Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

РОЗРОБКА IOS-ДОДАТКУ З ВИКОРИСТАННЯМ 3D-ОБ'ЄКТІВ

Текстова частина до курсової роботи
за спеціальністю «Комп'ютерні науки»

Керівник курсової роботи

ст. викладач Борозенний С. О.

_____
(*підпис*)

"____" _____ 2023 р.

Виконала студентка 3 р. н.

факультету інформатики

Іваненко В. А.

Київ - 2023

Зміст

Анотація.....3
Вступ.....4
1. Теоретичні аспекти розробки iOS-додатків та використання 3D-об'єктів 6
1.1. Огляд платформи iOS та її особливостей.....6
1.2. Інструменти та мови програмування для розробки iOS-додатків.....6
1.3. Принципи роботи з 3D-об'єктами в iOS.....7
1.4. Бібліотеки та інструменти для роботи з 3D-об'єктами в iOS.....8
2. Проектування та розробка iOS-додатка з використанням 3D-об'єктів.....9
2.1 Визначення вимог до додатка.....9
2.2 Проектування інтерфейсу користувача.....10
2.3 Вибір технологій для розробки.....11
2.4 Обґрунтування вибору формату .gltf.....12
2.5 Шаблон проектування.....13
3. Реалізація додатка.....14
3.1 Основні компоненти.....14
3.2 Графічний інтерфейс користувача.....16
3.3. Інтеграція 3D-моделей органів людини.....19
Висновки.....20
Список літератури.....22

Анотація

У даній курсовій роботі представлено процес розробки навчального iOS-додатка для медиків та школярів, що використовує 3D-моделі органів людини. Програма, створена за допомогою мови програмування Swift 5 в розробницькому середовищі Xcode 14.3, демонструє можливості SwiftUI, Swift Package Manager та бібліотеки Model3DView для створення високоякісних тривимірних візуалізацій.

Основними функціями додатка є відображення деталізованих 3D-моделей різних органів людини, можливість пошуку за системами органів та зберігання моделей в закладки для швидкого доступу.

Додаток розроблено з метою підвищення рівня освіти та розширення кругозору медичних працівників шляхом дослідження анатомії людини в інтерактивному та візуально привабливому форматі. Він також може бути корисним для школярів, що цікавляться біологією або розглядають можливість медичної кар'єри.

Ключові слова: iOS, Xcode, Swift, Apple, JSON, iPhone, 3D-об’єкти, мобільний додаток, медицина.

Вступ

Обрання теми "Розробка iOS-додатку з використанням 3D-об’єктів" обумовлено актуальністю використання інформаційних технологій, зокрема мобільних додатків, в медичній освіті. 3D-моделювання стає все більш поширеним у навчальному процесі, особливо в медицині, де воно дозволяє набагато краще зрозуміти структуру та функції органів людини. Зростаюча потреба в цифрових навчальних ресурсах акцентує важливість розробки високоякісних, інтуїтивно зрозумілих додатків, які можуть бути використані як професійними медиками, так і студентами медичних спеціальностей, а також школярами.

Метою дослідження є розробка iOS-додатка, який використовує 3D-моделі органів людини, для підвищення рівня медичної освіти та розширення кругозору медичних працівників. Для досягнення цієї мети було поставлено наступні завдання: проектування інтерфейсу користувача, вибір інструментів для розробки, розробка функціональності додатка, інтеграція 3D-моделей.

Об’єкт дослідження - процес розробки iOS-додатка з використанням 3D-об’єктів. Предмет дослідження - створення iOS-додатку, що використовує 3D-моделі органів людини для навчальних цілей.

Методологічна основа дослідження базується на застосуванні таких методів, як аналіз та синтез, а також системний підхід. Аналіз та синтез були використані для вивчення теоретичних матеріалів та вибору оптимального набору технологій та інструментів для розробки додатка. Системний підхід було використано для розробки архітектури додатка, проектування його інтерфейсу та реалізації функціональності.

Робота складається з трьох розділів.

У першому розділі описується специфіка роботи в середовищі iOS, перелічені інструменти, що можуть використовуватись для розробки додатка та принципи роботи з 3D-об’єктами.

Другий розділ присвячений визначенню вимог до додатку, проектуванню користувальницького інтерфейсу та вибору інструментів, що будуть використовуватись в розробці.

У третьому розділі наводиться реалізація додатку з прикладами вихідного коду та результатом у вигляді скріншотів додатку, описано способи, в які класи взаємодіють одне з одним.

Таким чином, вибір цієї теми є актуальним та має велике практичне значення. Результати цього дослідження можуть бути використані для подальшої розробки та вдосконалення додатків для медичної освіти, а також для розширення теоретичних знань з області розробки мобільних додатків та 3D-моделювання.

1. Теоретичні аспекти розробки iOS-додатків та використання 3D-об'єктів

1.1. Огляд платформи iOS та її особливостей

iOS — це операційна система, розроблена Apple для її мобільних пристроїв, таких як iPhone та iPad. Вона відрізняється стабільністю, безпекою, високою продуктивністю та зручністю використання.

Особливості платформи iOS включають:

- безпека: iOS відома своїми суворими механізмами безпеки, які включають шифрування даних, ізоляцію додатків та регулярні оновлення безпеки;
- екосистема: iOS тісно інтегрована з іншими продуктами Apple, що дозволяє легко синхронізувати дані та застосунки між пристроями;
- App Store: це централізований магазин додатків, що забезпечує високу якість та безпеку додатків для користувачів.

1.2. Інструменти та мови програмування для розробки iOS-додатків

Swift, мова програмування від Apple, використовується для розробки додатків на iOS. Swift було створено з метою зробити код більш безпечним, водночас забезпечуючи високу продуктивність. Вона включає в себе ряд сучасних особливостей, таких як опціональні типи, безпечне управління пам'яттю, структури даних та багато іншого.

Xcode, інтегроване середовище розробки (IDE) від Apple, є основним інструментом для створення iOS-додатків. Xcode включає в себе редактор коду, інструменти для дизайну інтерфейсу, компілятор, інструменти для тестування та налагодження, а також засоби для управління ресурсами проекту. Окрім того, присутній такий засіб як Canvas, за допомогою якого

можна спостерігати зміни вигляду інтерфейсу динамічно залежно від коду, що значно полегшує та пришвидшує розробку додатків.

SwiftUI — це революційний фреймворк для створення інтерфейсів користувача на Swift, який дозволяє розробникам створювати динамічні та високоякісні інтерфейси користувача з консистентним кодом.

1.3. Принципи роботи з 3D-об'єктами в iOS

Робота з 3D-об'єктами в iOS базується на використанні фреймворків, які надають інструменти для створення, рендерингу та анімації 3D-моделей.

- створення 3D-моделей: iOS надає інструменти для імпорту 3D-моделей з різних форматів, а також для створення нових моделей безпосередньо в коді;
- рендеринг 3D-моделей: фреймворки, як-от SceneKit, надають різноманітні інструменти для рендерингу 3D-моделей, включаючи різні види освітлення, тіні, текстури та ефекти;
- анімація 3D-моделей: інструменти iOS дозволяють анімувати 3D-моделі, створюючи рухи, обертання, масштабування та інші ефекти;
- віртуальна та доповнена реальність: iOS надає фреймворки, такі як ARKit, для створення додатків з віртуальною та доповненою реальністю. Це включає відображення 3D-моделей в реальному світі через камеру пристрою;
- інтеграція 3D-моделей у інтерфейс користувача: За допомогою SwiftUI та бібліотеки Model3DView можна легко вбудовувати 3D-моделі безпосередньо в інтерфейс користувача.

1.4. Бібліотеки та інструменти для роботи з 3D-об'єктами в iOS

Для роботи з 3D-об'єктами в iOS, розробники можуть використовувати ряд бібліотек та інструментів:

- SceneKit — це високорівневий фреймворк для роботи з 3D-графікою, що дозволяє створювати складні 3D-сцени та анімації без необхідності роботи з низькорівневим кодом;
- Model I/O — це фреймворк, що дозволяє імпортувати та експортувати 3D-моделі в різних форматах, а також створювати, обробляти та оптимізувати 3D-моделі безпосередньо в iOS;
- Swift Package Manager — це інструмент від Apple для автоматизації процесу завантаження, компіляції та поєднання залежностей вашого проекту. Це включає бібліотеки для роботи з 3D-об'єктами, як-от Model3DView;
- Model3DView — це бібліотека, розроблена для використання з SwiftUI, яка дозволяє вам інтегрувати 3D-моделі безпосередньо в інтерфейс користувача вашого додатка. Це значно спрощує процес відображення 3D-моделей в SwiftUI;
- ARKit — це фреймворк від Apple, що надає інструменти для створення додатків з доповненою реальністю. Хоча він не є необхідним для всіх додатків, що використовують 3D-моделі, ARKit може використовуватися для створення більш іммерсивних та вражаючих досвідів, що включають 3D-моделі в реальному світі.

Використання цих інструментів та бібліотек дозволяє розробникам створювати потужні та вражаючі додатки, які використовують 3D-моделі для створення різноманітних досвідів для користувачів.

2. Проектування та розробка iOS-додатка з використанням 3D-об'єктів

2.1 Визначення вимог до додатка

Перед початком проектування та розробки додатка необхідно визначити його вимоги. Вони включають функціональні та нефункціональні вимоги, а також вимоги до користувацького інтерфейсу.

Функціональні вимоги визначають, що додаток має робити. У нашому випадку, функціональні вимоги включають:

- перегляд 3D-моделей органів та систем людини;
- можливість зберігати вибрані моделі в закладках для швидкого доступу;
- пошук по системам органів;
- відображення інформації про кожен орган: назва, назва на латині, опис, діаграма та 3D-модель;
- підтримка орієнтації та зуму для 3D-моделей;
- можливість автоматичної зміни фону додатку в залежності від обраної системної теми;
- адаптація під різні розміри екрану пристрою.

Нефункціональні вимоги визначають як додаток має працювати. Вони включають:

1. Швидкість відкликання: додаток має відповідати на вхідні дії користувача в максимально короткі терміни.
2. Надійність: додаток має зберігати коректні дані та відновлювати роботу після помилок.
3. Сумісність: додаток має працювати на всіх підтримуваних версіях iOS.

Вимоги до користувацького інтерфейсу включають:

1. Навігація: додаток має мати чітку та зрозумілу структуру навігації.
2. Дизайн: додаток має мати привабливий та зручний для користувача інтерфейс, втілювати принципи Human Interface Guidelines: бути естетично цілісним, інтуїтивним та легким для сприйняття.
3. Доступність: додаток має бути доступний для людей з різними можливостями.

2.2 Проектування інтерфейсу користувача

Проектування інтерфейсу користувача є критичним етапом в розробці будь-якого додатка. Воно диктує, як користувачі будуть взаємодіяти з додатком, а також впливає на їх загальне враження від використання додатка.

Основний екран додатка представляє собою список доступних систем (наприклад, нервова система, кровообіг, травлення тощо). Користувач може вибрати будь-яку систему, щоб побачити пов'язані з нею 3D-моделі органів. На цьому екрані також присутній пошуковий рядок для швидкого пошуку потрібної системи.

При виборі конкретної системи відкривається новий екран, на якому відображаються всі органи, що належать до обраної системи. Кожен елемент списку містить невелику 3D-модель органу та його назву. При натисканні на орган користувач переходить на екран детальної інформації про модель.

На екрані деталей має відображатись повноекранна 3D-модель вибраного органу, яку можна обертати та масштабувати. Також на цьому екрані відображається детальна інформація про орган, включаючи його назву, латинську назву та повний опис. Також надається діаграма органу

для того, щоб користувач міг не тільки бачити модель, а й розуміти, що на ній зображено. Користувач може додати орган до списку обраних, натиснувши на відповідну іконку.

Додатковий екран "Збережені" містить список органів, які користувач додав до вибраних. Це дає змогу швидко отримати доступ до найбільш важливих для користувача моделей.

Весь дизайн інтерфейсу користувача має бути виконаний в нейтральній темі, що змінюватиметься залежно від системного кольорового режиму. 3D-моделі завжди матимуть темний фон, щоб користувачам було легше їх сприймати та бачити деталі.

2.3 Вибір технологій для розробки

Для розробки додатка було вибрано мову програмування Swift, для створення інтерфейсу користувача було вибрано фреймворк SwiftUI.

Щодо роботи з 3D-моделями, то в роботі застосовуватиметься бібліотека Model3DView. Завдяки їй, можна легко вставити 3D-модель в SwiftUI вид і надати користувачу можливість маніпулювати моделлю (обертати, збільшувати, зменшувати), використовуючи відповідні функції-модифікатори.

Поряд з Model3DView також є SceneView, що є частиною ARKit, бібліотеки від Apple для розширеної реальності. Обидві бібліотеки дозволяють інтегрувати 3D-моделі в iOS додатки, але вони мають ряд важливих відмінностей, які можуть вплинути на вибір між ними.

1. Сумісність з SwiftUI: Model3DView розроблена спеціально для роботи з SwiftUI, тому її легко інтегрувати в SwiftUI додатки. З іншого боку, SceneView є частиною UIKit.

2. Простота використання: Model3DView надає простий API для роботи з 3D-моделями. Вона дозволяє легко завантажувати моделі, маніпулювати ними та реагувати на дії користувача. З іншого боку, SceneView є більш потужним і гнучким, але також більш складним у використанні.
3. Фокус на 3D-моделях: Model3DView фокусується на відображенні 3D-моделей, в той час як SceneView є частиною більшої системи, що підтримує розширену реальність. Якщо вам не потрібні можливості AR, Model3DView може бути більш підходящим вибором.
4. Ресурси: Model3DView вимагає менше ресурсів, що робить її більш ефективною для простого відображення 3D-моделей. SceneView, з іншого боку, є більш потужним інструментом і вимагає більше ресурсів.

Отже, вибір між Model3DView і SceneView залежить від конкретних потреб вашого проекту. Для простих додатків, що використовують SwiftUI і потребують лише відображення на кшталт нашого, Model3DView чудово підходить.

2.4 Обґрунтування вибору формату .gltf

Окремої уваги заслуговує вибір формату для зберігання та роботи з 3D-об’єктами. В цій роботі використовується такий формат як .gltf (GL Transmission Format) для інтеграції 3D-моделей в додаток, що має кілька обґрунтувань:

- Відкритий стандарт: Формат .gltf є відкритим стандартом, розробленим спеціально для передачі та відображення 3D-моделей у веб-та мобільних додатках. Це означає, що він має широку

підтримку та інтеграцію з різними платформами і інструментами розробки.

- Легкість передачі та завантаження: Формат .gltf використовує компактний бінарний формат, що зменшує розмір файлу моделі та полегшує її передачу та завантаження через мережу. Це особливо важливо для мобільних додатків, де обмежені ресурси та швидкість з'єднання.
- Підтримка матеріалів та анімації: Формат .gltf підтримує широкий спектр можливостей, включаючи матеріали, текстури та анімацію. Це дозволяє використовувати різні властивості та ефекти для створення реалістичного візуального відображення органів людини. Завдяки підтримці анімації, можна створювати рухи та інтерактивні ефекти, що поліпшують дослідження та навчання.

Тому було вирішено використовувати даний формат для збереження та роботи з 3D-об’єктами.

2.5 Шаблон проектування

Для розробки додатку було вирішено використовувати шаблон проектування MVVM (Model-View-ViewModel). Цей шаблон складається з трьох основних компонентів - моделей (англ. Model), моделей вигляду (англ. View Model) та виглядів (англ. View).

Шаблон MVVM дозволяє вам розділити логіку представлення від бізнес-логіки, що забезпечує чистоту та підтримуваність коду.

3. Реалізація додатка

3.1 Основні компоненти

Проєкт складається з таких директорій як Views, Data, Models та Objects та папки з медіа-ресурсами додатку Assets. Додаток складається з 9 класів та структур, що розміщені в 4-х директоріях. Далі детально розглядається кожен з класів та їхнє призначення.

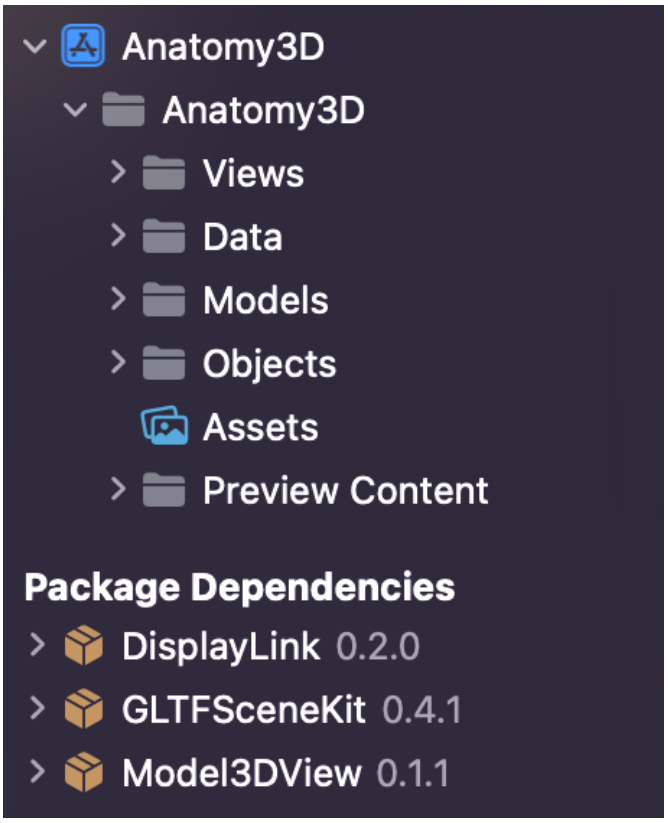

Рисунок 3.1.1 Структура проєкту з директоріями

Директорія Data з файлами “categories.json”, “models.json” та класом ModelsStorage відповідає за роботу з інформацією. Клас керує даними для представлень та включає бізнес-логіку для завантаження та збереження даних моделі. В файлах з розширенням .json зберігаються дані про моделі та категорії.

```
{
    "id": 1,
    "categoryId": 1,
    "title": "Lungs",
    "latin": "pulmo",
    "description": "The lungs are the primary organs of the respiratory system in
        humans and most other animals, including some snails and a small number of
        fish. In mammals and most other vertebrates, two lungs are located near the
        backbone on either side of the heart. Their function in the respiratory
        system is to extract oxygen from the air and transfer it into the
        bloodstream, and to release carbon dioxide from the bloodstream into the
        atmosphere, in a process of gas exchange.",
    "filename": "lungs/lungs.gltf",
    "imageName": "lungs",
    "isSaved": false
},
```

```
{
    "id": 0,
    "title": "Circulatory"
},
```

Рисунок 3.1.2 Екземпляри об’єктів Модель та Категорія в форматі JSON з файлів

Директорія Models зі структурами ModelOrgan та Category зберігає моделі даних - відповідно моделі органу та категорії з їхніми полями.

Директорія Views зберігає наші вигляди - рівень презентації, до якого входять такі класи, як Anatomy3DApp, ListView, CategoryDetail, FavoritesList та ModelDetails. Вони є основними елементами інтерфейсу користувача, що представляють інформацію в вигляді списків у перших чотирьох виглядах та вікна з інформацією про модель в останньому.

Списки для відображення даних реалізовані за допомогою таких засобів бібліотеки SwiftUI як NavigationView та NavigationLink.

```
struct ListView: View {

    @State var searchText: String = ""
    @State var categories: [Category]
    @Environment(\.scenePhase) private var scenePhase
    var sampleCategories: [Category]

    var body: some View {
        NavigationView {
            ZStack {
                List {
                    ForEach(self.categories, id: \.self) { category in
                        NavigationLink {
                            CategoryDetail(category: category.id)
                        } label: {
                            Text(category.title)
                        }
                    } .foregroundColor(.white)
                        .listRowBackground(Color.darkBlueBackground)
                }
                .navigationTitle("Systems")
            }
        } .searchable(text: $searchText, prompt: "Search systems...")
            .onChange(of: searchText) { searchText in
                if !searchText.isEmpty {
                    categories = sampleCategories.filter { $0.title.lowercased().contains(searchText.lowercased()) }
                } else {
                    categories = sampleCategories
                }
            }
```

Рисунок 3.1.2 Код, що використовувався для побудови списків навігації

В директорії Objects зберігаються 3D-моделі, що потім використовуються в інтерфейсі. На даний момент в ній розміщено ще 6 директорій, назви яких відповідають органам людини - heart, lungs, liver і т. д., що містять допоміжні файли для рендерингу об’єктів - з розширенням .gltf, .bin та папку з текстурами.

3.2 Графічний інтерфейс користувача

Додаток має власний логотип, розроблений в процесі написання курсової роботи для того, щоб вирізнятися серед інших.

Рисунок 3.2.1 Іконка додатку

Дизайн інтерфейсу додатка було створено з метою простоти та легкості навігації, забезпечуючи при цьому потужні функції для відображення та інтерактивного дослідження 3D-моделей органів.

Головний екран додатка містить TabView, яка в свою чергу містить два основні вікна: список категорій анатомічних систем (ListView) та список збережених моделей (FavoritesList). Користувач може переключатися між цими двома вкладками, використовуючи вкладки унижній частині екрана.

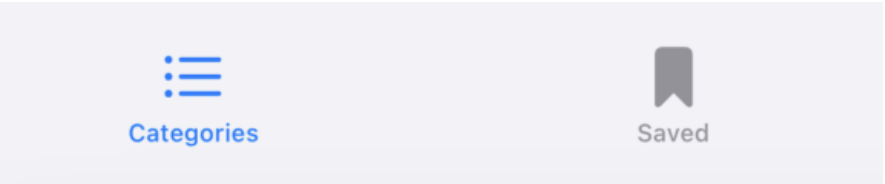

Рисунок 3.2.2 Елемент навігації TabView

ListView відображає список доступних категорій анатомічних систем. Користувач може вибрати одну з них, щоб переглянути 3D-моделі, що входять до цієї категорії. Крім того, ListView містить поле пошуку, яке дозволяє користувачам шукати серед доступних категорій.

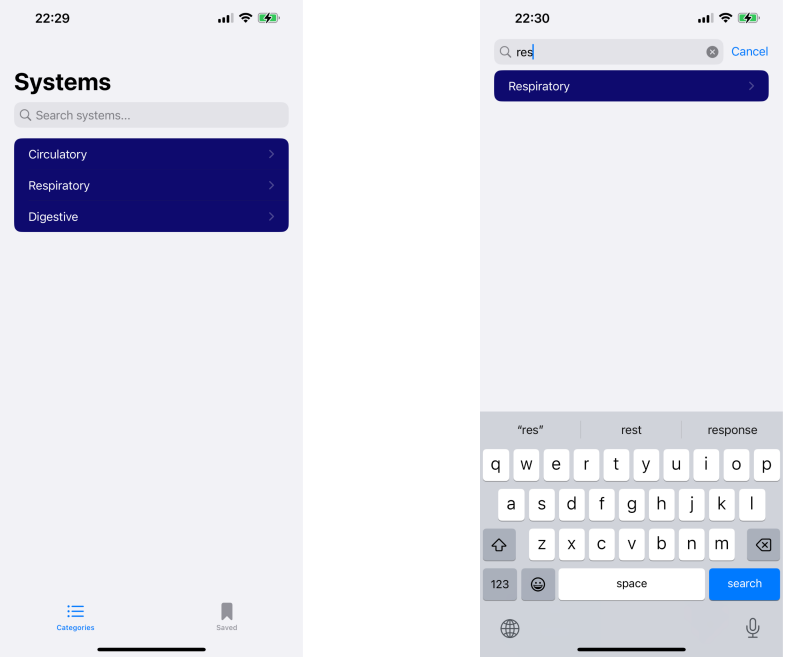

Рисунок 3.2.3 Вкладка Categories та пошук за системами органів

При виборі системи органів можна перейти на наступний рівень навігації - список усіх органів, що входять до обраної системи.

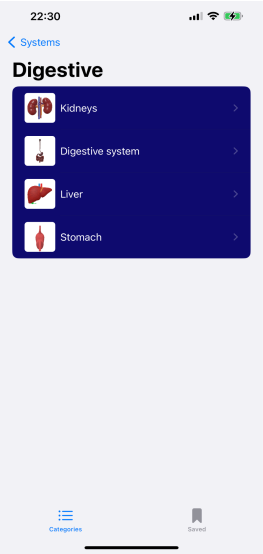

Рисунок 3.2.4 Список органів, що входять до Digestive system

При виборі моделі користувач переходить на екран ModelDetails, де можна бачити деталізовану 3D-модель органа, її опис, зображення, а також має можливість зберегти модель для швидкого доступу.

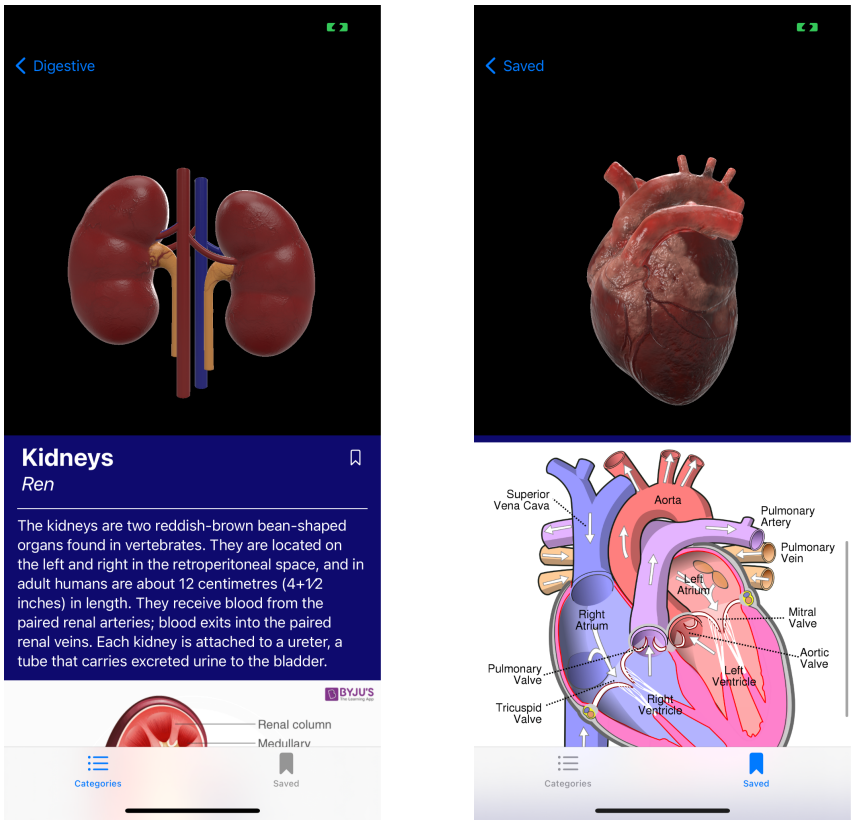

Рисунок 3.2.5 Вигляд ModelDetails - з 3D-об’єктом, описом, кнопкою для збереження та діаграмою

FavoritesList відображає список моделей, які користувач зберіг для швидкого доступу. Користувач може вибрати одну з моделей, щоб переглянути її в деталях.

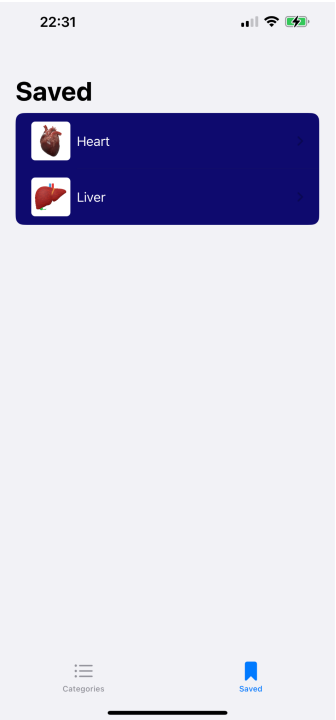

Рисунок 3.2.6 Список збережених моделей

3.3. Інтеграція 3D-моделей органів людини

Для інтеграції 3D-моделей органів людини в додаток було використано формат файлу .gltf. Ось приклад розгортання 3D-моделі у форматі .gltf за допомогою бібліотеки Model3DView:

```
Model3DView(named: model.filename)
    .cameraControls(OrbitControls(
        camera: $camera,
        sensitivity: 0.5,
        friction: 0.1
    ))
    .frame(maxHeight: geometry.size.height / 2)
   .background(.black)
```

Рисунок 3.3.1 Приклад розгортання 3D-об’єкту

У цьому прикладі Model3DView приймає ім'я файлу моделі filename. Це дозволяє Model3DView завантажити та відобразити 3D-модель у форматі .gltf.

Висновки

У даній курсовій роботі було розроблено iOS-додаток для медиків з використанням 3D-об'єктів. Додаток надає можливість досліджувати та вивчати 3D-моделі органів людини з детальними описами. Основна мета роботи полягала в створенні навчального інструменту для медичних фахівців, який допомагає їм краще розуміти анатомію та функції органів людини.

У процесі розробки були використані Swift та Xcode, SwiftUI та Swift Package Manager. Була використана бібліотека Model3DView для відображення 3D-моделей органів. Додаток був спроектований з урахуванням потреб медиків та забезпечує зручний інтерфейс користувача для навігації по категоріях органів, перегляду детальної інформації про моделі та збереження обраних моделей для подальшого доступу.

Результатом роботи є функціональний додаток, який дозволяє медикам вивчати та досліджувати органи людини з використанням 3D-моделей. Він може бути використаний як навчальний інструмент для медичних університетів та навчальних закладів, а також як довідковий посібник для медичних фахівців.

Додаток також має потенціал для розширення функціональності, наприклад, додавання більш докладних описів, розширення набору 3D-моделей та вдосконалення інтерактивності. Також можна розглянути можливість інтеграції додаткових функцій, таких як пошук за латинськими назвами, фільтрація або додаткові медичні ресурси.

У підсумку, розроблений iOS-додаток з використанням 3D-об'єктів є корисним та зручним інструментом для медиків та студентів медичних спеціальностей. Він допомагає покращити розуміння анатомії та функцій органів людини, надає зручність у вивченні та використанні 3D-моделей.

Додаток може бути використаний як самостійний навчальний ресурс або як доповнення до традиційних методів навчання.

Список літератури

1. SwiftUI - Apple Developer Documentation [Електронний ресурс]. - Режим доступу до ресурсу: https://developer.apple.com/documentation/swiftui/.

2. Understanding the Codable Protocol in Swift - Swift by Sundell [Електронний ресурс]. - Режим доступу до ресурсу: https://www.swiftbysundell.com/basics/codable/.

3. SceneKit - Apple Developer Documentation [Електронний ресурс]. - Режим доступу до ресурсу: https://developer.apple.com/documentation/scenekit/.

4. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design / Robert C. Martin. – Upper Saddle River: Prentice Hall, 2017. – 1432 с.

5. SwiftUI by Tutorials: Learning to Create Stunning Apps with SwiftUI / by Antonio Bello, Phil Laszkowicz, Bill Morefield, Audrey Tam, Jayant Varma and Michael Katz. — First edition. — Razeware LLC, 2020. — 492 p.

6. SwiftUI Essentials - iOS Edition: Learn to Develop iOS Apps using SwiftUI, Swift 5 and Xcode 11 / Neil Smyth. - Payload Media, 2020. - 770 p.

7. Model-View-ViewModel (MVVM) in Swift - Swift by Sundell [Електронний ресурс]. - Режим доступу до ресурсу: https://www.swiftbysundell.com/articles/mvvm-in-swift/.

8. Model3DView [Електронний ресурс]. - Режим доступу до ресурсу: https://github.com/frzi/Model3DView.