вигляду та функціонування додатку.

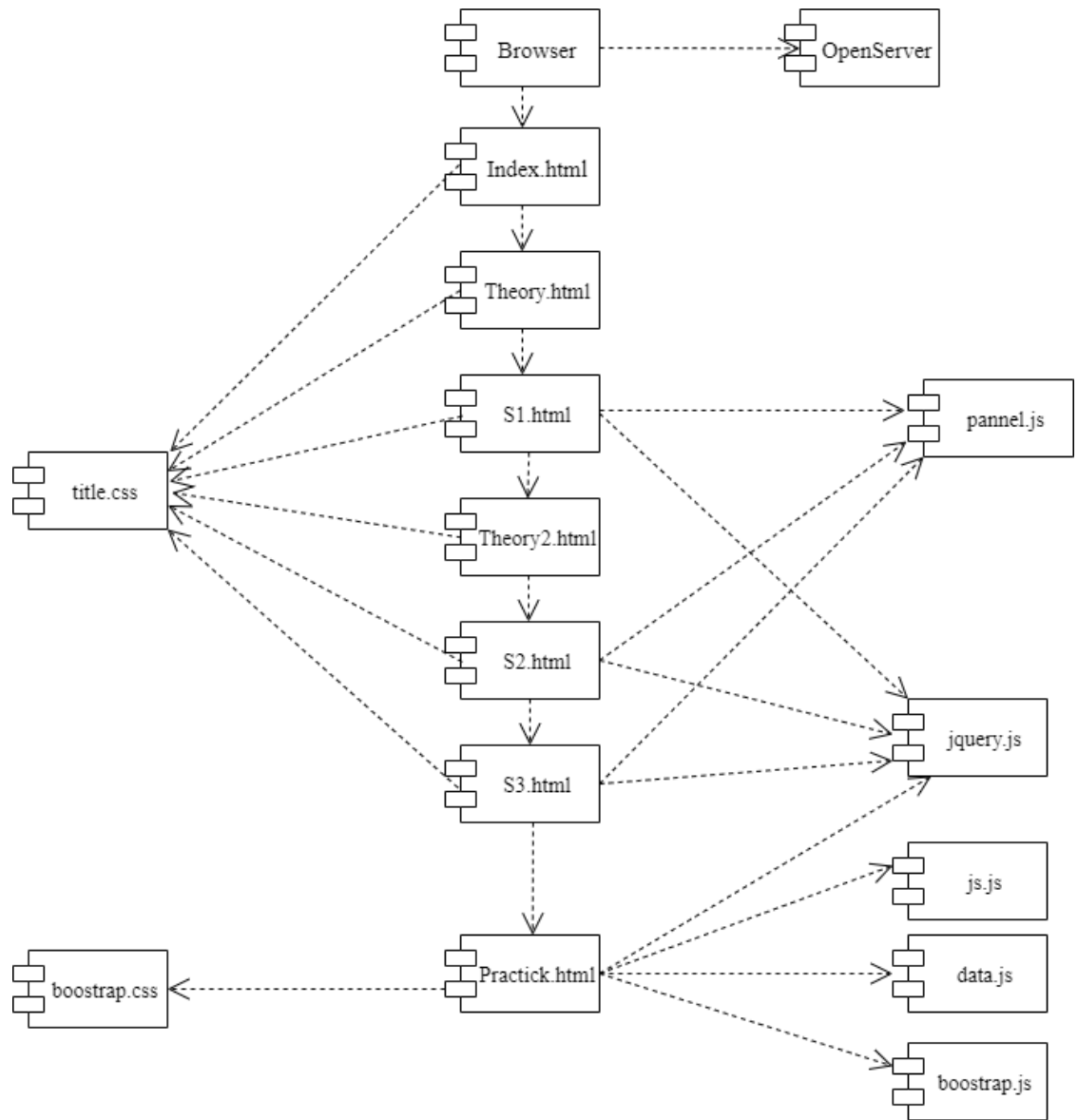


Рисунок 2.4 - Діаграма компонентів інтерактивного додатку

3 РОЗРОБКА ІНТЕРАКТИВНОГО ВЕБ-ДОДАТКУ

3.1 Структура програмного додатку

Інтерактивний додаток містить такі основні частини:

- Головна сторінка – містить інформацію про інтерактивний додаток, а також клавіші керування переходом до наступного етапу.
- Теоретичні відомості – містить теоретичний матеріал по темі текстування об'єктів.
- Практична візуалізація – демонструється виконання коду та результат.
- Практичне завдання – надаються питання та область для відповіді.
- Результат – виводить на екран неправильні відповіді користувача та результат проходження завдання.

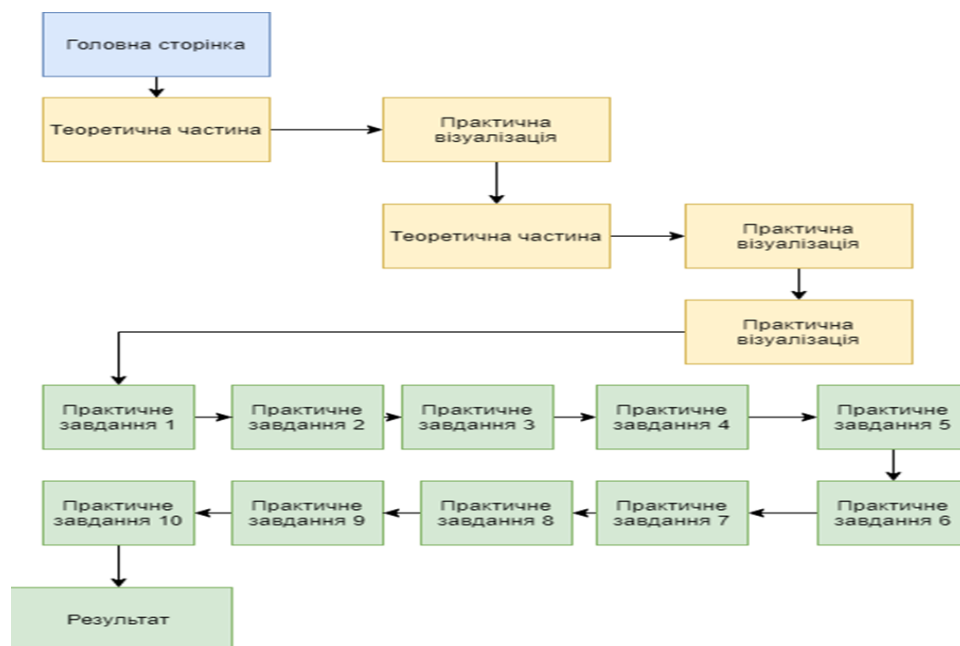


Рисунок 3.1 – Структура інтерактивного додатку

3.2 Програмна реалізація додатку

Для розробки додатку було створено чотири макети використовуючи стандартизовану мову розмітки HTML [8]. Після верстки макетів та розміщення початкових елементів було розроблено візуальну частину додатку за допомогою параметрів форматування CSS [9].

Створені макети можна поділити на такі підгрупи:

1. Титульна сторінка.

На титульній сторінці було створено посилання для контакту з викладачем, а також розміщено інформацію яку надає інтерактивний додаток, і розроблено кнопку для переходу до наступного розділу. На рисунку 3.2 продемонстровано титульну сторінку додатку.

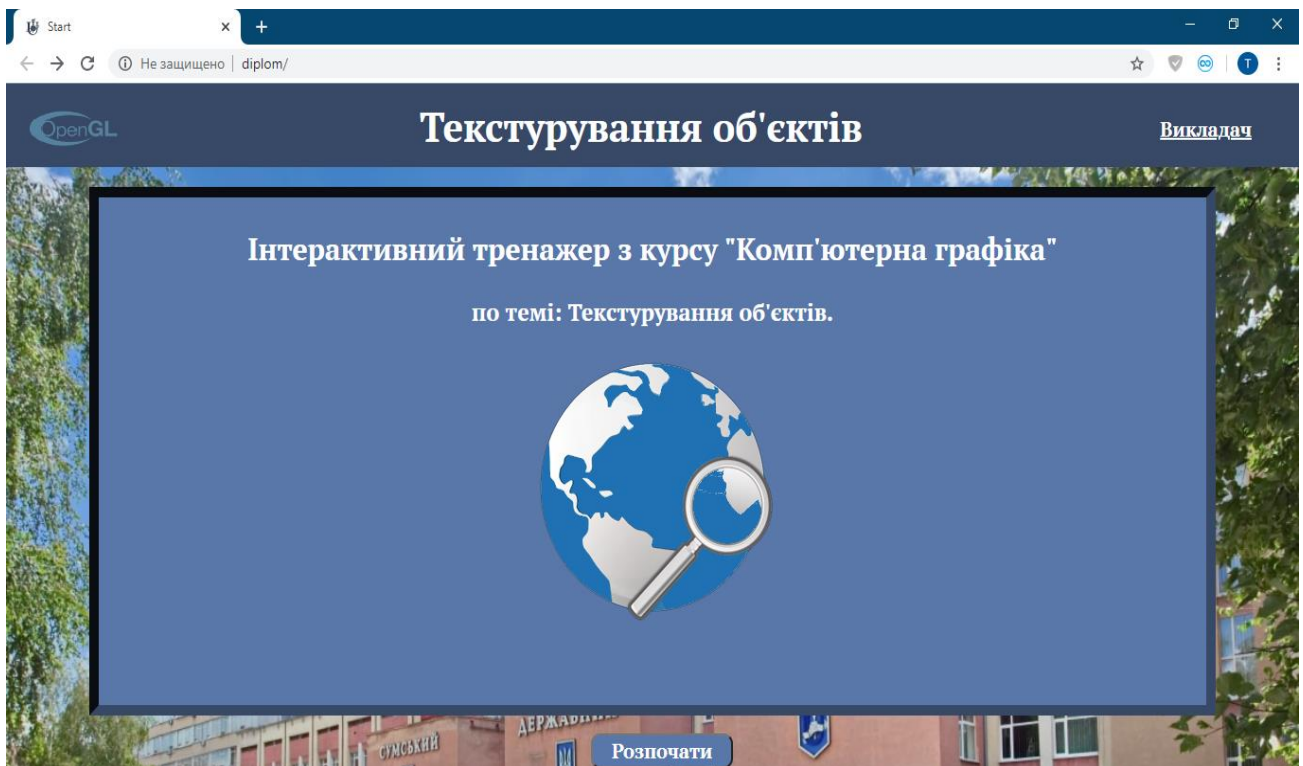


Рисунок 3.2– Титульна сторінка

Також одразу було розроблено візуальний вигляд кнопки за допомогою безкоштовної бібліотеки.

2. Теоретична частина.

Данна частина додатку створена для подання теоретичного матеріалу, для зручності була створена таблиця та зручно розміщений теоретичний матеріал для ознайомлення. На рисунку 3.3 демонструються дві розроблені таблиці для зручного подання матеріалу.

| Значення змінної format | Формат пікселів |
|-------------------------|---------------------------|
| GL_RGB | (R, G, B) |
| GL_RGBA | (R, G, B, A) |
| GL_RED | Тільки червона компонента |
| GL_GREEN | Тільки зелена компонента |
| GL_BLUE | Тільки синя компонента |
| GL_LUMINANCE | Яскравість |

| Значення змінної type | Тип даних |
|-----------------------|--|
| GL_UNSIGNED_BYTE | беззнакове 8-bit ціле (unsigned char або GLubyte) |
| GL_BYTE | знакове 8-bit ціле (char або GLbyte) |
| GL_BYTE | знакове 8-bit ціле (char або GLbyte) |
| GL_BITMAP | беззнакове 8-bit ціле (unsigned char або GLubyte) в тому ж форматі, що і бітовий масив |
| GL_UNSIGNED_SHORT | беззнакове 16-bit ціле (unsigned short або GLushort) |
| GL_SHORT | знакове 16-bit ціле (short або GLshort) |
| GL_UNSIGNED_INT | беззнакове 32-bit ціле (unsigned int або GLuint) |
| GL_INT | знакове 32-bit ціле (int або GLint) |
| GL_FLOAT | з плаваючою точкою (float або GLfloat) |

Останній аргумент в цій функції є покажчиком на нульовий елемент відображуваного масиву.
 Функція glPixelStore приймає в якості першого аргументу константу, яка визначає настроюваний параметр, а наступний аргумент визначає її значення.

[Наступний крок](#)

Рисунок 3.3 – Структура інтерактивного додатку.

3. Демонстрація програмного виконання.

Дана частина додатку повинна демонструвати студенту процес компіляції коду та допомагати в розумінні реалізації даного коду.

Для зручності в даній частині за допомогою мови програмування JavaScript [10], та безкоштовної бібліотеки функцій JQuery [15] було розроблено

висувний слайд для допоміжної інформації по практичній частині, а саме програмного коду (рис 3.4).

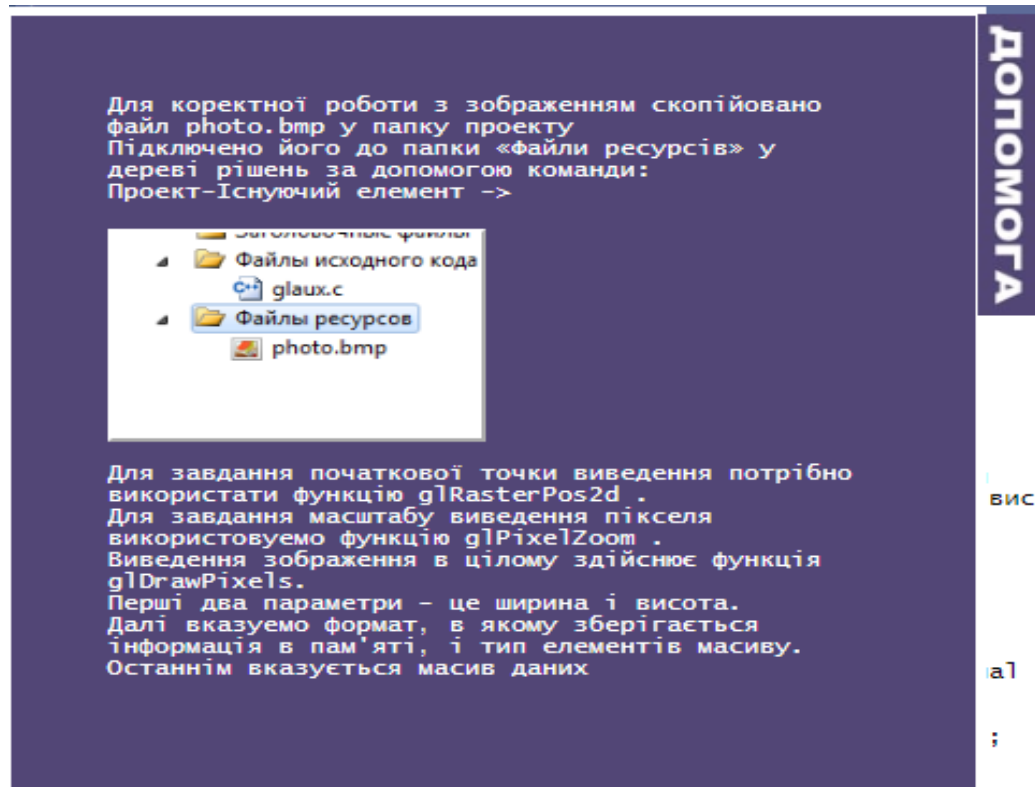


Рисунок 3.4 – Висувний слайд з допоміжною інформацією

Також було створено кнопку, яка імітує процес компіляції програмного продукту Microsoft Visual Studio [14]. Visual Studio – це середовище для розробки, налагодження та збірки коду та додатків, використовується для розробки програмних додатків [14].

Після натискання даної клавіші відкривається вікно з результатом виконання даного коду (рис. 3.5-3.6).

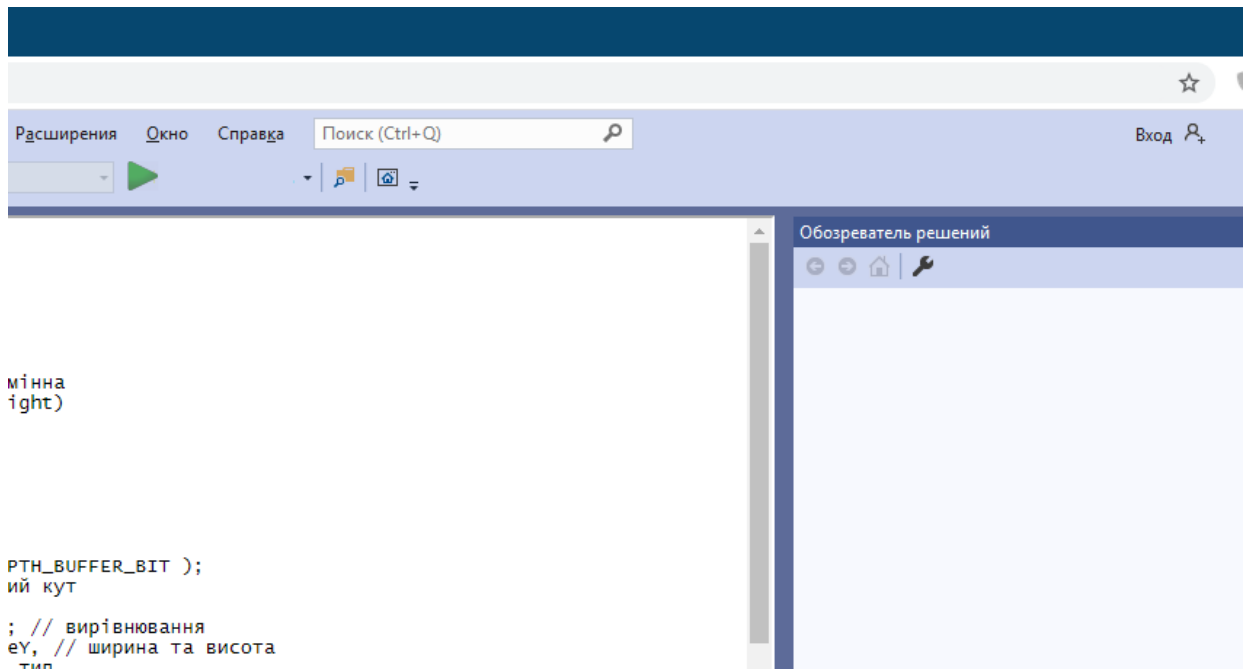


Рисунок 3.5 – Неактивна клавіша компіляції

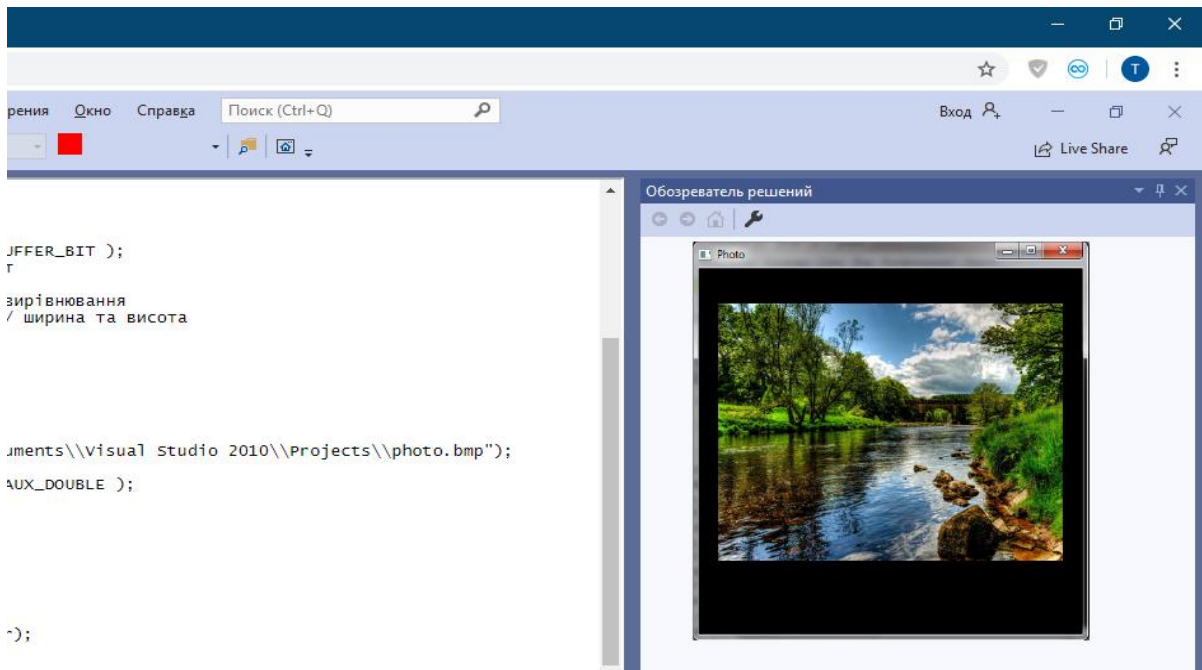
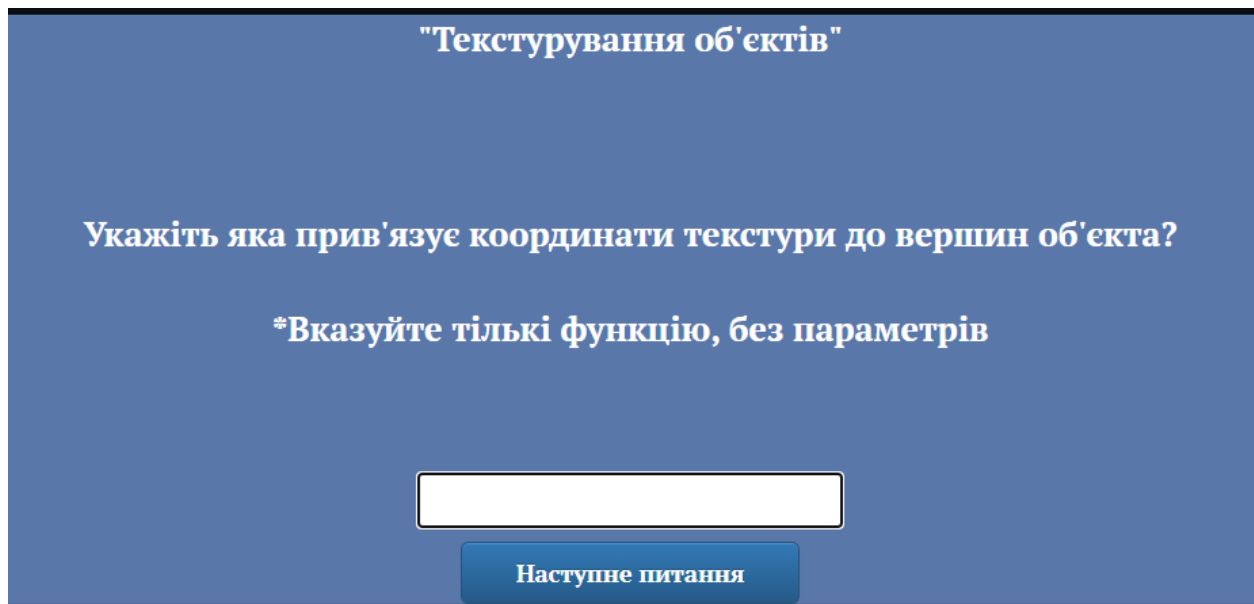


Рисунок 3.6 – Клавіша компіляції після натискання

4. Практична частина.

При розробці цієї частини додатку було адаптовано вже готовий макет з

інтернет ресурсу WEBRIZ [17], а також було використано фреймворк Bootstrap, який надає великі можливості для кращої візуалізації інтерактивного додатку [16]. Було адаптовано область вводу значень та клавішу підтвердження відповіді (рис. 3.7).



"Текстурування об'єктів"

Укажіть яка прив'язує координати текстури до вершин об'єкта?

***Вказуйте тільки функцію, без параметрів**

Наступне питання

Рисунок 3.7 – Питання та область вводу значення для відповіді.

Після проходження 10 завдань користувач отримує результат своєї роботи у вигляді оцінки в процентному змісті з оформленою шкалою заповнення, а також з неправильними відповідями які він надав протягом виконання завдання. Також за допомогою функцій фреймворку bootstrap [16] було оформлено шкалу, яка заповнюється відповідно кількості правильних відповідей користувача (рис. 3.8).

| "Текстування об'єктів" | | |
|---------------------------|--|---|
| Завдання виконано на 20%. | | |
| № | Вопрос | Ответ |
| 1 | Укажіть значення змінної type для цілого, знакового 8-bit типу даних? *Використовуйте великі літери при написанні відповіді | Ви відповіли не вірно. Ваша відповідь: glaux.lib; |
| 2 | Оголосіть глобальну змінну* image | Ви відповіли не вірно. Ваша відповідь: glaux.lib; |

Рисунок 3.8 –Результат виконання

У додатку була адаптована функція перемішування питань (рис. 3.9), що дозволить знизити можливість підказки студентами один одному правильних відповідей.

```

function peremeshat_vse_voprosi(V) {
  for (var i = V.length - 1; i > 0; i--) {
    var j = Math.floor(Math.random() * (i + 1));
    var temp = V[i];
    V[i] = V[j];
    V[j] = temp;
  }
  return V;
}

```

Рисунок 3.9 – Перемішування питань при проходженні завдань

Також реалізована функція шифрування відповідей, як існуючих, так і тих, що дає користувач для порівняння та перевірки (рис. 3.10). Це зроблено з метою запобігання перегляду та знаходження вірних відповідей у кодї сторінки.

```

function kodirov(stroka) {
    var b64ch = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=';
    var b64enc = '';
    var a, b, c;
    var A, B, C, en4;
    for (var i=0; i<stroka.length; ) {
        a = stroka.charCodeAt(i++);
        b = stroka.charCodeAt(i++);
        c = stroka.charCodeAt(i++);
        A = a >> 2;
        B = ((a & 3) << 4) | (b >> 4);
        C = isNaN(b) ? 64:(((b & 15) << 2) | (c >> 6));
        en4 = isNaN(c) ? 64:(c & 63);
        b64enc += b64ch.charAt(A) + b64ch.charAt(B) +
            b64ch.charAt(C) + b64ch.charAt(en4);
    }
    return b64enc;
}

```

Рисунок 3.10 – Шифрування відповідей

Додатково було створено файл NewCode.html та адаптовано data.js для зручності викладача, якщо він забажає змінити питання та відповіді на них.

При запиті до NewCode.html у вікні браузера з'явиться область введення даних, які потрібно закодувати (рис. 3.11) А в файлі data.js зручно виведені питання та окремо відповіді на ці питання, які зберігаються в зашифрованому стані.

Також реалізовано відключення перемішування питань (рис. 3.12-3.13).

Яку відповідь потрібно зашифрувати?

Shifr

Зашифрувати

Зашифрована відповідь: U2hpZnl=

Рисунок 3.11 – Шифрування відповідей

Викладач має змогу сам обрати запитання та відповіді на них, та відкривши один файл, зручно змінити потрібне завдання та за бажанням корегувати роботу додатку на власний розсуд.

```

1 var test= "Текстурування об'єктів";
2 var peremeshat_voprosi = "да";
3 var vopros_1 = "Укажіть значення змінної type для цілого знакового 32-bit типу даних?<br><br> *Використовуйте великі літери при написанні відповіді";
4
5 var vopros_2 = "Оголосіть глобальну змінну* image";
6
7 var vopros_3 = "функція для завдання початкової точки виведення?<br><br>*Вкажіть тільки функцію, без параметрів";
8
9 var vopros_4 = "Укажіть значення змінної type для типу даних з плаваючою точкою? <br><br> *Використовуйте великі літери при написанні відповіді";
10
11 var vopros_5 = "Укажіть значення змінної type для цілого, знакового 8-bit типу даних? <br><br> *Використовуйте великі літери при написанні відповіді ";
12
13 var vopros_6 = "Укажіть яка прив'язує координати текстури до вершин об'єкта? <br><br> *Вкажіть тільки функцію, без параметрів";
14
15 var vopros_7 = "Укажіть яка приймає в якості першого аргументу константу, яка визначає настроюваний параметр, а другий аргумент визначає її значення?<br><br> *Вкажіть тільки функцію, без параметрів";
16
17 var vopros_8 = "Який параметр потрібно встановити щоб повторити текстуру? <br><br> *Використовуйте великі літери при написанні відповіді ";
18
19 var vopros_9 = "Вкажіть яку бібліотеку не підключено? OpenGL32.lib; glu32.lib;<br><br> *Не забувайте ставити крапку з комою в кінці.";
20
21 var vopros_10 = "Режим вирівнювання (GL_UNPACK_ALIGNMENT) може приймати значення? <br><br> *Вкажіть через кому без відступів чотири числових параметра";

```

Рисунок 3.12– Питання до практичної частини

```

var pr_otv_zadachi_1 = "R0xfSU5U";
var pr_otv_zadachi_2 = "QVVYX1JHqkltYwDlUmVjICogaW1hZ2U7";
var pr_otv_zadachi_3 = "Z2xSYXN0ZXJQb3MyZA==";
var pr_otv_zadachi_4 = "R0xfRkxPQVQ=";
var pr_otv_zadachi_5 = "R0xfQ1lURQ==";
var pr_otv_zadachi_6 = "Z2xUZXhDb29yZA==";
var pr_otv_zadachi_7 = "Z2xQaXh1bFN0b3Jl";
var pr_otv_zadachi_8 = "R0xfUkVQRUFU";
var pr_otv_zadachi_9 = "Z2xhdXgubGliOw==";
var pr_otv_zadachi_10 = "MSwyLDQsOA==";

```

Рисунок 3.13– Зашифровані відповіді

Загальний код додатку, а саме файли розмітки, скрипти та файли формального опису зовнішнього вигляду наведені в додатку В.

3.3 Використання програмного додатку

Інтерактивний додаток грає важливу роль в спрощенні проходження практичних завдань та автоматизації процесу навчання.

Для початку роботи потрібно перейти з особистої сторінки сайту дистанційного навчання СумДУ на сторінку додатку, де користувач побачить титульну сторінку (рис.3.14).

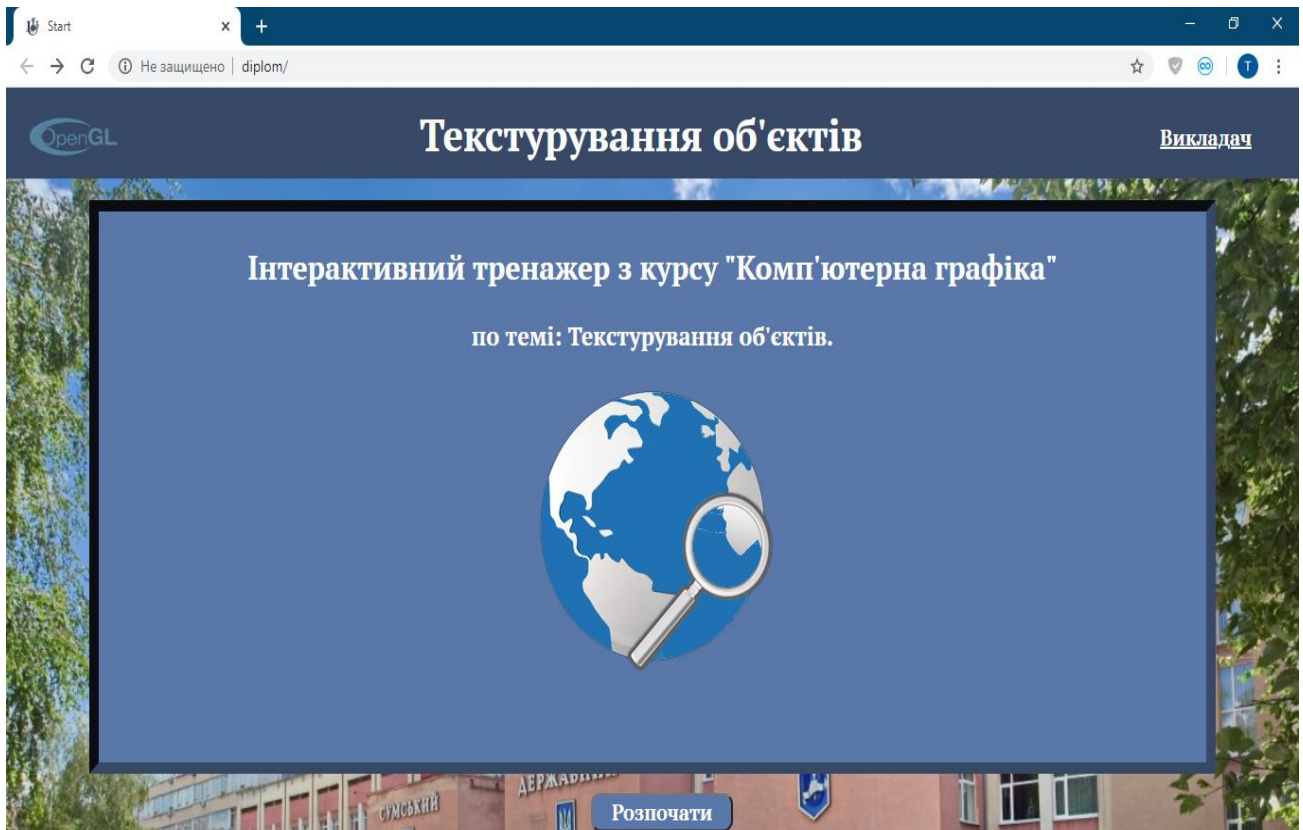


Рисунок 3.14 – Титульна сторінка інтерактивного додатку.

Після натискання кнопки «Розпочати» користувач переходить до наступної вкладки «Теоретичні відомості» (рис.3.15).

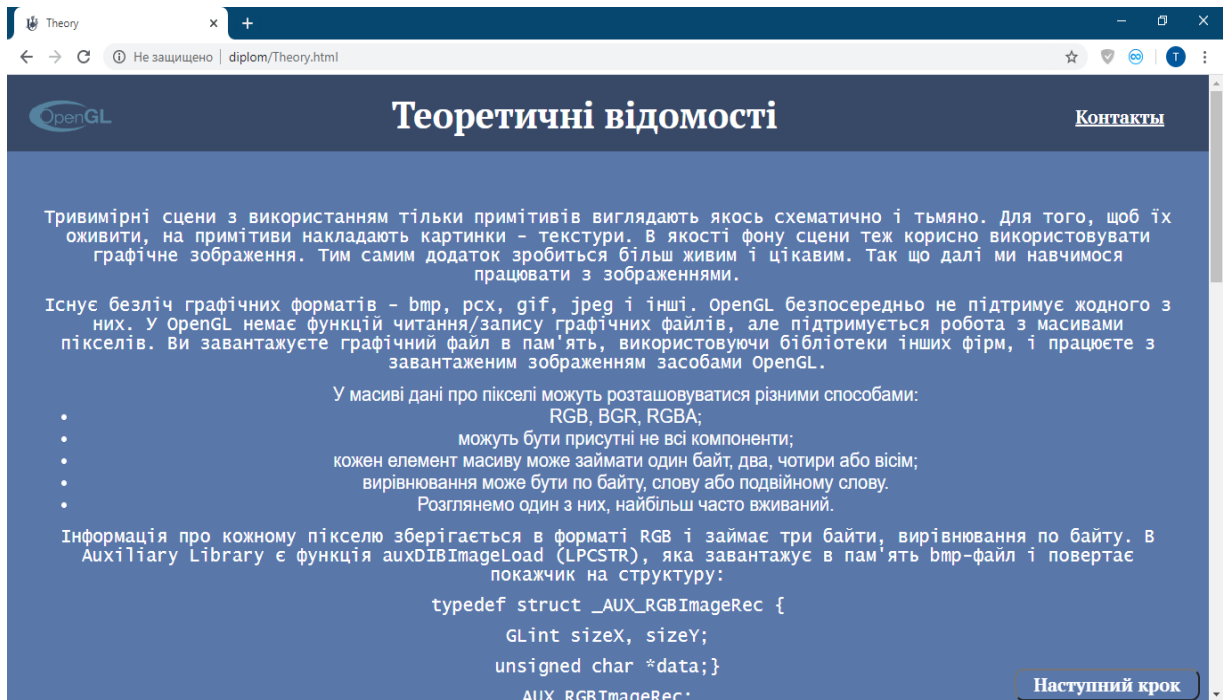


Рисунок 3.15– Сторінка з теоретичним матеріалом.

Після ознайомлення з теоретичним матеріалом користувач натискає кнопку «наступний крок» і переходить до демонстраційної частини (рис.3.16).

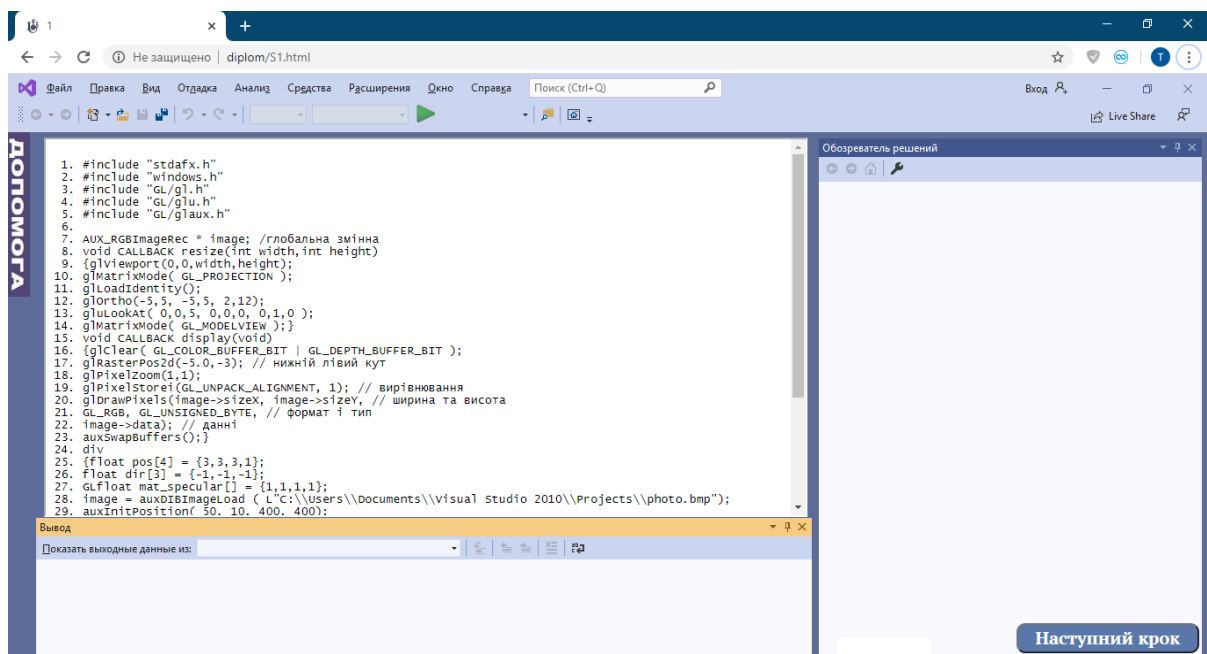


Рисунок 3.16– Сторінка з демонстраційною частиною

На цій сторінці користувач матиме змогу скористатися підказкою за допомогою бокової панелі з інформацією про доданий код в демонстрації (рис. 3.17)

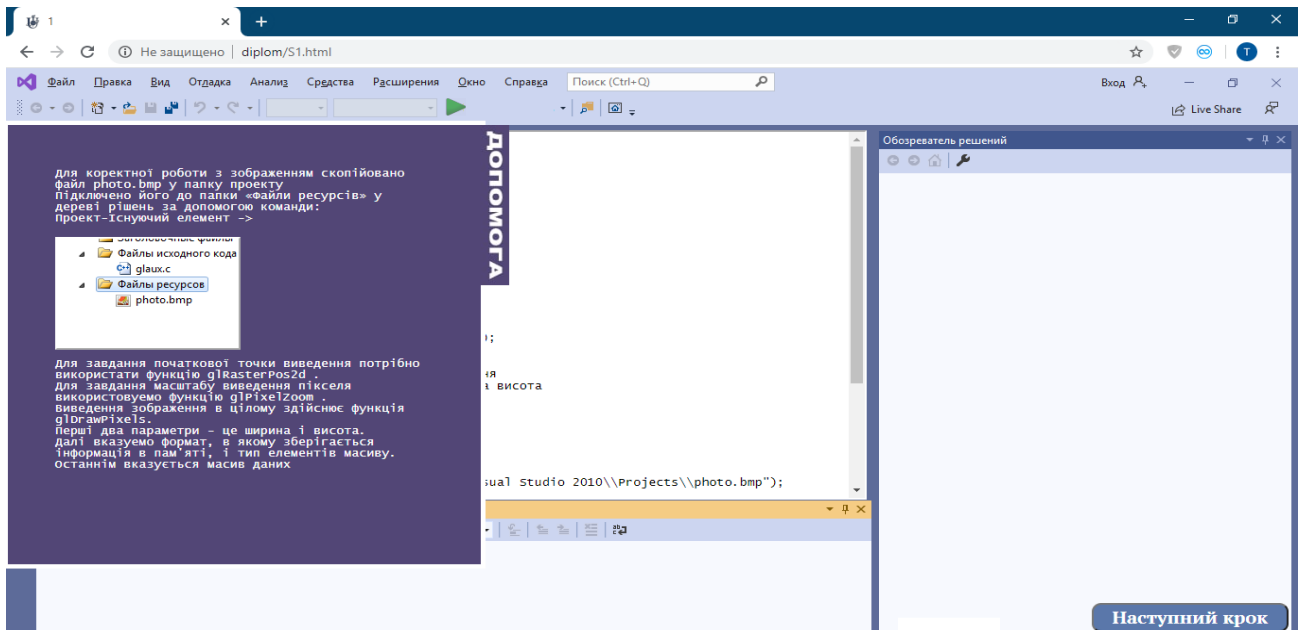


Рисунок 3.17 – Демонстрація бокової панелі з підказками

Також користувач може натиснути клавішу, яка імітує компіляцію в Visual Studio [14](рис.3.18).

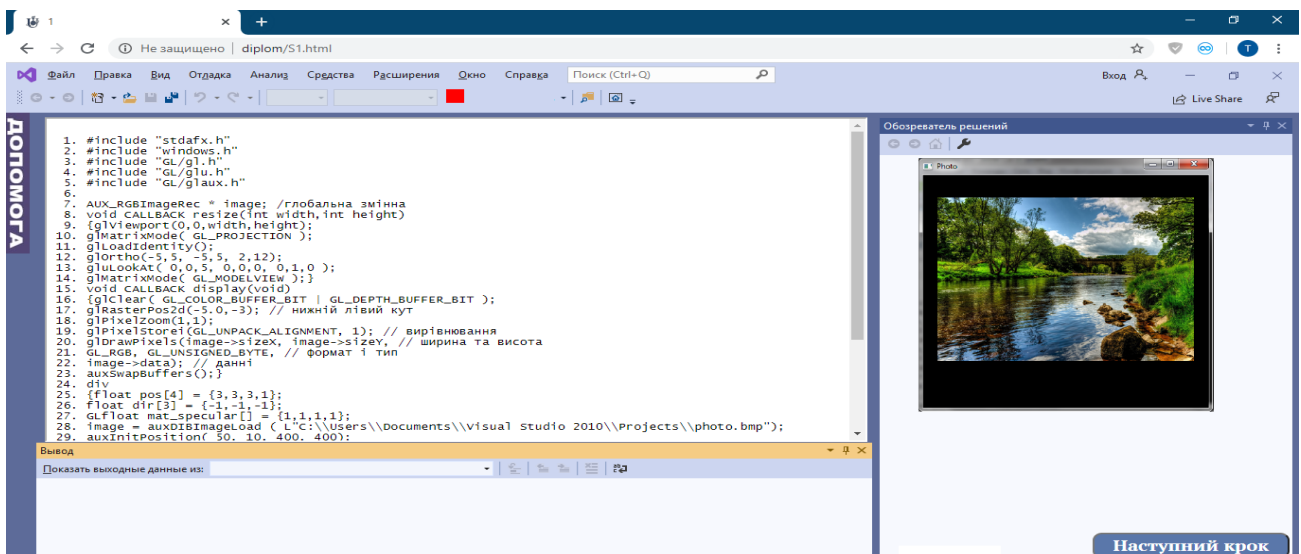


Рисунок 3.18 – Демонстрація роботи кнопки «компіляція»

Таких демонстрацій програмного коду три, після яких користувач матиме змогу перейти до практичних завдань. Кожна частина має свій код, а також свої підказки та окремий результат.

Далі користувач натискає клавішу для переходу до практичної частини додатку і отримує на екрані практичне завдання та область введення даних (рис.3.19).

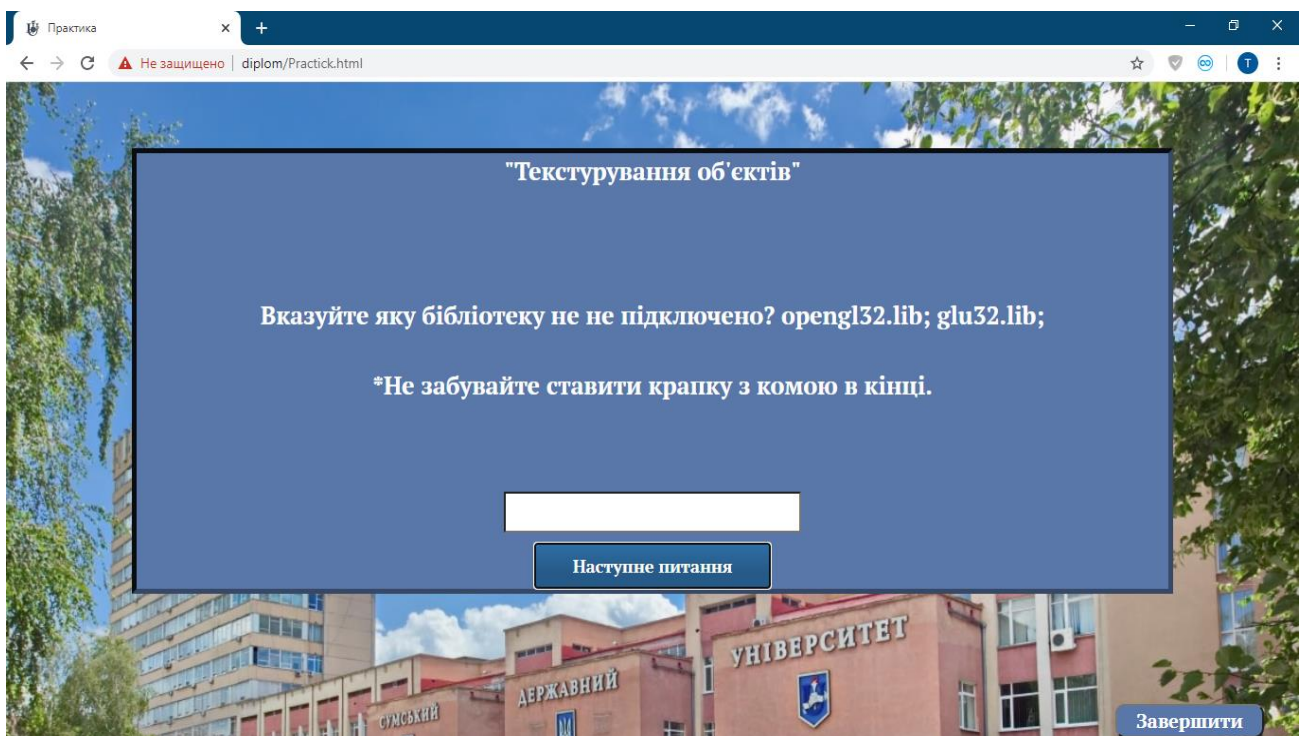


Рисунок 3.19 – Демонстрація практичної частини додатку

Після виконання усіх завдань користувач отримує результат роботи та пояснення, які відповіді були дані невірно (рис. 3.20).

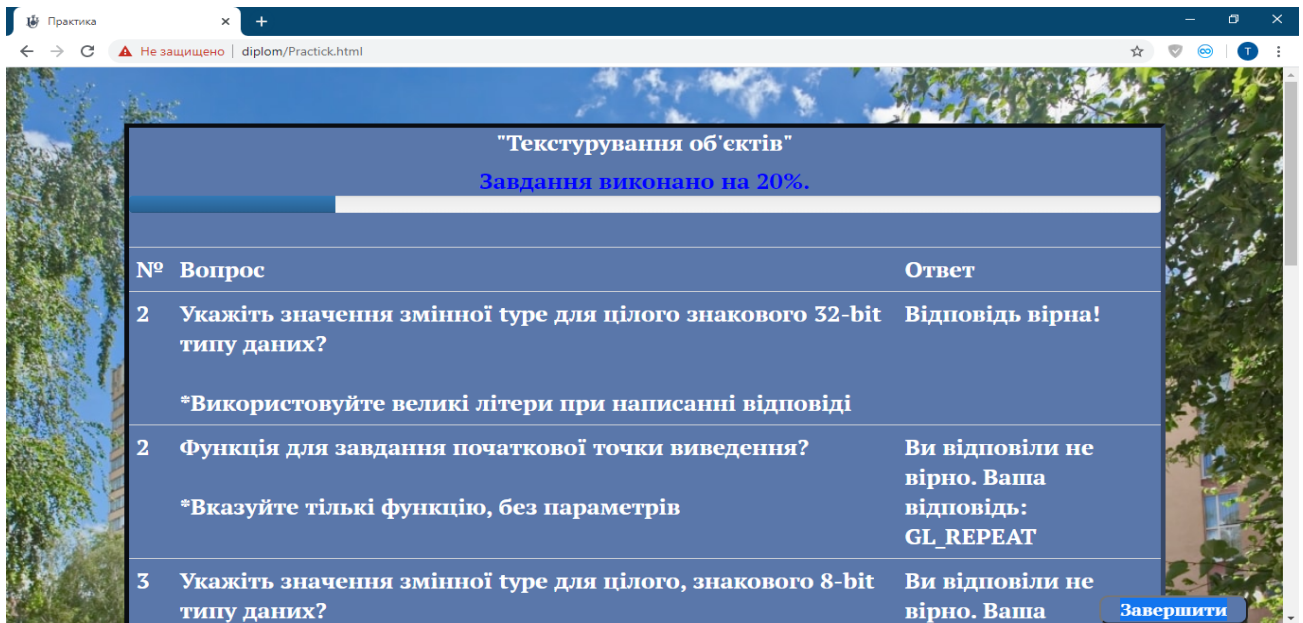


Рисунок 3.20 – Демонстрація результату

Працездатність додатку була перевірена – додаток запускається у таких браузерах: Google Chrome, Яндекс Браузер та браузері Opera на базі Windows 10 (рис. 3.21-3.23).

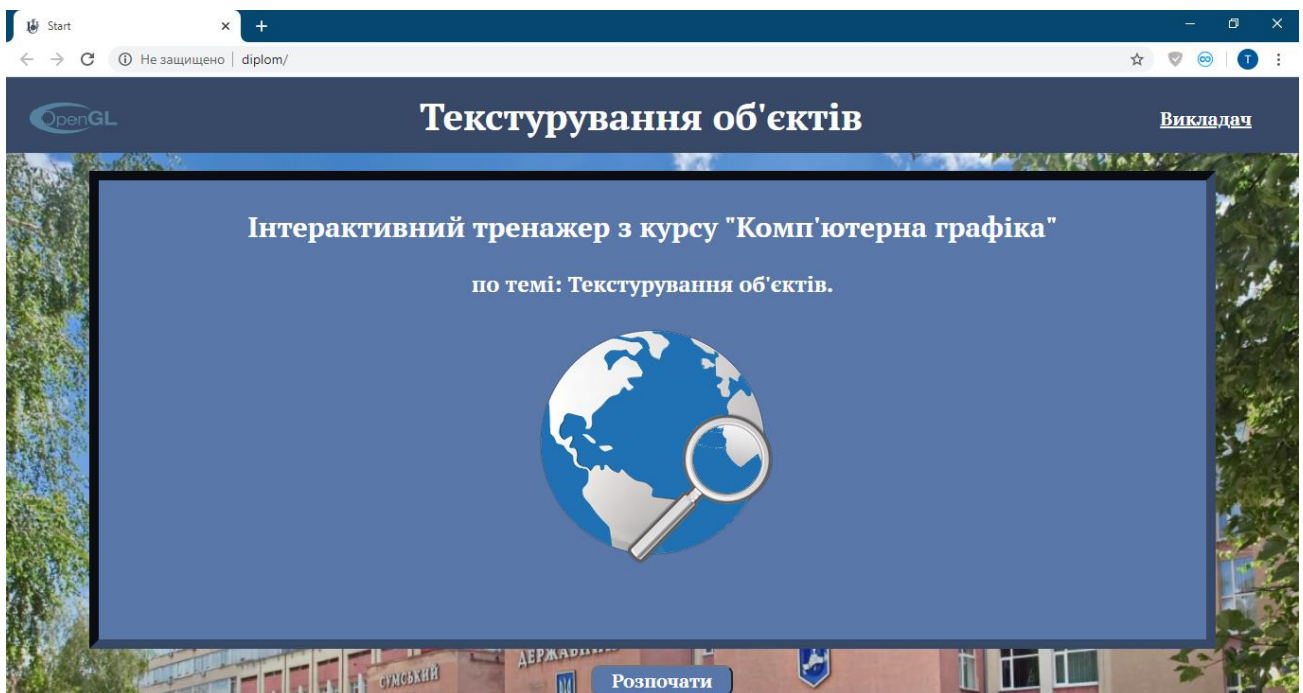


Рисунок 3.21 – Робота додатку в Google Chrome

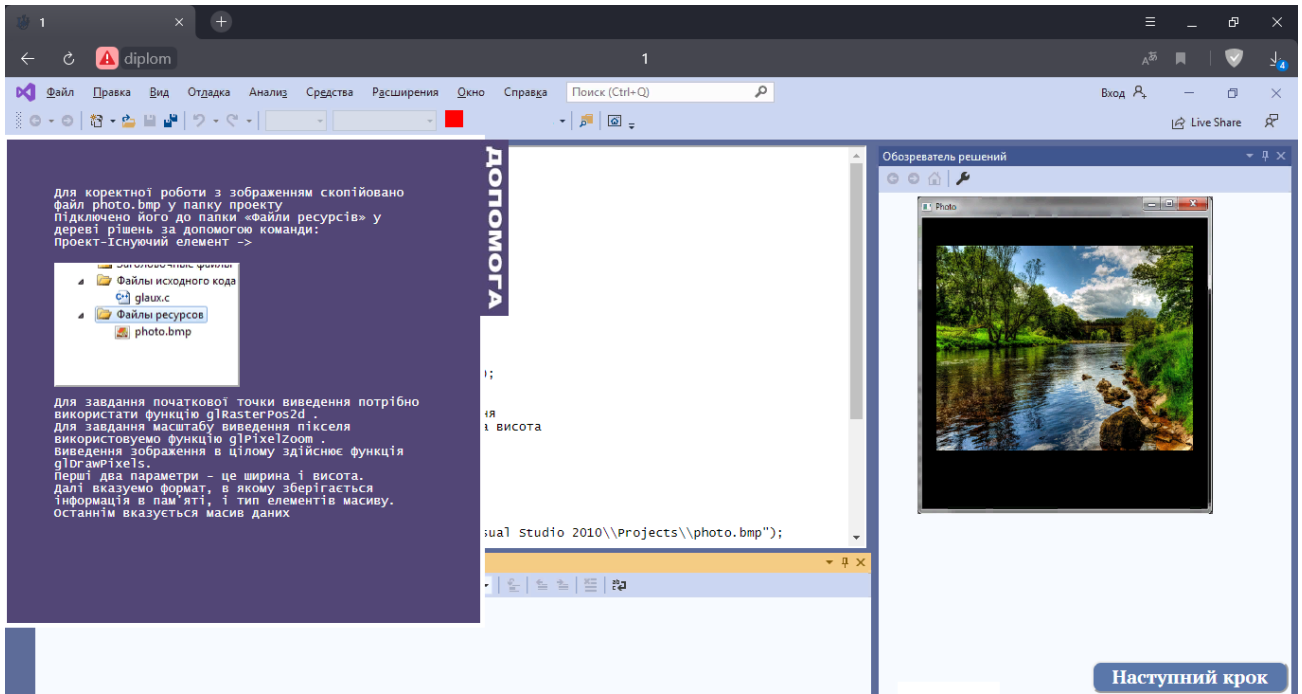


Рисунок 3.22 – Робота додатку в Яндекс браузері

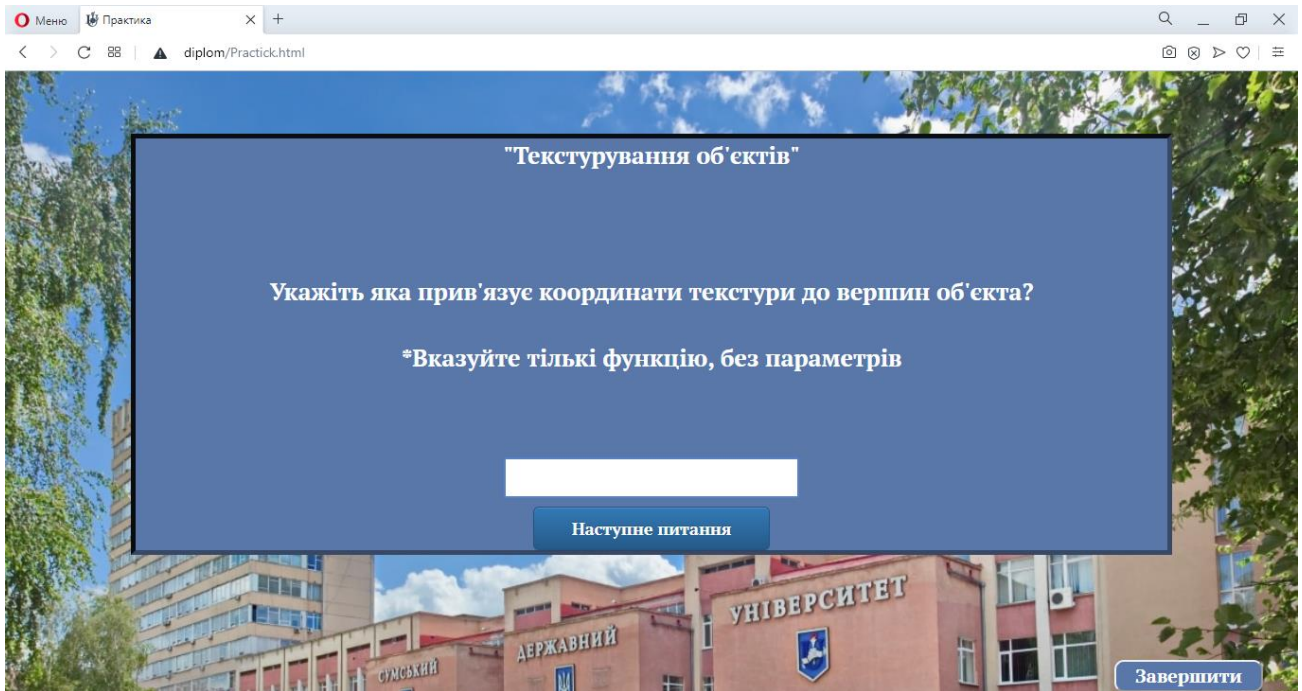


Рисунок 3.23 – Робота додатку в браузері Опера

В результаті практичної реалізації поставлені задачі проекту виконані – в підсумку розроблено веб-тренажер за темою «Текстурування об'єктів» дистанційного курсу «Комп'ютерна графіка».

Даний проект сприятиме підвищенню якості навчання, дозволить студентам швидше та зручніше вивчати матеріали з дисципліни «Комп'ютерна графіка».

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи бакалавра створено інтерактивний додаток для дистанційного навчання Сумського державного університету.

При виконанні роботи було досліджено та проаналізовано подібні аналоги та знайдено недоліки та переваги подібних додатків, після сформовано основні задачі та функціональні вимоги до власного проекту.

За допомогою методології IDEF та мови графічного опису UML проведено структурно-функціональне моделювання роботи додатку. Виконано планування робіт, розроблено календарний план.

Розроблено структуру веб-додатку, зрозумілий та зручний інтерфейс.

Виконано наповнення додатку – розроблено теоретичні та практичні відомості, сформульовано питання тренажеру.

Розроблено механізм проходження завдань та обчислення результату.

В підсумку розроблений продукт допомагатиме студентам перевіряти та закріплювати знання з теми «Текстурування об'єктів» дистанційного курсу «Комп'ютерна графіка».

Даний проект дасть змогу студентам швидше та зручніше проходити навчання при дистанційній освіті, а викладачу контролювати процес навчання. В подальшому додаток можна використовувати при навчанні на денній та заочних форм навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Корбут О. Г. Дистанційне навчання: моделі, технології, перспективи. //Науково-практична конференція «Новітні освітні технології» URL: <http://confesp.fl.kpi.ua/ru/node/1123> (дата звернення: 10.04.2020).
2. Ортинський В.Л. Педагогіка вищої школи. URL:https://pidruchniki.com/12090810/pedagogika/tehnologiya_distsiynogo_navchannya (дата звернення: 11.04.2020).
3. Білоус О. А., Кравченко Ю. А. Дистанційний поточний контроль знань студентів при вивченні математичних дисциплін. – 2016. // Актуальні питання природничо-математичної освіти. 2016. №7-8 . – С. 6-12.
4. Дистанційне навчання СумДУ // Факультет електроніки та інформаційних технологій// кафедра інформаційні технології. URL: <https://elit.sumdu.edu.ua/uk/abiturientam/spetsialnosti-fakultetu/46-itp.html> (дата звернення: 10.04.2020).
5. Інтернет довідник//Як налаштувати дистанційне навчання: досвід та поради вчителів. URL: <https://rubryka.com/article/distance-learning/> (дата звернення: 10.04.2020).
6. Национальный открытый университет «ИНТУИТ» // курс: Создание графических моделей с помощью Open Graphics Library. URL: <https://www.intuit.ru/studies/courses/2313/613/info> (дата звернення: 14.04.2020).
7. Комерційний веб-додаток // Web GL based game engine build for modern web browsers // Shader editor. URL: <http://www.kickjs.org/> (дата звернення: 14.04.2020).
8. Основы HTML. URL: <http://htmlbook.ru/samhtml> (дата звернення: 16.04.2020).

9. Основы CSS. URL: <http://htmlbook.ru/content/osnovy-css> (дата звернення: 16.04.2020).
10. Веб-технологии для разработчиков. JavaScript. URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата звернення: 16.04.2020).
11. Методологія функціонального моделювання IDEF0. URL: <https://nsu.ru/smk/files/idef.pdf> (дата звернення: 14.04.2020).
12. UML діаграми та діаграми послідовності. URL: <http://flash.retejo.info/схефрагхо/uml/diagrama-poslidovnosti> (дата звернення: 14.04.2020).
13. Діаграма компонентів. URL: https://studopedia.com.ua/1_280678_diagrama-komponentiv-component-diagram.html (дата звернення: 10.04.2020).
14. Microsoft Visual Studio. URL: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio (дата звернення: 10.04.2020).
15. jQuery для начинающих. URL: <https://habr.com/ru/post/38208/> (дата звернення: 16.04.2020).
16. Начало работы с bootstrap. URL: <https://bootstrap-4.ru/docs/4.5/getting-started/introduction/> (дата звернення: 12.05.2020).
17. Создание учебного теста с ответами при помощи HTML и JavaScript. URL: <https://webriz.ru/36-sozdanie-uchebnogo-testa-s-otvetami.html> (дата звернення: 12.05.2020).

ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ

1. Призначення й мета створення інтерактивного додатку

1.1. Призначення додатку

Інтерактивний додаток повинен виконувати роль практичного тренажеру для вивчення предмету «Комп'ютерна графіка».

Мета створення додатку

Закріплення теоретичних знань та практичних навичок по темі текстурування об'єктів дисципліни «Комп'ютерна графіка».

1.2. Цільова аудиторія

До цільової аудиторії проекту відносяться наступні категорії:

Студенти дистанційної форми навчання.

Студенти денної та заочної форми навчання.

Викладачі.

Інші зацікавлені відвідувачі.

2. Вимоги до додатку в цілому

2.1 Вимоги до структури й функціонування

Інтерактивний додаток повинен функціонувати у вигляді тренажера на базі системи дистанційної освіти dl.sumdu.edu.ua. Додаток повинен бути розроблений із структурних частин, пов'язаних між собою, та зручними і логічними

переходами між частинами проекту, а також зі зручними функціональними можливостями.

2.2 Вимоги до користувачів та персоналу

Додаток не потребує спеціальних навичок для користування, достатньо володіти навичками користування ПК та інтернет браузером. Для адміністрування додатку потрібно мати навички роботи з мовою програмування Java Script та розробки сайтів.

2.3 Вимоги до збереженні інформації

В результаті роботи додатку повинні зберігатися ім'я користувача та його оцінка, котрі фіксуються на робочому екрані, а також інші дані проходження тренажеру в базі даних на сайті dl.sumdu.edu.ua.

2.4 Вимоги до розмежування доступу

Інформація у додатку є доступною для студентів та викладачів дистанційної форми навчання.

Відповідно до прав доступу можна виділити 2 групи: Користувач (студент) та Розробник.

Користувач має доступ до відкритої частини, може проходити завдання та отримати результат, не може редагувати інформацію в закритих полях.

Розробник має повний доступ до адміністрування додатку: може додавати і видаляти інформацію, а також надавати доступ для студентів. Вхід виконується за спеціальним логіном та паролем.

2.5 Структура додатку

Додаток повинен складатися з наступних розділів:

Початкова сторінка – виводиться довідкова інформація тренажер, вибір мови інтерфейсу;

Теоретична частина – подається довідка про роботу додатку, та необхідні відомості та матеріали для виконання практичної частини;

Практична частина – основа проекту, інтерактивна частина з можливістю проходження практичного завдання та зручним переходом як до кінця тренажеру, так і до теоретичної частини;

Результат – подається інформація про результат проходження практичної частини завдання.

2.6 Навігація

Інтерфейс повинен бути простим, зручним та послідовним. Інформація, повинна бути зручно розташована у робочому вікні та доступна для користувача.

Навігація по додатку повинна давати зручний доступ до всіх відкритих елементів на протязі усього користування.

2.7 Наповнення додатку (контент)

Сторінки будуть наповнені інформацією з практичного завдання по темі «моделювання текстур» предмету «Комп’ютерна графіка» та зручно розподілені по відповідним критеріям та шаблонам. Теоретична частина повинна бути пов’язана з практичною частиною контенту по кожному завданню.

2.8 Вимоги до функціональних можливостей

Керування контентом повинно відбуватися через адміністрування додатку

і мати можливість додавати, видаляти та редагувати всю інформацію через спеціальний доступ.

На «початковій» сторінці буде представлена наступна інформація.

- Інформація про предмет та тема практичного завдання;
- Клавiша Переходу до наступного розділу;

Сторінка – «теоретична частина» містить наступні елементи.

- Теоретичний матеріал по предмету «комп'ютерна графіка»
- Клавiша переходу до практичної частини.
- Гіперпосилання з контактами викладача для зв'язку та консультацією.

Сторінка - «Практична частина» містить такі елементи.

- Практичне завдання.
- Область для розв'язання завдання.
- Клавiша переходу до наступного завдання.
- Клавiша повернення до теоретичної частини..
- Кількість завдань та кількість виконаних завдань.

Сторінка - «Результат» містить такі елементи.

- Оцінка студента.
- Вірні та помилкові відповіді на запитання.

Стиль додатку буде наслідувати інтерфейс програмного продукту Microsoft “Visual Studio” і використовувати білий та сині кольори для дизайну.

2.9 Вимоги до видів забезпечення

Реалізація додатку відбувається на базі:

- HTML5
- CSS3
- JavaScript
- Open Server

Сайт повинен бути виконаний українською, російською та англійською мовами. Повинна бути передбачена можливість перемикання між цими трьома мовами на веб-сайті.

Програмне забезпечення користувача повинне задовольняти наступним вимогам:

- Веб-браузер: Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище;
- Включена підтримка JavaScript, Flash і cookies.

3. Склад і зміст робіт зі створення додатку

Етапи розробки додатку наведено в табл. 1.

Таблиця 1 – Етапи створення додатку

| № | Склад і зміст робіт | Строк розробки (у робочих днях) |
|----|--|---------------------------------|
| 1 | Створення каркасу: Проектування розмітки та наповнення додатку | 4 днів |
| 2 | Технічне налаштування: Налаштування клавіш навігації | 7 днів |
| 3 | Програмування додатку : Налаштування вводу відповідей користувача та виводу результату роботи. | 7 днів |
| 4 | Програмування додаткових елементів: Створення та налаштування елементів керування часу та оцінювання обраних відповідей | 5 днів |
| 5 | Графічне оформлення : Підбір кольорів та графічних оформлень додатку | 4 днів |
| 6 | Ідентифікація користувача: Розроблення та реалізація блоку ідентифікації у додатку | 5 днів |
| 7 | Зберігання результату: Налаштування зберігання та вивід результату викладачу | 4 днів |
| 8 | Наповнення контентом | 4 дні |
| 9 | Налаштування взаємодії компонентів Об'єднання усіх компонентів в один цільний проект | 5 днів |
| 11 | Документація: Створення документації користувача. | 8 днів |

| | |
|--|--------|
| Загальна тривалість робіт (з урахуванням резервного строку на налагодження й виправлення помилок) і строк закінчення проекту | 52 дні |
|--|--------|

4. Вимоги до складу й змісту робіт із введення сайту в експлуатацію

Для вводу в експлуатацію інтерактивного додатку він повинен бути завантажений на сайт та хостинг дистанційної платформи Сумського державного університету і налаштований адміністратором.

ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ

Б.1 Ідентифікація ідеї проекту

Мета даної роботи полягає у створенні інтерактивного додатку по курсу «комп'ютерна графіка» для сайту дистанційного навчання Сумського Державного Університету. Тобто необхідно створити веб-додаток для імітації роботи програми Visual Studio і роботи OpenGL в браузері. Розробити вивід теоретичної частини, практичної частини та підказок по роботі. Також створити вивід результатів для вчителя. Дизайн створений у додатку повинен бути максимально реалістичною відтворений до програми Visual Studio.

Б.2 Деталізація мети методом SMART

Сутність деталізації мети проекту за допомогою SMART-методу впливає з розшифрування термінів, які формують його назву: конкретна (Specific), вимірювана (Measurable), досяжна (Achievable), реалістична (Relevant), обмежена у часі (Time-framed).

S – конкретність, специфічність. Вимагає щоб сформульована мета давала чітке якісне уявлення про специфічні унікальні та інноваційні властивості майбутнього продукту проекту порівняно з іншими його альтернативами.

M – вимірюваність. Передбачає показників вартості які вимірюються. При відсутності фізичних способів та інструментарію виміру використовуються експерти – як інструмент для виміру.

A – узгодженість. Встановлює, що мета повинна впливати з реальних проблем, місії, стратегічних планів, планів розвитку, а також узгоджуватись з інтересами зацікавлених сторін проекту.

R – реалістичність, релевантність. Показує, що мета є такою, яку можливо досягти з урахуванням реально доступних ресурсних можливостей та обмежень (людських, фінансових тощо).

T – обмеженість в часі. Зумовлює необхідність «прив’язки» мети до певних обґрунтованих термінів її досягнення (або початку та тривалості дій по її досягненню).

Після проведення аналізу методом SMART можна визначити кінцеву мету проекту: створити програмний продукт, який дозволить поліпшити роботу студента та викладача дистанційної форми навчання. Мета є досяжною, підґрунтям створення програмного продукту є навички отримані на курсі з дисципліни ТСПП і веб-технології та веб-дизайн. Проект буде виконано вчасно, що підтверджується календарним планом проекту.

Б.3 Фаза розробки

Планування змісту структури робіт IT- проекту (WBS)

WBS є ієрархічною та інкрементною декомпозицією проекту у фазі, кінцеві результати та пакети робіт. Вона є ієрархічною структурою, що показує подальший розподіл необхідних для виконання мети зусиль.

Перший рівень структури називатиметься «WEB-тренажер для дистанційного навчання в Сумському державному університеті». Цей рівень декомпозиується на чотири рівні: ініціалізація, планування, реалізація, завершення.

На рисунку Б.1 представлена WBS-структура проекту.

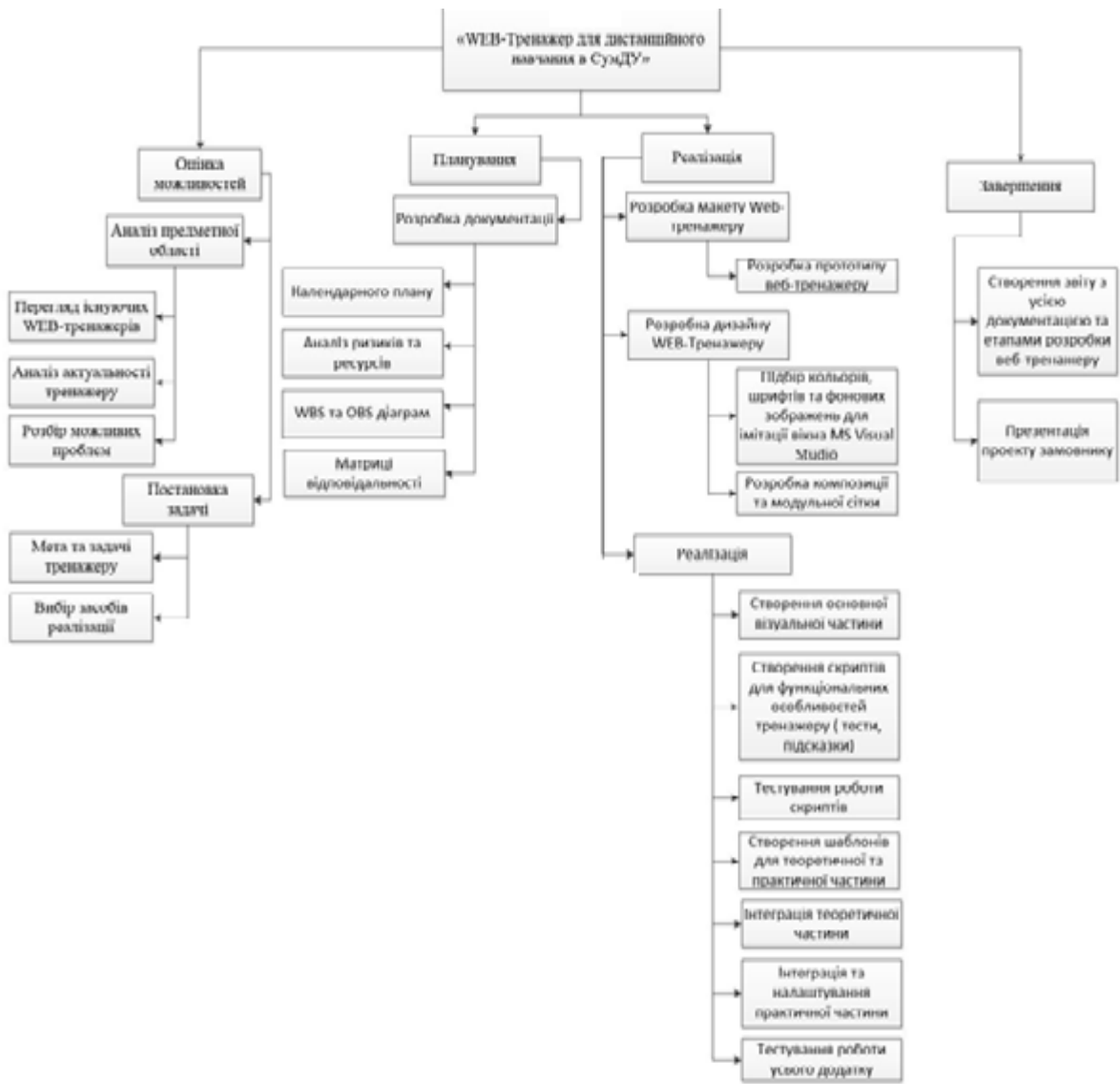


Рисунок Б.1 – WBS-структура проекту

Б.4 Планування структури організації, для впровадження готового проекту (OBS)

Організаційна структура проекту (OBS) – є графічним відображенням учасників проекту (фізичних та юридичних осіб) та їхніх відповідальних осіб, залучених до реалізації проекту.

На рисунку Б.2 представлена OBS-структура проекту.

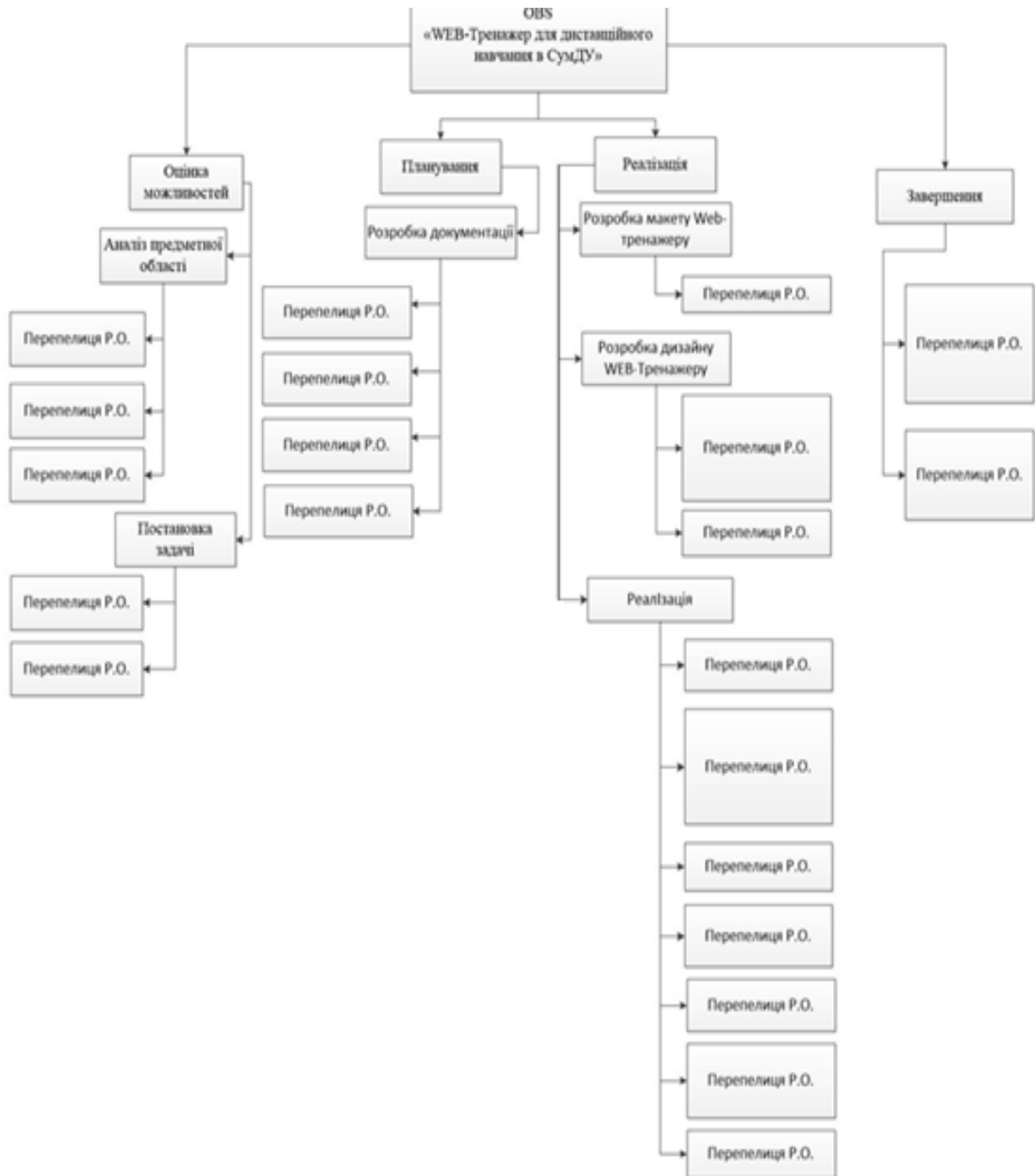


Рисунок Б.2 – OBS-структура проекту

Б.5 Побудова матриці відповідальності (виконавців пакетів робіт)

На підставі OBS та WBS структур будують матрицю відповідальності проекту. Матриця відповідальності закріплює за кожною елементарною роботою виконавця. Матриця відповідальності представлена в таблиці Б.1.

Таблиця Б.1 – Матриця відповідальності

| WBS/OBS | | Перепелиця Р. О. | Баранова І.В. |
|-----------------|---|------------------|---------------|
| Технічні знання | Java Script | R | I |
| Дизайн | Створення макетів | R | |
| | Редагування | R | |
| | Затвердження макету кожної сторінки додатку | R | A |
| Програмування | Кодинг модулів | R | |
| | Створення пробних шаблонів сторінок | R | |
| | Загальне тестування коду | R | |
| Тестування | Функціональне тестування | R | A |
| | Інтерфейсу | R | A |
| | Продуктивності | R | A |
| | «Юзабіліті» | R | A |
| | Виправлення помилок | R | A |
| Здача проекту | Технічне завдання | R | I |
| | Опис програмного продукту | R | A |
| | Інструкція користувача | R | A |

Б.6 Побудова календарного графіку виконання ІТ-проекту

Діаграма Ганта – горизонтальна лінійна діаграма, на якій задачі проекту представляються протяжними в часі відрізками, що характеризуються датами початку та закінчення, затримками і, можливо, іншими тимчасовими параметрами. Для отримання реального уявлення про тривалість виконання робіт з урахуванням обмеженості у використанні ресурсів, на підставі часткової мережевої моделі, а також, проекту в цілому з урахуванням вихідних та святкових днів, було побудовано календарний графік робіт. Цей графік представлено за допомогою програмного засобу MS Project.

Графік виконання дипломного проекту представлено у вигляді Діаграми Ганта на рисунку Б.3.

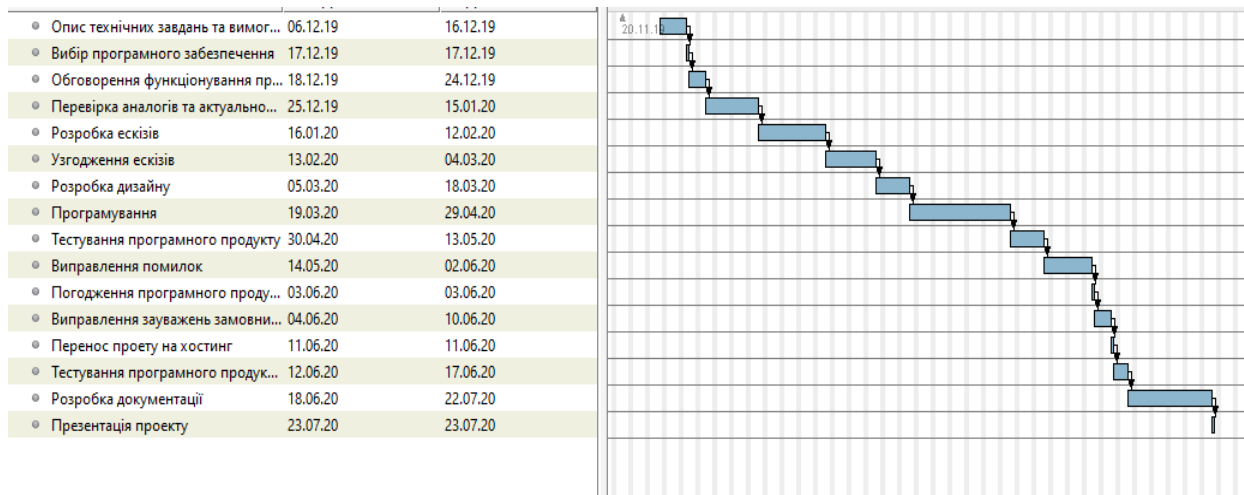


Рисунок Б.3 - Діаграма Ганта

Б.7 Ідентифікація ризиків

Ідентифікація ризиків – це виявлення ризиків, здатних вплинути на проект, і документальне оформлення їх характеристик. Це ітеративний процес, який

періодично повторюється на всьому протязі проекту, оскільки в рамках його життєвого циклу можуть виявлятися нові ризики.

Якісний аналіз передує кількісному. Він передбачає визначення факторів ризику, ідентифікацію потенційних областей виникнення ризику, виявлення напрямків діяльності та етапів, на яких може реалізуватися ризик. Протягом якісного аналізу також встановлюється можливість кількісної оцінки ризиків, реалізація яких може вплинути на діяльність підприємства.

Кількісна оцінка ризиків часто супроводжує якісну оцінку і також вимагає процес ідентифікації ризиків.

На рисунку Б.4 наведено класифікацію ризиків.

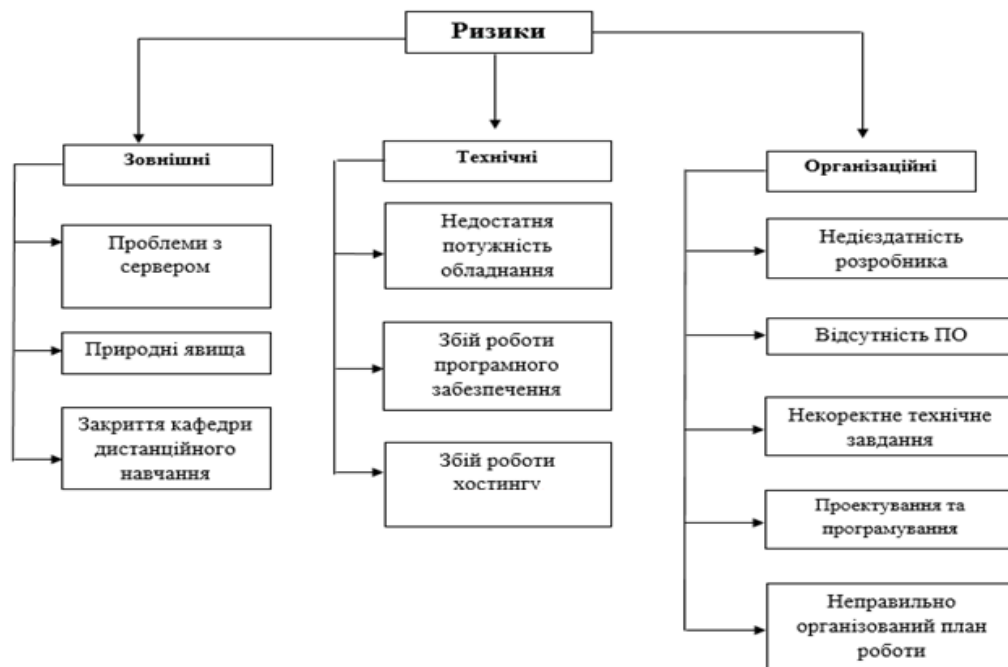


Рисунок Б.4 – Ризики

Матриця ризиків

Ризики представити за допомогою RBM матриці (Risk Breakdown Matrix)

Класифікація ризиків за імовірністю виникнення:

- слабо ймовірні;
- мало ймовірні;
- імовірні;
- досить імовірні;
- майже імовірні.

Класифікація ризиків за імовірністю виникнення за величиною втрат:

- мінімальна;
- низька
- середня
- висока
- максимальна

Виконаємо класифікацію ризиків даного проекту. Для цього складемо таблицю Б.2.

Таблиця Б.2 – Класифікація ризиків дипломного проекту

| Ризик | | Ймовірність виникнення | Величина втрат |
|---|-----|-----------------------------------|---------------------------|
| Проблеми з сервером | R1 | 3 | 4 |
| Природні явища | R2 | 4 | 2 |
| Закриття центру дистанційного навчання | R3 | 2 | 4 |
| Недостатня потужність обладнання | R4 | 2 | 3 |
| Збій роботи програмного забезпечення | R5 | 4 | 5 |
| Збій роботи хостингу | R6 | 2 | 4 |
| Недієздатність розробника | R7 | 1 | 5 |
| Некоректне технічне завдання | R8 | 2 | 2 |
| Проектування та програмування | R9 | 2 | 3 |
| Неправильно організований план роботи | R10 | 1 | 2 |

| | | | | |
|--|-----|-------|-------|----|
| | R10 | | | R7 |
| | R8 | R4;R9 | R3;R6 | |
| | | | R1 | |
| | R2 | | | R5 |
| | | | | |

Рисунок Б.5 – Матриця імовірності втрат

Класифікація за ступенем впливу та за рівнем ризику (табл. Б.3)

Таблиця Б.3 – Класифікація за ступенем впливу та за рівнем ризику

| Ризик | | Ступінь впливу | Рівень ризику |
|--|-----|----------------|---------------|
| Проблеми з сервером | R1 | 15 | Недопустимі |
| Природні явища | R2 | 8 | Виправдані |
| Закриття центру дистанційного навчання | R3 | 17 | Неприпустимі |
| Недостатня потужність обладнання | R4 | 10 | Виправдані |
| Збій роботи програмного забезпечення | R5 | 14 | Виправдані |
| Збій роботи хостингу | R6 | 8 | Виправдані |
| Недієздатність розробника | R7 | 4 | Прийнятні |
| Некоректне технічне завдання | R8 | 10 | Виправдані |
| Проектування та програмування | R9 | 14 | Виправдані |
| Неправильно організований план роботи | R10 | 5 | Виправдані |

План по усуненню ризиків:

- вибір потужного обладнання для виконання проекту;
- зіставлення структурованого плану роботи;
- враховувати досвід аналогів;
- ретельний вибір інструментів виконання проекту.

ДОДАТОК В. КОД ДОДАТКУ

Файл Index.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Start</title>
<link href="https://fonts.googleapis.com/css2?family=PT+Serif:ital,wght@0,700;1,400&display=swap"
rel="stylesheet">
<link rel="stylesheet" type="text/css" href='CSS/title.css'/>
<link rel="shortcut icon" href="/img/favicon.ico" type="image/x-icon">
</head>
<body id="fonn">
<header>
<div> 
<h1 id="text1"> Текстування об'єктів</h1> </div>
<ul class = "menu">
<li><a style="color:white" href = https://dl.sumdu.edu.ua/login target="_blank">Викладач</a></li>
</ul>
</header>
<div id="content2">
<h2>Інтерактивний тренажер з курсу "Комп'ютерна графіка" </h2>
<h3>по темі: Текстування об'єктів.</h3>

</div>
<div id="footer">
<button id="button1" class="button" onclick="location.href='Theory.html';"> Розпочати</button>
</div>
</body>
</html>

```

Файл Theory.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Theory</title>
<link href="https://fonts.googleapis.com/css2?family=PT+Serif:ital,wght@0,700;1,400&display=swap"
rel="stylesheet">
<link rel="stylesheet" type="text/css" href='CSS/title.css'/>
<link rel="shortcut icon" href="/img/favicon.ico" type="image/x-icon">
</head>
<body>
<div>
<header>
<div> 

```

```

<h1 id="text1">Теоретичні відомості</h1> </div>
<ul class = "menu">
<li><a style="color:white" href = "https://dl.sumdu.edu.ua/login" target="_blank">Контакти</a></li>
</ul>
</header>
</div>
<div id="content1">

```

<p> Тривимірні сцени з використанням тільки примітивів виглядають якось схематично і тьмяно. Для того, щоб їх оживити, на примітиви накладають картинки - текстури. В якості фону сцени теж корисно використовувати графічне зображення. Тим самим додаток зробиться більш живим і цікавим. Так що далі ми навчимося працювати з зображеннями.</p>

<p> Існує безліч графічних форматів - bmp, pcx, gif, jpeg і інші. OpenGL безпосередньо не підтримує жодного з них. У OpenGL немає функцій читання/запису графічних файлів, але підтримується робота з масивами пікселів. Ви завантажуєте графічний файл в пам'ять, використовуючи бібліотеки інших фірм, і працюєте з завантаженим зображенням засобами OpenGL.</p>

```
<p>
```

```
<ul>У масиві дані про пікселі можуть розташовуватися різними способами:
```

```
<li>RGB, BGR, RGBA; </li>
```

```
<li>можуть бути присутні не всі компоненти;</li>
```

```
<li>кожен елемент масиву може займати один байт, два, чотири або вісім; </li>
```

```
<li>вирівнювання може бути по байту, слову або подвійному слову. </li>
```

```
<li>Розглянемо один з них, найбільш часто вживаний.
```

```
</li>
```

```
</ul>
```

```
<p>
```

<p>Інформація про кожному пікселю зберігається в форматі RGB і займає три байти, вирівнювання по байту. В Auxiliary Library є функція auxDIBImageLoad (LPCSTR), яка завантажує в пам'ять bmp-файл і повертає покажчик на структуру:</p>

```
<p>typedef struct _AUX_RGBImageRec {</p>
```

```
<p>GLint sizeX, sizeY;</p>
```

```
<p>unsigned char *data;}</p>
```

```
<p>AUX_RGBImageRec;</p>
```

```
<p>
```

Для простоти будемо користуватися саме цією функцією.

У OpenGL є функції для виведення масиву пікселів на екран (glDrawPixels), копіювання (glCopyPixels), масштабування (gluScaleImage). Тут ми розглянемо тільки glDrawPixels. Всі інші функції роботи з зображеннями влаштовані схожим чином. Для того, щоб відобразити графічний файл у вікні OpenGL, ви повинні завантажити його в пам'ять, вказати вирівнювання, встановити точку, з якої починається виведення зображення, і вивести його на екран.</p>

```
<p>
```

Функція для виведення пікселів растрового зображення має наступний прототип:</p>

```
<p>
```

```
void glDrawPixels (GLsizei width, GLsizei height, GLenum format, GLenum type, const GLvoid *
pixels);</p>
```

```
<p>
```

Змінні width и height визначають розміри відображуваного растрового масиву. Лівий нижній кут масиву буде розташований в поточній позиції растра.</p>

```
<h4>
```

```
<table>
```

```
<tr>
```

```
<th id="r">Значення змінної format</th>
```

```
<th>Формат пікселів</th>
```

```
</tr>
```

```
<tr>
```

```
<td id="r">GL_RGB</td>
```

```
<td>(R, G, B)</td>
```

```
</tr>
```

```
<tr>
```



```

<td id="r">GL_RGBA</td>
<td>(R, G, B, A)</td>
</tr>
<tr>
<td id="r">GL_RED</td>
<td>Тільки червона компонента</td>
</tr>
<tr>
<td id="r">GL_GREEN</td>
<td>Тільки зелена компонента</td>
</tr>
<tr>
<td id="r">GL_BLUE</td>
<td>Тільки синя компонента</td>
</tr>
<tr>
<td id="r">GL_LUMINANCE</td>
<td>Яскравість</td>
</tr>
</table></h4>
<h4>
<table>
<tr>
<th id="r">Значення змінної type</th>
<th>Тип даних</th>
</tr>
<tr>
<td id="r">GL_UNSIGNED_BYTE</td>
<td>беззнакове 8-bit ціле (unsigned char або GLubyte)</td>
</tr>
<tr>
<td id="r">GL_BYTE</td>
<td>знакове 8-bit ціле (char або GLbyte)</td>
</tr>
<tr>
<td id="r">GL_BYTE</td>
<td>знакове 8-bit ціле (char або GLbyte)</td>
</tr>
<tr>
<td id="r">GL_BITMAP</td>
<td>беззнакове 8-bit ціле (unsigned char або GLubyte) в тому ж форматі, що і бітовий масив</td>
</tr>
<tr>
<td id="r">GL_UNSIGNED_SHORT</td>
<td>беззнакове 16-bit ціле (unsigned short або GLushort)</td>
</tr>
<tr>
<td id="r">GL_SHORT</td>
<td>знакове 16-bit ціле (short або GLshort)</td>
</tr>
<tr>
<td id="r">GL_UNSIGNED_INT</td>
<td>беззнакове 32-bit ціле (unsigned int або GLuint)</td>
</tr>
<tr>
<td id="r">GL_INT</td>
<td>знакове 32-bit ціле (int або GLint)</td>
</tr>

```

```

<tr>
  <td id="r">GL_FLOAT</td>
  <td>з плаваючою точкою (float або GLfloat)</td>
</tr>
</table></h4>
<p>
Останній аргумент в цій функції є покажчиком на нульовий елемент відображуваного
масиву.</p>
<p>
Функція glPixelStore приймає в якості першого аргументу константу, яка визначає настроюваний
параметр, а другий аргумент визначає її значення.</p>
<p>При роботі з бітовими масивами потрібно налаштування наступних параметрів:</p>
<p>1. Порядок проходження байт (GL_UNPACK_SWAP_BYTES) за замовчуванням FALSE. Це
означає, що байти розташовані в правильному порядку і розпакування не потрібно.</p>
<p>2. Режим вирівнювання (GL_UNPACK_ALIGNMENT) може приймати значення 1, 2, 4, 8, а
за замовчуванням має значення.</p>
<p>Зображення виводяться прямо на екран. Тому все відбувається в двовимірних координатах.
Позиція, з якої починається виведення зображення, вказується за допомогою функції glRasterPos2d (x, y).</p>
<p>Також ви можете встановити розмір пікселя, викликавши функцію glPixelZoom. Перший
параметр цієї функції - ширина, другий - висота пікселя. Викликаємо цю функцію з аргументами (1,1), що
відповідає нормальному пікселю. Якщо замінити (1,1) на (3,2) – ви побачите, як картинка розтягнеться в три рази
по горизонталі і в два рази по вертикалі. Це сталося, тому що тепер кожен піксель зображення відповідає
прямокутнику 3x2 у вікні.</p>
</div>
<div id="footer">
<button id="button2" class="button" onclick="location.href='S1.html';"> Наступний крок </button>
</div>
</body>
</html>

```

Файл S1

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>1</title>
<link href="https://fonts.googleapis.com/css2?family=PT+Serif:ital,wght@0,700;1,400&display=swap"
rel="stylesheet">
<link rel="stylesheet" type="text/css" href='CSS/title.css'/>
<link rel="shortcut icon" href="/img/favicon.ico" type="image/x-icon">
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.1/jquery.min.js"></script>
<script type="text/javascript" src="js/panel.js" ></script>
</head>
<body id="fon">
<div id="pract">
<div id="left-Top">
</div>
<div id="left">
<ol>
<li>#include "stdafx.h"</li>
<li>#include "windows.h" </li>
<li>#include "GL/gl.h" </li>
<li>#include "GL/glu.h" </li>
<li>#include "GL/glaux.h" </li>

```

```

</li></li>
<li>AUX_RGBImageRec * image; /глобальна змінна</li>
<li>void CALLBACK resize(int width,int height)
<li>{ glViewport(0,0,width,height);</li>
<li> glMatrixMode( GL_PROJECTION );</li>
<li> glLoadIdentity();</li>
<li> glOrtho(-5,5, -5,5, 2,12); </li>
<li> gluLookAt( 0,0,5, 0,0,0, 0,1,0 );</li>
<li> glMatrixMode( GL_MODELVIEW );} </li>
<li> void CALLBACK display(void)</li>
<li>{ glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );</li>
<li>glRasterPos2d(-5.0,-3); // нижній лівий кут</li>
<li>glPixelZoom(1,1);</li>
<li>glPixelStorei(GL_UNPACK_ALIGNMENT, 1); // вирівнювання</li>
<li>glDrawPixels(image->sizeX, image->sizeY, // ширина та висота</li>
<li>GL_RGB, GL_UNSIGNED_BYTE, // формат і тип</li>
<li>image->data); // данні</li>
<li>auxSwapBuffers();}</li>
<li>div</li>
<li>{float pos[4] = {3,3,3,1};</li>
<li>float dir[3] = {-1,-1,-1};</li>
<li>GLfloat mat_specular[] = {1,1,1,1};</li>
<li>image = auxDIBImageLoad ( L"C:\\Users\\Documents\\Visual Studio 2010\\Projects\\photo.bmp");</li>
<li>auxInitPosition( 50, 10, 400, 400);</li>
<li>auxInitDisplayMode( AUX_RGB | AUX_DEPTH | AUX_DOUBLE );</li>
<li>auxInitWindow( L"Тым" );</li>
<li>auxIdleFunc(display);</li>
<li>auxReshapeFunc(resize);</li>
<li>glEnable(GL_DEPTH_TEST);</li>
<li>glEnable(GL_COLOR_MATERIAL);</li>
<li>glEnable(GL_LIGHTING);</li>
<li>glEnable(GL_LIGHT0);</li>
<li>glLightfv(GL_LIGHT0, GL_POSITION, pos);</li>
<li>glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, dir);</li>
<li>auxMainLoop(display);}</li>
</ol>
</div>
</div>
<div class="panel">
<a class="handle" href="#">Смена</a>
<span>
Для коректної роботи з зображенням скопійовано файл photo.bmp у папку проекту <br/>
Підключено його до папки «Файли ресурсів» у дереві рішень за допомогою команди:<br/>
Проект-Існуючий елемент -><br/>

<br/><br/><br/><br/><br/><br/><br/><br/>
Для завдання початкової точки виведення потрібно використати функцію glRasterPos2d .<br/>
Для завдання масштабу виведення пікселя використовуємо функцію glPixelZoom .<br/>
Виведення зображення в цілому здійснює функція glDrawPixels. <br/>
Перші два параметри - це ширина і висота. <br/>
Далі вказуємо формат, в якому зберігається інформація в пам'яті, і тип елементів масиву. <br/>
Останнім вказується масив даних
</span>
</div>
<div id="footer">
<button id="button2" class="button" onclick="location.href='Theory2.html';"> Наступний крок </button>
</div>
<b id="zap" class="spoiler-title"></b>

```

```

<div class="spoiler-body">
<div id="fin1">

</div></div>
</div>
<script type="text/javascript">
$(document).ready(function(){
$('.spoiler-body').hide();
$('.spoiler-title').click(function(){
  $(this).toggleClass('opened').toggleClass('closed').next().slideToggle();
  if($(this).hasClass('opened')) {
    $(this).html('');
  }
  else {
    $(this).html('');
  }
});
});
</script>
</body>
</html>

```

Файл Theory2.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Theory</title>
<link href="https://fonts.googleapis.com/css2?family=PT+Serif:ital,wght@0,700;1,400&display=swap"
rel="stylesheet">
<link rel="stylesheet" type="text/css" href='CSS/title.css'/>
<link rel="shortcut icon" href="/img/favicon.ico" type="image/x-icon">
</head>
<body>
<div>
<header>
<div> 
<h1 id="text1"> Теоретичні відомості</h1> </div>
<ul class = "menu">
<li><a style="color:white" href = "https://dl.sumdu.edu.ua/login" target="_blank">Контакты</a></li>
</ul>
</header>
</div>
<div id="content1">
Одного виведення зображень недостатньо для створення повноцінних тривимірних сцен.<br/>
Часто виникає потреба накладати зображення на тривимірні об'єкти і повертати/зсувати їх.<br/>
Для цих цілей існують текстури. Також текстури допоможуть покрити поверхню всього об'єкта у вигляді мозаїки.<br/>
Скажімо, коли у вас є цегляна стіна, то вам не треба завантажувати зображення з купою цегли.<br/>
Досить завантажити одну цеглину і вказати, що цю текстуру потрібно розмножити по всій площині.<br/>
Спочатку розберемо створення і накладення текстур на площину.<br/>
Для того, щоб накласти текстуру на об'єкт, необхідно:<br/>
<ul>
<li>Завантажити графічний файл в пам'ять.</li>

```

- Створити ім'я-ідентифікатор текстури.
- Зробити його активним.
- Створити саму текстуру в пам'яті.
- Встановити параметри текстури.
- Встановити параметри взаємодії текстури з об'єктом.
- Зв'язати координати текстури з об'єктом.

Було підключено бібліотеки `opengl32.lib`; `glu32.lib`; `glaux.lib`. </br></br>

Оголосить наступні глобальні змінні:

-
- unsigned int photo_tex;
- AUX_RGBImageRec* photo_image;
- unsigned int space_tex;
- AUX_RGBImageRec* space_image;
-

Змінні `photo_tex` і `space_tex` слугуватимуть ідентифікаторами текстур. А в `photo_image` і `space_image` ми завантажимо `bmp`-файли. Приклад другого файлу для завантаження `space.bmp` </br>

```
 </br>
```

Потрібно зазначити, що текстури в OpenGL повинні мати розмір $2n \times 2m$, де n і m цілі числа. Це зроблено для прискорення роботи, тому що стискати або розтягувати такі текстури швидше і зручніше. Отже рекомендується використовувати графічні файли з розміром зображення, кратним ступеню двійки.</br>

Вставте цей код в функцію `main`.</br>

```
photo_image = auxDIBImageLoad("photo.bmp");</br>
```

```
space_image = auxDIBImageLoad("space.bmp");</br>
```

Тепер необхідно створити ім'я-ідентифікатор текстури. Його потрібно створювати, коли в додатку використовується більше однієї текстури, щоб була можливість якось їх розрізнити. В іншому випадку, коли текстура лише одна, ідентифікатор їй не потрібен.</br>

Функція `glGenTextures` приймає на вхід два параметра. Перший вказує кількість імен-ідентифікаторів текстур, які потрібно створити. Другий параметр - покажчик на масив елементів типу `unsigned int`. Кількість елементів в масиві повинні збігатися з числом, зазначеним в якості першого параметра. Наприклад, наступний код створює десять імен текстур.</br>

```
unsigned int names[10];</br>
```

```
glGenTextures(10, names);</br>
```

Зберігати ідентифікатори текстур в масиві не завжди зручно. Такий спосіб підходить для зберігання фону заднього плану або типів стін: цегляна, кам'яна і т.п. У масиві зберігають ті елементи, між якими є щось спільне. У нашому випадку створені два різних ідентифікатора. Отже додайте наступний код в функцію `main`.

```
</br>
```

```
glGenTextures(1, &photo_tex);
```

```
</br>
```

```
glGenTextures(1, &space_tex);</br>
```

Тепер ми прив'язуємося до текстури фотографії, тобто робимо її активною. Для цього служить функція `glBindTexture`. Перший параметр повинен бути `GL_TEXTURE_2D` або `GL_TEXTURE_1D`. Він показує, з одновимірним або двовимірним зображенням будемо працювати. Всі приклади тут стосуються двовимірних текстур. Другий параметр `glBindTexture` - ідентифікатор, який ми створили вище за допомогою `glGenTextures`. Додайте виклик цієї функції в `main`.</br>

```
glBindTexture(GL_TEXTURE_2D, photo_tex);</br>
```

Тепер ми повинні створити саму текстуру в пам'яті. Масив байт в структурі `AUX_RGBImageRec` не є ще текстурою, тому що у текстури багато різних параметрів. Створивши текстуру, ми наділимо її певними властивостями.</br>

Серед параметрів текстури ви вкажете рівень деталізації, спосіб масштабування і зв'язування текстури з об'єктом. Рівень деталізації потрібен для накладення текстури на менші об'єкти, тобто коли площа на екрані менше розмірів зображення. Нульовий рівень деталізації відповідає вихідному зображенню розміром $2n \times 2m$, перший рівень - $2n-1 \times 2m-1$, k -ий рівень - $2n-k \times 2m-k$. Число рівнів відповідає $\min(n, m)$. Для створення текстури є дві функції `glTexImage [1/2]D` і `gluBuild [1/2]DMipmaps`.</br>

Основна відмінність в тому, що перша функція створює текстуру одного певного рівня деталізації і сприймає тільки зображення, розмір яких кратний ступеню двійки. Друга функція більш гнучка. Вона генерує текстури всіх рівнів деталізації. Також ця функція не вимагає, щоб розмір зображення був кратний ступеню

двійки. Вона сама стисне/розтягне зображення відповідним чином, хоча може виявитися, що і не цілком відповідним.</br>

Ми скористаємося функцією `glTexImage2D`. Перший параметр `target` зарезервованій і в поточній версії має дорівнювати `GL_TEXTURE_2D`. Другий `level` - рівень деталізації. Нам потрібно вхідне зображення, тому рівень деталізації - нуль.</br>

Третій параметр вказує кількість компонентів кольору. У нас зображення зберігається в форматі RGB, тому значення цього параметра дорівнює трьом.</br>

Четвертий і п'ятий параметри - ширина і висота зображення.</br>

Шостий `border` - ширина границі; у нас границі не буде, тому значення цього параметра - нуль.</br>

Далі, сьомий параметр - формат зберігання пікселів в масиві - `GL_RGB` і тип - `GL_UNSIGNED_BYTE`. І нарешті, восьмий параметр - покажчик на масив даних.</br>

Ще ви повинні викликати функцію `glPixelStorei` і задати, що вирівнювання в масиві даних йде по байту. Додайте наступний код в функцію `main`.</br>

```
glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
```

```
</br>
```

```
glTexImage2D(GL_TEXTURE_2D, 0, 3,</br>
```

```
photo_image->sizeX,</br>
```

```
photo_image->sizeY,</br>
```

```
0, GL_RGB, GL_UNSIGNED_BYTE,</br>
```

```
photo_image->data);</br>
```

Аналогічний результат можна отримати, вставивши виклик `gluBuild2DMipmaps` з параметрами, зазначеними нижче.</br>

```
gluBuild2DMipmaps(GL_TEXTURE_2D, 3,</br>
```

```
photo_image->sizeX,</br>
```

```
photo_image->sizeY,</br>
```

```
GL_RGB, GL_UNSIGNED_BYTE,</br>
```

```
photo_image->data);</br>
```

Тепер потрібно встановити параметри текстури. Для цього слугує функція `glTexParameter [if] (GLenum target, GLenum pname, GLenum param)`.</br>

Перший параметр приймає значення `GL_TEXTURE_1D` або `GL_TEXTURE_2D`. Другий - `pname` - визначає параметр текстури, який ви будете змінювати.</br>

І третій параметр - це встановлюване значення. Якщо ви скористалися `gluBuild2DMipmaps` замість `glTexImage2D`, то вам не треба встановлювати наступні параметри, бо вже сформовані текстури всіх рівнів деталізації, і OpenGL зможе підібрати текстуру потрібного рівня, якщо площа об'єкта не збігається з площею текстури. </br>

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);</br>
```

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);</br>
```

Ви вказали, що для зменшення і збільшення текстури використовується алгоритм `GL_NEAREST`. Це означає, що кольором пікселя об'єкта, на який накладається текстура, стає колір найближчого пікселя елемента текстури. Замість `GL_NEAREST` можна вказати `GL_LINEAR`, тобто колір елемента об'єкта буде обчислюватися як середнє арифметичне чотирьох елементів текстури.</br>

Є ще чотири алгоритму обчислення кольору елемента об'єкта. Їх можна встановлювати, коли ви створили текстуру з усіма рівнями деталізації, тому що застосовують алгоритми `GL_NEAREST` і `GL_LINEAR` до одного або двох найближчих рівнями деталізації.</br>

Ще можна встановити взаємодію текстури з об'єктом. Тут є два режими при використанні трьох компонентів кольору. </br>

Перший режим `GL_MODULATE` встановлений за замовчуванням, коли враховується колір об'єкта і колір текстури. Результуючий колір виходить перемноженням компонентів кольору текстури на компоненти кольору об'єкта. Скажімо, якщо колір текстури - (r, g, b), а колір об'єкта, на який вона накладається, - (r0, g0, b0), то результуючим кольором буде - (r*r0, g*g0, b*b0). </br>

У разі, якщо колір об'єкта чорний - (0,0,0), то ви не побачите на ньому текстуру, так як вона вся буде чорною. Другий режим взаємодії - коли колір об'єкта не враховується, результуючим кольором буде колір текстури. Ці параметри можна встановити наступним чином.</br>

```
glTexEnv(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE)</br>
```

```
glTexEnv(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_DECAL)</br>
```

За замовчуванням використовується режим `GL_MODULATE`.</br>

Тепер зробимо активну текстуру `space_tex` і повторити для неї те ж саме. На цьому закінчується створення текстури. Залишилося пов'язати координати текстури з координатами об'єкта.</br>

```

</div>
<div id="footer">
    <button id="button2" class="button" onclick="location.href='S2.html';"> Наступний крок</button>
</div>
</body>
</html>

```

Файл S2.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>2</title>
<link rel="stylesheet" type="text/css" href='CSS/title.css'/>
<link rel="shortcut icon" href="/img/favicon.ico" type="image/x-icon">
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.1/jquery.min.js"></script>
<script type="text/javascript" src="js/panel.js" ></script>
<link href="https://fonts.googleapis.com/css2?family=PT+Serif:ital,wght@0,700;1,400&display=swap"
rel="stylesheet">
</head>
<body id="fon">
<div id="pract">
<div id="left">
<ol>
<li>#include "stdafx.h"</li>
<li>#include windows.h </li>
<li>#include GL/gl.h </li>
<li>#include GL/glu.h </li>
<li>#include GL/glaux.h </li>
<li>unsigned int photo_tex;</br>
<li> AUX_RGBImageRec* photo_image;</li>
<li> unsigned int space_tex;</li>
<li> AUX_RGBImageRec* space_image;</li>
<li>void CALLBACK resize(int width, int height)</li>
<li>{ glViewport(0, 0, width, height);</li>
<li>glMatrixMode(GL_PROJECTION);</li>
<li>glLoadIdentity();</li>
<li>glOrtho(-5, 5, -5, 5, 2, 12);</li>
<li>gluLookAt(0, 0, 5, 0, 0, 0, 0, 1, 0);</li>
<li>glMatrixMode(GL_MODELVIEW);}</li>
<li>void CALLBACK display(void)</li>
<li>{ glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);</li>
<li>glEnable(GL_TEXTURE_2D);</li>
<li>glColor3d(1, 1, 1);</li>
<li>glBindTexture(GL_TEXTURE_2D, space_tex);</li>
<li>glBegin(GL_QUADS);</li>
<li>glTexCoord2d(0, 0); glVertex3d(-5, -5, -0.1);</li>
<li>glTexCoord2d(0, 1); glVertex3d(-5, 5, -0.1);</li>
<li>glTexCoord2d(1, 1); glVertex3d(5, 5, -0.1);</li>
<li>glTexCoord2d(1, 0); glVertex3d(5, -5, -0.1);</li>
<li>glEnd();</li>
<li>glBindTexture(GL_TEXTURE_2D, photo_tex);</li>
<li>glBegin(GL_QUADS);</li>
<li>glTexCoord2d(0, 0); glVertex2d(-5, -4);</li>

```

```

<li>glTexCoord2d(0, 1); glVertex2d(-5, 4);</li>
<li>glTexCoord2d(1, 1); glVertex2d(5, 4);</li>
<li>glTexCoord2d(1, 0); glVertex2d(5, -4);</li>
<li>glEnd();</li>
<li>glDisable(GL_TEXTURE_2D);</li>
<li>auxSwapBuffers();</li>
<li>void main()</li>
<li>{float pos[4] = {3,3,3,1};</li>
<li>float dir[3] = {-1,-1,-1};</li>
<li>GLfloat mat_specular[] = {1,1,1,1};</li>
<li>auxInitPosition( 50, 10, 400, 400);</li>
<li>auxInitDisplayMode( AUX_RGB | AUX_DEPTH | AUX_DOUBLE );</li>
<li> auxInitWindow(L"Tym" );</li>
<li>auxIdleFunc(display);</li>
<li>auxReshapeFunc(resize);</li>
<li>glEnable(GL_DEPTH_TEST);</li>
<li>glEnable(GL_COLOR_MATERIAL);</li>
<li> glEnable(GL_LIGHTING);</li>
<li> glEnable(GL_LIGHT0);</li>
<li> glLightfv(GL_LIGHT0, GL_POSITION, pos);</li>
<li> glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, dir);</li>
<li> glShadeModel(GL_SMOOTH);</li>
<li> photo_image = </li>
<li>auxDIBImageLoad(L"C:\\Users\\Documents\\Visual Studio 2010\\Projects\\photo.bmp");</li>
<li> space_image = </li>
<li>auxDIBImageLoad(L"C:\\Users\\Documents\\Visual Studio 2010\\Projects\\space.bmp");</li>
<li> glGenTextures(1, &photo_tex);</li>
<li> glBindTexture(GL_TEXTURE_2D, photo_tex);</li>
<li> glPixelStorei(GL_UNPACK_ALIGNMENT, 1);</li>
<li>glTexImage2D(GL_TEXTURE_2D, 0.5, 3,</li>
<li> photo_image->sizeX,</li>
<li> photo_image->sizeY,</li>
<li> 0, GL_RGB, GL_UNSIGNED_BYTE,</li>
<li> photo_image->data);</li>
<li>glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);</li>
<li>glGenTextures(1, &space_tex);</li>
<li> glBindTexture(GL_TEXTURE_2D, space_tex);</li>
<li> gluBuild2DMipmaps(GL_TEXTURE_2D, 3,</li>
<li> space_image->sizeX,</li>
<li> space_image->sizeY,</li>
<li> GL_RGB, GL_UNSIGNED_BYTE,</li>
<li> space_image->data);</li>
<li> auxMainLoop(display);}</li>
</ol>

```

</div>

<div class="panel">

Для подальшої роботи використали той же файл photo.bmp який вже додано до проекту.

А також додано space.bmp

Змінні photo_tex і space_tex слугуватимуть ідентифікаторами текстур.

Функція glGenTextures приймає на вхід два параметра. Перший вказує кількість імен-ідентифікаторів текстур, які потрібно створити. Другий параметр - покажчик на масив елементів типу unsigned int.

Для встановлення параметрів текстурислугує функція glTexParameter [if] (GLenum target, GLenum pname, GLenum param).

Перший параметр приймає значення `GL_TEXTURE_1D` або `GL_TEXTURE_2D`. Другий - `rname` - визначає параметр текстури, який ви будете змінювати. І третій параметр - це встановлюване значення. Якщо ви скористалися `gluBuild2DMipmaps` замість `glTexImage2D`, то вам не треба встановлювати наступні параметри, бо вже сформовані текстури всіх рівнів деталізації, і OpenGL зможе підібрати текстуру потрібного рівня, якщо площа об'єкта не збігається з площею текстури.

Функція `glTexCoord2d` зіставляє координати текстури вершин чотирикутника. Нижній лівий кут текстури має координати (0,0), а верхній правий - (1,1).

```

</span>
</div>
</div>
<div id="footer">
<button id="button2" class="button" onclick="location.href='S3.html';"> Наступний крок </button>
</div>
<b id="zap" class="spoiler-title"></b>
<div class="spoiler-body">
<div id="fin1">

</div></div>
</div>
<script type="text/javascript">
$(document).ready(function(){
$('.spoiler-body').hide();
$('.spoiler-title').click(function(){
$(this).toggleClass('opened').toggleClass('closed').next().slideToggle();
if($(this).hasClass('opened')) {
$(this).html('');
}
else {
$(this).html('');
}
});
});
</script>
</body>
</html>

```

Файл S3.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>3</title>
<link rel="stylesheet" type="text/css" href='CSS/title.css'/>
<link rel="shortcut icon" href="/img/favicon.ico" type="image/x-icon">
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.1/jquery.min.js"></script>
<script type="text/javascript" src="js/panel.js" ></script>
<link href="https://fonts.googleapis.com/css2?family=PT+Serif:ital,wght@0,700;1,400&display=swap"
rel="stylesheet">
</head>
<body id="fon">
<div id="pract">

<div id="left">
<ol>

```

```

<li>#include "stdafx.h"</li>
<li>#include windows.h </li>
<li>#include GL/gl.h </li>
<li>#include GL/glu.h </li>
<li>#include GL/glaux.h </li>
<li>unsigned int photo_tex;</li>
<li>AUX_RGBImageRec* photo_image;</li>
<li>unsigned int space_tex;</li>
<li>AUX_RGBImageRec* space_image;</li>
<li>void CALLBACK resize(int width, int height)</li>
<li>{   glVertex(0, 0, width, height);</li>
<li>glMatrixMode(GL_PROJECTION);</li>
<li>glLoadIdentity();</li>
<li>   glOrtho(-5, 5, -5, 5, 2, 12);</li>
<li>   gluLookAt(0, 0, 5, 0, 0, 0, 1, 0);</li>
<li>   glMatrixMode(GL_MODELVIEW);}</li>
<li>void CALLBACK display(void)</li>
<li>{   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);</li>
<li>   glEnable(GL_TEXTURE_2D);</li>
<li>   glColor3d(1, 1, 1);</li>
<li>   glBindTexture(GL_TEXTURE_2D, space_tex);</li>
<li>   glBegin(GL_QUADS);</li>
<li>   glVertex2d(0, 0); glVertex3d(-5, -5, -0.1);</li>
<li>   glVertex2d(0, 1); glVertex3d(-5, 5, -0.1);</li>
<li>   glVertex2d(1, 1); glVertex3d(5, 5, -0.1);</li>
<li>   glVertex2d(1, 0); glVertex3d(5, -5, -0.1);</li>
<li>   glEnd();</li>
<li>   glBindTexture(GL_TEXTURE_2D, photo_tex);</li>
<li>   glBegin(GL_QUADS);</li>
<li>   glVertex2d(0, 0); glVertex2d(-4, -4);</li>
<li>   glVertex2d(0, 2); glVertex2d(-4, 4);</li>
<li>   glVertex2d(3, 2); glVertex2d(4, 4);</li>
<li>   glVertex2d(3, 0); glVertex2d(4, -4);</li>
<li>   glEnd();</li>
<li>   glDisable(GL_TEXTURE_2D);</li>
<li>   auxSwapBuffers();}</li>
<li>void main()</li>
<li>{float pos[4] = {3,3,3,1};</li>
<li>float dir[3] = {-1,-1,-1};</li>
<li>GLfloat mat_specular[] = {1,1,1,1};</li>
<li>auxInitPosition( 50, 10, 400, 400);</li>
<li>auxInitDisplayMode( AUX_RGB | AUX_DEPTH | AUX_DOUBLE );</li>
<li>auxInitWindow(L"Tym" );</li>
<li>auxIdleFunc(display);</li>
<li>auxReshapeFunc(resize);</li>
<li>glEnable(GL_DEPTH_TEST);</li>
<li>glEnable(GL_COLOR_MATERIAL);</li>
<li>glEnable(GL_LIGHTING);</li>
<li>glEnable(GL_LIGHT0);</li>
<li>glLightfv(GL_LIGHT0, GL_POSITION, pos);</li>
<li>glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, dir);</li>
<li>glShadeModel(GL_SMOOTH);</li>
<li>photo_image = </li>
<li>auxDIBImageLoad(L"C:\\Users\\Documents\\Visual Studio 2010\\Projects\\photo.bmp");</li>
<li>space_image = </li>
<li>auxDIBImageLoad(L"C:\\Users\\Documents\\Visual Studio 2010\\Projects\\space.bmp");</li>
<li>glGenTextures(1, &photo_tex);</li>
<li>glBindTexture(GL_TEXTURE_2D, photo_tex);</li>

```

```

<li>    gluBuild2DMipmaps(GL_TEXTURE_2D, 3,</li>
<li>    photo_image->sizeX,</li>
<li>    photo_image->sizeY,</li>
<li>    GL_RGB, GL_UNSIGNED_BYTE,</li>
<li>    photo_image->data);</li>
<li>    glGenTextures(1, &space_tex);</li>
<li>    glBindTexture(GL_TEXTURE_2D, space_tex);</li>
<li>    gluBuild2DMipmaps(GL_TEXTURE_2D, 3,</li>
<li>    space_image->sizeX,</li>
<li>    space_image->sizeY,</li>
<li>    GL_RGB, GL_UNSIGNED_BYTE,</li>
<li>    space_image->data);</li>
<li>    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);</li>
<li>    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);</li>
<li>    auxMainLoop(display);}</li>
</ol>
</div>
<div class="panel">
<a class="handle" href="#">1</a>
<span>
<div>

```

<p>Розмножити текстуру на площині не складає великих труднощів. Відредагуємо програму з попереднього завдання.

Для того, щоб мати можливість повторювати текстуру, потрібно встановити параметр GL_REPEAT для її S і T координат.

S-координата текстури - це горизонтальна координата, T-координата - вертикальна. Друге значення параметра, яке може бути встановлено для координат, - GL_CLAMP. Він гарантує, що текстура НЕ буде розмножена. За умовчанням встановлено GL_REPEAT. У функцію main додайте наступні рядки:</p>

<p>Функція glTexCoord, як вже було зазначено вище, прив'язує координати текстури до вершин об'єкта.

Лівий нижній кут текстури має координату (0,0), а правий верхній - (1,1).

Якщо ви вказуєте в якості прив'язки значення, яке більше одиниці, то текстура повторюється.

У нашому прикладі, координату (0,0) текстури ми прив'язали до лівої нижньої вершини площини з координатою (-4, -4), а координату (3,2) текстури - до правої верхньої вершини (4,4).

Тим самим, ми отримали розмноження текстури по горизонталі три рази і по вертикалі два рази.

Інші дві вершини ми зв'язали відповідним чином. Якщо там вказати не ті числа, то зображення нахилиться.</p>

```

</div>
</span>
</div>
</div>

```

```

<div id="footer">

```

```

<button id="button2" class="button" onclick="location.href='Practick.html';"> До тестування </button>

```

```

</div>

```

```

<b id="zap" class="spoiler-title"></b>

```

```

<div class="spoiler-body">

```

```

<div id="fin1">

```

```



```

```

</div></div>

```

```

</div>

```

```

<script type="text/javascript">

```

```

$(document).ready(function(){

```

```

$('.spoiler-body').hide();

```

```

$('.spoiler-title').click(function(){

```

```

    $(this).toggleClass('opened').toggleClass('closed').next().slideToggle();

```

```

    if($(this).hasClass('opened')) {

```

```

        $(this).html('');

```

```

    }

```

```

    else {
        $(this).html('');
    }
});
});
</script>
</body>
</html>

```

Файл Practick.html

```

<html>
<head>
<title>Практика</title>
<meta charset="utf-8">
<link rel="shortcut icon" href="/img/favicon.ico" type="image/x-icon">
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"
integrity="sha384-HSMxcRTRxnN+BdgoJdxbYKvThecOKuH5zCYotlSAcp1+c8xmyTe9GYg119a69psu"
crossorigin="anonymous">
<!-- Дополнительные стили (не обязательно) -->
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap-theme.min.css"
integrity="sha384-6pzBo3FDv/PJ8r2KRkGHifhEocL+1X2rVCTTtkUfGk7/0pbek5mMa1upzvWbrUbOZ"
crossorigin="anonymous">
<!-- Подключаем jQuery (необходим для Bootstrap JavaScript) -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<!-- Bootstrap JavaScript -->
<script src="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js" integrity="sha384-
aJ21OjIMXNL5UyII/XNwTMqvzeRMZH2w8c5cRVpzpU8Y5bApTppSuUkhZXN0VxHd"
crossorigin="anonymous"></script>
<link href="https://fonts.googleapis.com/css2?family=PT+Serif:ital,wght@0,700;1,400&display=swap"
rel="stylesheet">
<script type="text/javascript" src="../data/data.js"></script>
<script type="text/javascript" src="../data/js.js"></script>
<style>
body {
display: -ms-flexbox;
display: -webkit-box;
display: flex;
-ms-flex-align: center;
-ms-flex-pack: center;
-webkit-box-align: center;
align-items: center;
-webkit-box-pack: center;
justify-content: center;
padding-top: 40px;
padding-bottom: 40px;
font-family: 'PT Serif', serif;
background: url(../img/sumdu.jpg) no-repeat center top fixed;
-webkit-background-size: cover;
-moz-background-size: cover;
-o-background-size: cover;
background-size: cover;
}
p, span, input, td {

```

```

font-size:24px;
color: white;
font-size: 25px;
}
#footer1 { /* Нижний блок */
  position: fixed; /* Фиксированное положение */
  padding: 10px; /* Поля вокруг текста */
  color: white; /* Цвет текста */
  bottom: 20px;
  right: 2%;
  height: 4%;
  font-family: 'PT Serif', serif;
}
button{
  font-size:18px;
  font-family: 'PT Serif', serif;
}
.w {
  background-color: white;
  color: black;
  text-align: center;
}
#ot {
  color:blue;
}
#B1 {
  position: fixed;
  right: 1%;
  bottom: 1%;
}
.button {
display: inline-block; /* Строчно-блочный элемент */
  padding: 5px 20px; /* Добавляем поля */
  text-decoration: none; /* Убираем подчёркивание у ссылки */
  cursor: pointer; /* Курсор в виде руки */
  background: #5A77AA; /* Фон для браузеров, не поддерживающих градиент */
  /* Градиент */
  border-radius: 10px; /* Скругляем уголки */
  font: 15px/1 Arial, sans-serif; /* Рубленный шрифт */
  color: white; /* Цвет текста и ссылки */
font-size: 15pt;
font-family: 'PT Serif', serif;
}
#p {
margin-top:10%;
}
.fon{
border: 2px solid black;
}
</style>
</head>
<body>
<div style=" position: absolute; margin-left: auto;margin-right: auto;width: 1100px; height: auto; text-align:center;border:5px inset #384867; top: 10%; background-color:#5A77AA; ">
<p> "<span id="tema"></span>"</p>
<div id="vopros1" style="display:block;">
<p id="p"><span id="v_1"></span> <br></br>
<input style="margin-top: 5%;" class="w" type="text" id="z_1"></p>

```

```

</div>
<div id="vopros2" style="display:none;">
<p id="p"><span id="v_2"></span></br>
<input style="margin-top: 5%;" class="w" type="text" id="z_2"></p>
</div>
<div id="vopros3" style="display:none;">
<p id="p"><span id="v_3"></span> </br></br>
<input style="margin-top: 5%;" class="w" type="text" id="z_3"></p>
</div>
<div id="vopros4" style="display:none;">
<p id="p"><span id="v_4"></span> </br></br>
<input style="margin-top: 5%;" class="w" type="text" id="z_4"></p>
</div>
<div id="vopros5" style="display:none;">
<p id="p"><span id="v_5"></span> </br></br>
<input style="margin-top: 5%;" class="w" type="text" id="z_5"></p>
</div>
<div id="vopros6" style="display:none;">
<p id="p"><span id="v_6"></span></br></br>
<input style="margin-top: 5%;" class="w" type="text" id="z_6"></p>
</div>
<div id="vopros7" style="display:none;">
<p id="p"><span id="v_7"></span> </br></br>
<input style="margin-top: 5%;" class="w" type="text" id="z_7"></p>
</div>
<div id="vopros8" style="display:none;">
<p id="p"><span id="v_8"></span> </br></br>
<input style="margin-top: 5%;" class="w" type="text" id="z_8"></p>
</div>
<div id="vopros9" style="display:none;">
<p id="p"><span id="v_9"></span> </br></br>
<input style="margin-top: 5%;" class="w" type="text" id="z_9"></p>
</div>
<div id="vopros10" style="display:none;">
<p id="p"><span id="v_10"></span> </br></br>
<input style="margin-top: 5%;" class="w" type="text" id="z_10"></p>
</div>
<button id="kn_sl" class="btn btn-lg btn-primary btn-block" style="margin-left:auto;margin-
right:auto;width:250px;" onclick="sled_vopr();">Наступне питання</button>
<button id="kn_pr" class="btn btn-lg btn-primary btn-block" style="margin-left:auto;margin-
right:auto;display:none;width:250px;" onclick="proverit();">Перевірка</button>
<div id="rezultat"></div>
<div id="footer1">
<button id="button2" class="button" onclick="location.href='index.html';"> Завершити </button>
</div>
</div>
</body>
</html>

```

Файл NewCod.html

```

<!DOCTYPE html>
<html>
<head>

```

```

<meta charset="utf-8">
<title>Shifr</title>
<link rel="stylesheet" type="text/css" href='CSS/title.css'/>
<link rel="shortcut icon" href="/img/favicon.ico" type="image/x-icon">
</head>
<body>
<div style="text-align:center;">
<div id="vopros1" style="display:block;">
<p>Яку відповідь потрібно зашифрувати?</p>
<p><input type="text" id="z_1"></p>
</div>
<button id="kn_sl" onclick="zakodirovat();">Зашифрувати</button>
<div id="rezultat"></div>
</div>
<script>
function zakodirovat(){
otv_uch_1 = document.getElementById('z_1').value;
otv_uch_1 = kodirov(otv_uch_1);
document.getElementById('rezultat').innerHTML = "<br><br><span id='ot'>Зашифрована відповідь:
"+otv_uch_1+"</span>";
}
function kodirov(stroka) {
var b64ch = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/' ;
var b64enc = "";
var ch1, ch2, ch3;
var en1, en2, en3, en4;
for (var i=0; i<stroka.length;) {
ch1 = stroka.charCodeAt(i++);
ch2 = stroka.charCodeAt(i++);
ch3 = stroka.charCodeAt(i++);
en1 = ch1 >> 2;
en2 = ((ch1 & 3) << 4) | (ch2 >> 4);
en3 = isNaN(ch2) ? 64:(((ch2 & 15) << 2) | (ch3 >> 6));
en4 = isNaN(ch3) ? 64:(ch3 & 63);
b64enc += b64ch.charAt(en1) + b64ch.charAt(en2) +
b64ch.charAt(en3) + b64ch.charAt(en4);
}
return b64enc;
}
</script>
</body>
</html>

```

Файл title.css

```

body {
margin: 0;
font-family: glyphicons, sans-serif;
font-size: 20px;
background: #5A77AA;
}
header {
position: absolute;
font-family: 'PT Serif, serif;

```

```

width: 100%;
color: white;
background-color: #384867;
height: 80px;
}
h1 {
position: absolute;
font-family: 'PT Serif', serif;
margin-top: 14px;
text-align: center;
margin-left: 32%;
}
ol li {
text-align: left;
font-size: 10pt;
font-family: "Console", sans-serif;
}
p {
margin-bottom: -8px;
left: 5px;
font-family: "Console", sans-serif;
}
input {
border: 2px solid #000080; /* Параметры рамки */
font-size: 14px; /* Размер текста */
}
table {
frame: border;
border: 2px; /* Параметры рамки вокруг */
bordercolor: #000000;
width: 100%;
text-align: center;
}
tr td {
padding: 3px; /* Поля вокруг содержимого таблицы */
border: 4px solid #384867; /* Параметры рамки */
text-align: center;
width: 40%;
}
@font-face {
font-family: "Console";
src: url("fonts/Console/Console.ttf") format("truetype");
font-style: normal;
font-weight: normal;
}
@font-face {
font-family: glyphicons; /* Гарнитура шрифта */
src: url("fonts/glyphicons-halflings-regular.eot"); /* Для IE5-8 */
}
#i {
position: fixed;
bottom: 4%;
}
.panel {
margin: 0;
padding: 50px;
width: 400px;
height: 300px;

```



```

background: #524676;
border: 5px solid white;
font-family: "Console",sans-serif;
font-size: 10pt;
color: white;
}
.button {
display: inline-block; /* Строчно-блочный элемент */
padding: 5px 20px; /* Добавляем поля */
text-decoration: none; /* Убираем подчёркивание у ссылки */
cursor: pointer; /* Курсор в виде руки */
background: #5A77AA; /* Фон для браузеров, не поддерживающих градиент */

/* Градиент */
border-radius: 10px; /* Скругляем уголки */
font: 15px/1 Arial, sans-serif; /* Рубленый шрифт */
color: white; /* Цвет текста и ссылки */
font-size: 15pt;
}

.poster{
position:relative;
margin:100px auto;
background:#ff6600;
height:200px;
width:150px;
}
.descr{
display:none;
margin-left:-350px;
padding:10px;
margin-top:17px;
background:#f3f3f3;
height:200px;
-moz-box-shadow:0 5px 5px rgba(0,0,0,0.3);
-webkit-box-shadow:0 5px 5px rgba(0,0,0,0.3);
box-shadow:0 5px 5px rgba(0,0,0,0.3);
}
.poster:hover .descr{
display:block;
position:absolute;
top:120px;
z-index:9999;
width:400px;
}
.title{
font-size: 36px;
padding-top: 10px;
text-align: center;
}
.menu{
list-style: none;
position: absolute;
top: 10px;
right: 50px;
}
.round {
border-radius: 150px; /* Радиус скругления */

```

```

border: 3px solid ; /* Параметры рамки */
box-shadow: 0 0 10px #666; /* Параметры тени */
}
#ind {
text-align: center;
border: 2px solid #333; /* Параметры рамки */
}
#button1 {
display: block;
font-family: 'PT Serif', serif;
margin-left: auto;
margin-right: auto;
}
#button2 {
display: block;
font-family: 'PT Serif', serif;
margin-left: auto;
margin-right: 20px;
}
#content2 { /* Правая колонка */
position: fixed;
font-family: 'PT Serif', serif;
top: 15%;
right: 5%;
left: 5%;
bottom: 5%;
margin: 0px 20px 20px 20px;
text-align: center;
padding: 5px; /* Поля вокруг текста */
border: 10px inset #384867; /* Параметры рамки */
color: white;
background-color:#5A77AA;
}
#content1 { /* Правая колонка */
position: absolute;
top: 12%;
margin: 35px 35px 55px 35px;
text-align: center;
padding: 5px; /* Поля вокруг текста */
border: 0px solid #333; /* Параметры рамки */
color: white;
}
#r {
width: 20%;
}
#header { /* Верхний блок */
padding: 2px; /* Поля вокруг текста */
}
#B1 {
position: fixed;
right: 1%;
bottom: 1%;
}
#B2 {
position: fixed;
left: 1%;
bottom: 1%;
}
#fonn {
background: url(../img/sumdu.jpg) no-repeat center top fixed;
-webkit-background-size: cover;
-moz-background-size: cover;
-o-background-size: cover;
}

```

```

    background-size: cover;}}
#text1{
position: fixed;
margin-top: 50px;
    #footer { /* Нижний блок */
    position: fixed; /* Фиксированное положение */
left: 0;
bottom: 0%; /* Левый нижний угол */
padding: 10px; /* Поля вокруг текста */
color: white; /* Цвет текста */
width: 100%; /* Ширина слоя */
height: 4%;}
#footer1 { /* Нижний блок */
    position: fixed; /* Фиксированное положение */
left: 0;
bottom: 0%; /* Левый нижний угол */
padding: 10px; /* Поля вокруг текста */
color: white; /* Цвет текста */
width: 100%; /* Ширина слоя */}
#sidebar1 { /* Левая колонка */
float: left; /* Обтекание справа */
border: 5px solid #000000; /* Параметры рамки вокруг */
width: 20%; /* Ширина колонки */
padding: 5px; /* Поля вокруг текста */
margin: 10px 10px 20px 5px; /* Значения отступов */}
#fon{
background: url(..img/444.png) no-repeat ;
background-size: 100%;}
#pract {
position: absolute;
margin-top: 2%; /* Значения отступов */}
#img0 {
position: absolute;
margin: 20px 20px 20px 20px; }
#img1 {
float:left; /* Выравнивание по левому краю */
margin: 2px 2px 2px 0; /* Отступы вокруг картинки */}
margin-top: 15px;
border: 2px outset white ; }
#img0 {
position: absolute;
margin: 20px 20px 20px 20px; }
#left-Top {
position: fixed;
float: left;
width: 50%;
height: 20%;
padding: 2px;
top: 5%;
left: 2%;
text-align: left;
word-break: break-all; }
#zap {
position: fixed;
top: 35px;
left: 465px; }
#left { /* Левая колонка */
position: fixed;

```

```

overflow: scroll;
overflow-x: hidden;
background: white;
  float: left;
  width: 63%;
  height: 64%;
border: 2px inset white;
top: 11%;
left: 3%;
padding: 2px;
  word-break: break-all;
text-align: 15px;}
#div1{
display: none;}
#fin1{
position: fixed;
top: 125px;
right: 90px;    }
#mydiv {
  position: absolute;
  z-index: 9;
  border: 0px solid #d3d3d3;
  text-align: center;    }
#mydivheader {
position: fixed;
  padding: 10px;
  cursor: move;
left: 500px;    }
#run {
position: fixed;    }

```

Файл data.js

```

var test= "Текстування об'єктів";
var peremeshat_voprosi = "да";
var vopros_1 = "Укажіть значення змінної type для цілого знакового 32-bit типу даних?</br> </br>
*Використовуйте великі літери при написанні відповіді";
var vopros_2 = "Оголосіть глобальну змінну* image";
var vopros_3 = "Функція для завдання початкової точки виведення?</br></br>*Вкажіть тільки функцію, без
параметрів";
  var vopros_4 = "Укажіть значення змінної type для типу даних з плаваючою точкою? </br> </br>
*Використовуйте великі літери при написанні відповіді";
var vopros_5 = "Укажіть значення змінної type для цілого, знакового 8-bit типу даних? </br> </br>
*Використовуйте великі літери при написанні відповіді ";
  var vopros_6 = "Укажіть яка прив'язує координати текстури до вершин об'єкта? </br> </br> *Вкажіть
тільки функцію, без параметрів";

var vopros_7 = "Укажіть яка приймає в якості першого аргументу константу, яка визначає настроюваний параметр,
а другий аргумент визначає її значення?</br> </br> *Вкажіть тільки функцію, без параметрів";
var vopros_8 = "Який параметр потрібно встановити щоб повторити текстуру? </br> </br> *Використовуйте великі
літери при написанні відповіді ";
var vopros_9 = "Вкажіть яку бібліотеку не не підключено? opengl32.lib; glu32.lib;</br> </br> *Не забувайте
ставити крапку з комою в кінці.";

```

```

var vopros_10 = "Режим вирівнювання (GL_UNPACK_ALIGNMENT) може приймати значення? </br> </br>
*Вкажіть через кому без відступів чотири числових параметра";
var pr_otv_zadachi_1 = "R0xfSU5U";
var pr_otv_zadachi_2 = "QVVYX1JHQkltYWdlUmVjCogaW1hZ2U7";
var pr_otv_zadachi_3 = "Z2xSYXN0ZXJQb3MyZA==";
var pr_otv_zadachi_4 = "R0xfRkxPQVQ=";
var pr_otv_zadachi_5 = "R0xfQlIURQ==";
var pr_otv_zadachi_6 = "Z2xUZXXhDb29yZA==";
var pr_otv_zadachi_7 = "Z2xQaXh1bFN0b3Jl";
var pr_otv_zadachi_8 = "R0xfUkVQRUFU";
var pr_otv_zadachi_9 = "Z2xdXgubGliOw==";
var pr_otv_zadachi_10 = "MSwyLDQsOA==";

```

Файл js.js

```

document.addEventListener('DOMContentLoaded', function(){
    document.getElementById('v_1').innerHTML = vopros_1;
    document.getElementById('v_2').innerHTML = vopros_2;
    document.getElementById('v_3').innerHTML = vopros_3;
    document.getElementById('v_4').innerHTML = vopros_4;
    document.getElementById('v_5').innerHTML = vopros_5;
    document.getElementById('v_6').innerHTML = vopros_6;
    document.getElementById('v_7').innerHTML = vopros_7;
    document.getElementById('v_8').innerHTML = vopros_8;
    document.getElementById('v_9').innerHTML = vopros_9;
    document.getElementById('v_10').innerHTML = vopros_10;
    document.getElementById('tema').innerHTML = test;

    voprosi
    [[vopros_1,pr_otv_zadachi_1],[vopros_2,pr_otv_zadachi_2],[vopros_3,pr_otv_zadachi_3],[vopros_4,pr_otv_zadachi_4]
    ,[vopros_5,pr_otv_zadachi_5],[vopros_6,pr_otv_zadachi_6],[vopros_7,pr_otv_zadachi_7],[vopros_8,pr_otv_zadachi_8]
    ,[vopros_9,pr_otv_zadachi_9],[vopros_10,pr_otv_zadachi_10]];
    if(peremeshat_voprosi == "да"){
        voprosi = peremeshat_vse_voprosi(voprosi);
    }
    document.getElementById('v_1').innerHTML = voprosi[0][0];
    document.getElementById('v_2').innerHTML = voprosi[1][0];
    document.getElementById('v_3').innerHTML = voprosi[2][0];
    document.getElementById('v_4').innerHTML = voprosi[3][0];
    document.getElementById('v_5').innerHTML = voprosi[4][0];
    document.getElementById('v_6').innerHTML = voprosi[5][0];
    document.getElementById('v_7').innerHTML = voprosi[6][0];
    document.getElementById('v_8').innerHTML = voprosi[7][0];
    document.getElementById('v_9').innerHTML = voprosi[8][0];
    document.getElementById('v_10').innerHTML = voprosi[9][0];
    document.getElementById('tema').innerHTML = test;
});

function sled_vopr(){
    if(document.getElementById('vopros9').style.display == "block"){
        document.getElementById('vopros9').style.display = "none";
        document.getElementById('vopros10').style.display = "block";
        document.getElementById('kn_sl').style.display = "none";
        document.getElementById('kn_pr').style.display = "block"; }
    if(document.getElementById('vopros8').style.display == "block"){

```

```

    document.getElementById('vopros8').style.display = "none";
    document.getElementById('vopros9').style.display = "block";}
if(document.getElementById('vopros7').style.display == "block"){
    document.getElementById('vopros7').style.display = "none";
    document.getElementById('vopros8').style.display = "block";}
if(document.getElementById('vopros6').style.display == "block"){
    document.getElementById('vopros6').style.display = "none";
    document.getElementById('vopros7').style.display = "block";}
if(document.getElementById('vopros5').style.display == "block"){
    document.getElementById('vopros5').style.display = "none";
    document.getElementById('vopros6').style.display = "block";}
if(document.getElementById('vopros4').style.display == "block"){
    document.getElementById('vopros4').style.display = "none";
    document.getElementById('vopros5').style.display = "block";}
if(document.getElementById('vopros3').style.display == "block"){
    document.getElementById('vopros3').style.display = "none";
    document.getElementById('vopros4').style.display = "block"; }
if(document.getElementById('vopros2').style.display == "block"){
    document.getElementById('vopros2').style.display = "none";
    document.getElementById('vopros3').style.display = "block";}
if(document.getElementById('vopros1').style.display == "block"){
    document.getElementById('vopros1').style.display = "none";
    document.getElementById('vopros2').style.display = "block"; }}
function proverit(){
otv_uch_1 = document.getElementById('z_1').value;
otv_ucgenika_1 = otv_uch_1;
otv_uch_1 = kodirov(otv_uch_1);
otv_uch_2 = document.getElementById('z_2').value;
otv_ucgenika_2 = otv_uch_2;
otv_uch_2 = kodirov(otv_uch_2);
otv_uch_3 = document.getElementById('z_3').value;
otv_ucgenika_3 = otv_uch_3;
otv_uch_3 = kodirov(otv_uch_3);
otv_uch_4 = document.getElementById('z_4').value;
otv_ucgenika_4 = otv_uch_4;
otv_uch_4 = kodirov(otv_uch_4);
otv_uch_5 = document.getElementById('z_5').value;
otv_ucgenika_5 = otv_uch_5;
otv_uch_5 = kodirov(otv_uch_5);
otv_uch_6 = document.getElementById('z_6').value;
otv_ucgenika_6 = otv_uch_6;
otv_uch_6 = kodirov(otv_uch_6);
otv_uch_7 = document.getElementById('z_7').value;
otv_ucgenika_7 = otv_uch_7;
otv_uch_7 = kodirov(otv_uch_7);
otv_uch_8 = document.getElementById('z_8').value;
otv_ucgenika_8 = otv_uch_8;
otv_uch_8 = kodirov(otv_uch_8);
otv_uch_9 = document.getElementById('z_9').value;
otv_ucgenika_9 = otv_uch_9;
otv_uch_9 = kodirov(otv_uch_9);
otv_uch_10 = document.getElementById('z_10').value;
otv_ucgenika_10 = otv_uch_10;
otv_uch_10 = kodirov(otv_uch_10);
ball = 0;
document.getElementById('vopros10').style.display = "none";
document.getElementById('kn_pr').style.display = "none";
otveti = "<table class='table'><tr class='table-active'><td>№</td><td>Вопрос</td><td>Ответ</td></tr>";

```

```

if(otv_uch_1 == voprosi[0][1]){
    ball +=1;
    otveti += "<tr class='table-success'><td>2</td><td>"+voprosi[0][0]+"<td> Відповідь вірна!";
} else {
    otveti += "<tr class='table-danger'><td>1</td><td>"+voprosi[0][0]+"</td><td>Ви відповіли не вірно. Ваша
відповідь: "+otv_ucgenika_1+"</td></tr>"; }
if(otv_uch_2 == voprosi[1][1]){
    ball +=1;
    otveti += "<tr class='table-success'><td>2</td><td>"+voprosi[1][0]+"<td> Відповідь вірна!";
} else {
    otveti += "<tr class='table-danger'><td>2</td><td>"+voprosi[1][0]+"</td><td>Ви відповіли не вірно. Ваша
відповідь: "+otv_ucgenika_2+"</td></tr>"; }
if(otv_uch_3 == voprosi[2][1]){
    ball +=1;
    otveti += "<tr class='table-success'><td>3</td><td>"+voprosi[2][0]+"<td> Відповідь вірна!";
} else {
    otveti += "<tr class='table-danger'><td>3</td><td>"+voprosi[2][0]+"</td><td>Ви відповіли не вірно. Ваша
відповідь: "+otv_ucgenika_3+"</td></tr>"; }
if(otv_uch_4 == voprosi[3][1]){
    ball +=1;
    otveti += "<tr class='table-success'><td>4</td><td>"+voprosi[3][0]+"<td> Відповідь вірна!";
} else {
    otveti += "<tr class='table-danger'><td>4</td><td>"+voprosi[3][0]+"</td><td>Ви відповіли не вірно. Ваша
відповідь: "+otv_ucgenika_4+"</td></tr>"; }
if(otv_uch_5 == voprosi[4][1]){
    ball +=1;
    otveti += "<tr class='table-success'><td>5</td><td>"+voprosi[4][0]+"<td> Відповідь вірна!";
} else {
    otveti += "<tr class='table-danger'><td>5</td><td>"+voprosi[4][0]+"</td><td>Ви відповіли не вірно. Ваша
відповідь: "+otv_ucgenika_5+"</td></tr>"; }
if(otv_uch_6 == voprosi[5][1]){
    ball +=1;
    otveti += "<tr class='table-success'><td>6</td><td>"+voprosi[5][0]+"<td> Відповідь вірна!";
} else {
    otveti += "<tr class='table-danger'><td>6</td><td>"+voprosi[5][0]+"</td><td>Ви відповіли не вірно. Ваша
відповідь: "+otv_ucgenika_6+"</td></tr>"; }
if(otv_uch_7 == voprosi[6][1]){
    ball +=1;
    otveti += "<tr class='table-success'><td>7</td><td>"+voprosi[6][0]+"<td> Відповідь вірна!";
} else {
    otveti += "<tr class='table-danger'><td>7</td><td>"+voprosi[6][0]+"</td><td>Ви відповіли не вірно. Ваша
відповідь: "+otv_ucgenika_7+"</td></tr>"; }
if(otv_uch_8 == voprosi[7][1]){
    ball +=1;
    otveti += "<tr class='table-success'><td>8</td><td>"+voprosi[7][0]+"<td> Відповідь вірна!";
} else {
    otveti += "<tr class='table-danger'><td>8</td><td>"+voprosi[7][0]+"</td><td>Ви відповіли не вірно.
Ваша відповідь: "+otv_ucgenika_8+"</td></tr>"; }
if(otv_uch_9 == voprosi[8][1]){
    ball +=1;
    otveti += "<tr class='table-success'><td>9</td><td>"+voprosi[8][0]+"<td> Відповідь вірна!";
} else {
    otveti += "<tr class='table-danger'><td>9</td><td>"+voprosi[8][0]+"</td><td>Ви відповіли не вірно. Ваша
відповідь: "+otv_ucgenika_9+"</td></tr>"; }
if(otv_uch_10 == voprosi[9][1]){
    ball +=1;
    otveti += "<tr class='table-success'><td>10</td><td>"+voprosi[9][0]+"<td> Відповідь вірна!";
} else {

```

```

    otveti += "<tr class='table-danger'><td>10</td><td>" + voprosi[9][0] + "</td><td>Ви відповіли не вірно.  

    Ваша відповідь: "+otv_ucgenika_10+"</td></tr>";}
    vsego_zadach = 10;
    procent_vip = ball/vsego_zadach * 100;
    procent_vip = procent_vip.toFixed();
    document.getElementById('rezultat').innerHTML = "<p style='font-weight:bold;'></p><span id='ot'>Завдання  

    виконано на "+procent_vip+"%.</span><br><div class='progress'><div class='progress-bar' role='progressbar'  

    style='width: "+procent_vip+"%' aria-valuenow='"+procent_vip+"'' aria-valuemin='0' aria-  

    valuemax='100'></div></div><br>" + otveti;
    document.getElementById('tema2').innerHTML = test;}
    function kodirov(stroka) {
        var b64ch = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/-';
        var b64enc = "";
        var a, b, c;
        var A, B, C, en4;
        for (var i=0; i<stroka.length; i) {
            a = stroka.charCodeAt(i++);
            b = stroka.charCodeAt(i++);
            c = stroka.charCodeAt(i++);
            A = a >> 2;
            B = ((a & 3) << 4) | (b >> 4);
            C = isNaN(b) ? 64:(((b & 15) << 2) | (c >> 6));
            en4 = isNaN(c) ? 64:(c & 63);
            b64enc += b64ch.charAt(A) + b64ch.charAt(B) +  

            b64ch.charAt(C) + b64ch.charAt(en4); }
        return b64enc; }
    function peremeshat_vse_voprosi(V) {
        for (var i = V.length - 1; i > 0; i--) {
            var j = Math.floor(Math.random() * (i + 1));
            var temp = V[i];
            V[i] = V[j];
            V[j] = temp; }
        return V;}

```

Файл plane.js

```

$(function(){
    $('.panel').tabSlideOut({ //КЛАСС ПАНЕЛИ
        tabHandle: '.handle', //КЛАСС КНОПКИ
        pathToTabImage: './img/imgbutton.jpg', //ПУТЬ К ИЗОБРАЖЕНИЮ КНОПКИ
        imageHeight: '300px', //ВЫСОТА КНОПКИ
        imageWidth: '100px', //ШИРИНА КНОПКИ
        tabLocation: 'left', //РАСПОЛОЖЕНИЕ ПАНЕЛИ ТОР - ВЫДВИГАЕТСЯ СВЕРХУ,  

        RIGHT - ВЫДВИГАЕТСЯ СПРАВА, ВОТТОМ - ВЫДВИГАЕТСЯ СНИЗУ, LEFT - ВЫДВИГАЕТСЯ СЛЕВА
        speed: 300, //СКОРОСТЬ АНИМАЦИИ
        action: 'click', //МЕТОД ПОКАЗА SLICK - ВЫДВИГАЕТСЯ ПО КЛИКУ НА КНОПКУ,  

        HOVER - ВЫДВИГАЕТСЯ ПРИ НАВЕДЕНИИ КУРСОРА
        topPos: '61px', //ОТСТУП СВЕРХУ
        fixedPosition: true //ПОЗИЦИОНИРОВАНИЕ БЛОКА FALSE - POSITION: ABSOLUTE,  

        TRUE - POSITION: FIXED });});
    (function($){
        $.fn.tabSlideOut = function(callerSettings) {
            var settings = $.extend({
                }, callerSettings||{});

```



```

settings.tabHandle = $(settings.tabHandle);
var obj = this;
if (settings.fixedPosition === true) {
    settings.positioning = 'fixed';
} else {
    settings.positioning = 'absolute';}
//IE6 DOESN'T DO WELL WITH THE FIXED OPTION
if (document.all && !window.opera && !window.XMLHttpRequest) {
    settings.positioning = 'absolute';}
//SET INITIAL TABHANDLE CSS
settings.tabHandle.css({
    'display': 'block',
    'width': settings.imageWidth,
    'height': settings.imageHeight,
    'textIndent': '-99999px',
    'background': 'url('+settings.pathToTabImage+') no-repeat',
    'outline': 'none',
    'position': 'absolute'});
obj.css({
    'line-height': '1',
    'position': settings.positioning});
var properties = {
    containerWidth: parseInt(obj.outerWidth(), 10) + 'px',
    containerHeight: parseInt(obj.outerHeight(), 10) + 'px',
    tabWidth: parseInt(settings.tabHandle.outerWidth(), 10) + 'px',
    tabHeight: parseInt(settings.tabHandle.outerHeight(), 10) + 'px'};
//SET CALCULATED CSS
if(settings.tabLocation === 'top' || settings.tabLocation === 'bottom') {
    obj.css({'left': settings.leftPos});
    settings.tabHandle.css({'right': 0});}
    if(settings.tabLocation === 'top') {
        obj.css({'top': '-' + properties.containerHeight});
        settings.tabHandle.css({'bottom': '-' + properties.tabHeight});}
if(settings.tabLocation === 'bottom') {
    obj.css({'bottom': '-' + properties.containerHeight, 'position': 'fixed'});
    settings.tabHandle.css({'top': '-' + properties.tabHeight});}
if(settings.tabLocation === 'left' || settings.tabLocation === 'right') {
    obj.css({
        'height': properties.containerHeight,
        'top': settings.topPos});
    settings.tabHandle.css({'top': 0}); }
if(settings.tabLocation === 'left') {
    obj.css({'left': '-' + properties.containerWidth});
    settings.tabHandle.css({'right': '-' + properties.tabWidth});}
if(settings.tabLocation === 'right') {
    obj.css({'right': '-' + properties.containerWidth});
    settings.tabHandle.css({'left': '-' + properties.tabWidth});
    $('html').css('overflow-x', 'hidden');}
//FUNCTIONS FOR ANIMATION EVENTS
settings.tabHandle.click(function(event){
    event.preventDefault();});
var slideIn = function() {
    if (settings.tabLocation === 'top') {
        obj.animate({top:'-' + properties.containerHeight}, settings.speed).removeClass('open');
    } else if (settings.tabLocation === 'left') {
        obj.animate({left: '-' + properties.containerWidth}, settings.speed).removeClass('open');
    } else if (settings.tabLocation === 'right') {
        obj.animate({right: '-' + properties.containerWidth}, settings.speed).removeClass('open');
    }
}

```

```

    } else if (settings.tabLocation === 'bottom') {
      obj.animate({bottom: '-' + properties.containerHeight}, settings.speed).removeClass('open'); } }];
var slideOut = function() {
  if (settings.tabLocation == 'top') {
    obj.animate({top: '-3px'}, settings.speed).addClass('open');
  } else if (settings.tabLocation == 'left') {
    obj.animate({left: '-3px'}, settings.speed).addClass('open');
  } else if (settings.tabLocation == 'right') {
    obj.animate({right: '-3px'}, settings.speed).addClass('open');
  } else if (settings.tabLocation == 'bottom') {
    obj.animate({bottom: '-3px'}, settings.speed).addClass('open');}];

var clickScreenToClose = function() {
  obj.click(function(event){
    event.stopPropagation();});
$(document).click(function(){
  slideIn();}); };
var clickAction = function(){
  settings.tabHandle.click(function(event){
    if (obj.hasClass('open')) {
      slideIn();
    } else {
      slideOut();} });
  clickScreenToClose();};
var hoverAction = function(){
  obj.hover(
    function(){
      slideOut();},
    function(){
      slideIn();});
  settings.tabHandle.click(function(event){
    if (obj.hasClass('open')) {
      slideIn();});});
  clickScreenToClose();};
//CHOOSE WHICH TYPE OF ACTION TO BIND
if (settings.action === 'click') {
  clickAction();}
if (settings.action === 'hover') {
  hoverAction();});
})(jQuery);
window.onload= function() {
  document.getElementById('toggler').onclick = function() {
    openbox('box', this);
    return false;});};
function openbox(id, toggler) {
  var div = document.getElementById(id);
  if(div.style.display == 'block') {
    div.style.display = 'none';
    toggler.innerHTML = 'Открыть';}
  else {
    div.style.display = 'block';
    toggler.innerHTML = 'Закрыть';}}

```