

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Херсонський державний університет

Факультет комп'ютерних наук, фізики та математики

Кафедра інформатики, програмної інженерії та економічної

кібернетики

**РОЗРОБКА ІНТЕРАКТИВНИХ 3D ОБ'ЄКТІВ ДОПОВНЕНОЇ
РЕАЛЬНОСТІ В КОМП'ЮТЕРНИХ ПРОГРАМАХ
НАВЧАЛЬНОГО ПРИЗНАЧЕННЯ**

Кваліфікаційна робота (проект)

на здобуття ступеня вищої освіти “магістр”

Виконав: здобувач 2 курсу 241м групи
Спеціальності 121 Інженерія програмного
забезпечення

Освітньо-професійної програми
«Інженерія програмного забезпечення»
другого (магістерського) рівня освіти

Соболь Микола Вікторович

Керівники: кандидат фізико-
математичних наук, доцент

Кравцов Геннадій Михайлович

кандидат фізико-математичних наук,
доцент Єрмолаєв В.А.

Рецензент: Лякутін В.В, ФОП «Лякутін»

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. Визначення проблематики та напрямку дослідження.	
Огляд наукової літератури з напрямку дослідження. Огляд наявних програмних рішень, що торкаються проблематики дослідження.	
	8
1.1. Визначення проблематики та напрямку дослідження.....	8
1.2. Огляд наукової літератури з напрямку дослідження.....	18
1.3. Огляд наявних програмних рішень, що торкаються проблематики дослідження	21
РОЗДІЛ 2. Проектування архітектури AR застосунку. Проектування архітектури інтерактивного об’єкта доповненої реальності. Розробка інтерактивного об’єкта доповненої реальності.	
	24
2.1. Проектування архітектури AR застосунку.	24
2.2. Проектування архітектури інтерактивного об’єкта доповненої реальності.....	31
2.3. Розробка інтерактивного об’єкта доповненої реальності.	34
ВИСНОВКИ	43
ДЖЕРЕЛА	Ошибка! Закладка не определена.
ДОДАТКИ	48
Додаток А.....	48
Додаток Б	56

Умовні позначення

AR – аугментована реальність.

VR – віртуальна реальність.

Префаб – шаблон об'єкту в ігровому рушії Unity 3D.

ВСТУП

Тенденції на впровадження дистанційного навчання в освітній процес середньої школи призвів до збільшення запиту на засоби інформаційних технологій. В першу чергу це стосується природничих дисциплін, адже саме вони потребують найбільшої наочності при викладанні матеріалу. Технологія AR цілком дозволяє досягнути значних результатів на цьому полі, адже головною перевагою цієї технології є потенціал до візуалізації. Однак, для використання AR-застосунку в освітньому процесі, останній потребує розробки спеціально змодельованих інтерактивних об'єктів, що відповідають поставленими перед ними вимогам. Саме питання моделювання архітектури та розробки таких інтерактивних об'єктів, з подальшим використанням їх в навчальних програмах середньої освіти, є завданням цього дослідження.

На комерційному полі вже існують приклади AR-застосунків, що намагаються розв'язувати цю проблему та результати яких були розглянуті в цьому дослідженні, таких як Merge EDU та Google Expeditions. Детальніше ці приклади, їх переваги та недоліки, будуть розглянуті далі в роботі.

Дослідження автора використовує результати таких робіт:

«Методи та технології віртуальної та доповненої реальності в комп'ютерних програмах навчального призначення», за авторством Пулінець А.Ю.

Мета дослідження: довести доцільність застосування інтерактивних об'єктів доповненої реальності при візуалізації фізичних процесів та явищ в освітньому процесі середньої школи.

Завдання дослідження:

- Розглянути та проаналізувати наукову літературу з проблеми використання AR-технології в освітньому процесі середньої школи.

- Розглянути та проаналізувати наукову літературу з проблеми використання AR-технології при візуалізації фізичних процесів та явищ, що вивчаються при проходженні курсу фізики в закладах середньої освіти.

- Описати модель та архітектуру інтерактивного об'єкта доповненої реальності.

- Розробити інтерактивний об'єкт доповненої реальності засобами ігрового рушія Unity 3D для використання у візуалізації фізичного закону чи явища.

Об'єкт дослідження: концепція і будова поняття інтерактивних об'єктів доповненої реальності, робота з ними та способи їх застосування в навчальних засобах.

Предмет дослідження: програмне забезпечення інтерактивних об'єктів доповненої реальності та їх опрацювання засобами рушія Unity 3D, враховуючи їх подальше використання в навчальних засобах.

Методи дослідження:

- Теоретичні: аналіз, синтез,

- Емпіричні: спостереження

Наукова новизна одержаних результатів, полягає в науково-методичних рекомендаціях, спрямованих на вдосконалення моделі та архітектури інтерактивних об'єктів доповненої реальності, зменшення їх собівартості при розробці. Основні результати, які були отримані в процесі вирішення поставлених завдань та становлять наукову новизну дослідження, полягають у наступному:

вдосконалено:

- Методичні положення з використання інтерактивних об'єктів доповненої реальності при візуалізації фізичних та явищ та процесів, що дозволило обґрунтувати їх використання як інструмент збільшення наочності викладання освітнього матеріалу з фізики.

- Архітектуру та модель інтерактивних об'єктів доповненої реальності, що дозволило збільшити їх гнучкість та універсальність для використання з різними за типами візуалізаціями.

набули подальшого розвитку:

- Архітектура AR-застосунку, а саме: розроблений уніфікований підхід до взаємодії AR-застосунку та інтерактивного об'єкта симуляції – відділення функціоналу об'єкта від основної архітектури застосунку.

Практичне значення одержаних результатів полягає у науковій обґрунтованості та прикладній спрямованості теоретичних положень, підходів і рекомендацій, викладених у роботі, використання яких зробить процес викладання теоретичного матеріалу з природничих наук, таких як фізика, у закладах середньої освіти, більш наочним для учнів, підвищить рівень зацікавленості учнів під час вивчення фізичних законів та явищ, зменшить собівартість демонстрації фізичних законів та явищ, відносно класичних способів, таких як лабораторний експеримент, усуне можливий ризик здоров'ю та життю учнів, що існує при використанні лабораторного експерименту, як засобу демонстрації фізичних явищ.

Апробація результатів дослідження

Результати розробки методології створення архітектури інтерактивних об'єктів ввійшли в такі роботи автора, як: стаття «Використання AR-технологій у візуалізації моделювання фізичних законів в навчальному процесі середньої школи», що була подана до наукового видання «Магістерські студії» (Херсонський державний університет) та «Проектування та розробка інтерактивних компонентів AR-застосунків для візуалізації моделювання фізичних законів в навчальному процесі середньої школи», що була подана до участі в науковій інтернет-конференції «Вітчизняна наука на зламі епох: проблеми та перспективи розвитку» (вип. 73) (Університет Григорія Сковороди в Переяславі).

РОЗДІЛ 1

ВИЗНАЧЕННЯ ПРОБЛЕМАТИКИ ТА НАПРЯМКУ ДОСЛІДЖЕННЯ. ОГЛЯД НАУКОВОЇ ЛІТЕРАТУРИ З НАПРЯМКУ ДОСЛІДЖЕННЯ. ОГЛЯД НАЯВНИХ ПРОГРАМНИХ РІШЕНЬ, ЩО ТОРКАЮТЬСЯ ПРОБЛЕМАТИКИ ДОСЛІДЖЕННЯ.

1.1. Визначення проблематики та напрямку дослідження

Розповсюдження напівпровідникових матеріалів та подальше винайдення транзистора [9], призвели до початку нової так званої «інформаційної» технічної революції [15]. З ходом розвитку напівпровідникового транзистора змінювався й рівень розвитку інформаційних технологій та його вплив на суспільство. Розпочавши з простої інтегральної схеми, що виконує лише декілька простих обчислень і використовується як науковий інструмент, мікропроцесорна технологія еволюціонувала до свого сучасного стану – мікроскопічного пристрою, що міститься майже в кожному сучасному пристрої: від кишенькового телефону і до космічних апаратів [13]. Цей маленький крок для кремнієвої пластинки став гігантським стрибком для нового витку розвитку технологій сучасної цивілізації.

Важко назвати іншу сучасну технологію, як інформаційна, яка б мала такий вплив на сучасне суспільство та життя кожного з нас. Плоди її застосування не просто нас оточують, а вже стали невіддільною частиною повсякденності. Розповсюдження інформаційних технологій проникло у всі сфери сучасної цивілізації: починаючи від повсякденних гаджетів (смартфонів, розумних годинників-асистентів, побутової техніки тощо), якими користується кожен з нас щодення, спрощуючи наш побут та збільшуючи ефективність нашої праці, закінчуючи

глобальними системами, що стежать та контролюють цілі підприємства чи навіть галузі або напрочуд високотехнологічними науковими інструментами, такими як суперкомп'ютери [10], наприклад, що залучені у вирішенні складних наукових завдань.

Починаючи від сфери розваг (комп'ютерні ігри, тривимірна графіка) та закінчуючи науковими дослідженнями, інформаційні технології справедливо зайняли свою нішу, як напрочуд ефективні інструменти. Тому не дивно, що рано чи пізно постало питання їх використання у такій важливій сфері людської діяльності, як освіта. Такі технології, як інтернет та засоби тривимірної графіки, дозволили розвинути та вивести на новий якісний рівень такі розділи освіти як дистанційне та індивідуальне навчання. Нині, майже будь-який навчальний матеріал можна отримати з будь-якої точки світу, маючи лише підключення до мережі. З розповсюдженням всесвітньої мережі та збільшення доступності таких технічних засобів, як персональні комп'ютери та смартфони, збільшився і попит на різноманітні матеріали дистанційного навчання, таких як відео- та аудіокурси тощо. Це своєю чергою зменшило роль фактору очної присутності при отриманні освіти та збільшило гнучкість та мобільність засобів підвищення кваліфікації для уже дипломованих спеціалістів.

Однак, швидке розповсюдження інформації вплинуло не лише на наукову та освітні сфери, а й на сфери розваг та публіцистики. За короткий період у декілька десятиліть, людство справило інформації більше, чим за всю попередню свою історію і швидкість її виникнення лише збільшується [5]. Цей ефект отримав назву «інформаційний вибух» [8]. В результаті, інформації стало настільки багато, що пересічному споживачеві, більша її частина сприймається як «інформаційний шум», тобто як, та що не має ніякої користі для нього. На жаль, це не завжди

відповідає дійсності, оскільки на первинне сприйняття інформації впливає більшою мірою її подача. Тому, інформація, яка на суб'єктивне сприйняття споживача є цікавою, тобто від якої він отримав найбільший емоційний вплив. Це процес є цілком зрозумілим: мозок як біологічна нейромережа буде з більшою ймовірністю утворювати нейронні зв'язки з найбільшою «ціною». З точки зору біохімії мозку, цю «ціну» визначають сила емоційного забарвлення яке асоціюється з отриманою інформацією: та яка супроводжувалась сильним емоційним забарвленням, неважливо позитивним чи негативним, з більшою ймовірністю та тривалістю закріпиться в пам'яті. Хоча цей ефект використовувався в комерційних та розважальних цілях вже давно, однак з «інформаційним вибухом» він отримав масове поширення. Для рекламування свого продукту чи в розважальних цілях, нову інформацію, в першу чергу, почали подавати через призму отримання емоцій від неї, а не користі чи повноти.

Незважаючи, що такий підхід використовується в історичній перспективі не дуже давно, однак вже помітні деякі його наслідки. В першу чергу, це так званий ефект «кліпового мислення».

Кліпове мислення (від англ. «clip» - «вирізка, уривок») [19] – говорячи про такий тип мислення, мається на увазі, коли інформація сприймається короткими яскраво забарвленими (включаючи в емоційному плані) уривками (кліпами), без спроби встановити між ними логічних зв'язків. Кліп визначають, набір коротких за своєю суттю тез, подача яких йде поза контекстом, оскільки передбачається, що вони в силу своєї актуальності мають на увазі об'єктивну реальність. як поточний контекст [4].

Вважається, що такий тип мислення є характерним для так званого Z-покоління [14] – людей, зростання яких припало розповсюдження

світової глобалізації та інформаційних технологій. Швидка зміна та перенасиченість інформації змушує жити та пристосовуватись до світу, що швидко змінюється. Однак, при зустрічі такого типу мислення з сучасною системою освіти, що навпаки, досить повільно пристосовується до новітніх тенденцій, з'являються безліч освітніх колізій на підставі не відповідності поточних методологій викладання навчального матеріалу та потреб й можливостей учнів. В першу чергу, це проявляється в тому, що більшість методологій навчання передбачають значної концентрації уваги учня на матеріалі, продовж достатньо тривалого часу. В силу своїх особливостей сприйняття матеріалу учні, на жаль, не можуть цього досягнути. Особливо це стосується проходження шкільних курсів природничих наук, що потребують підвищеної концентрації уваги та значної розумової діяльності, в ході якої, потрібно розуміння та опрацювання концепцій, причиною якого є їх часта абстрактна природа або недостатня наочність. В класичній методології, останнє намагаються подолати за допомогою візуалізації, зазвичай двомірної (ілюстрації в підручнику), або, що рідше, лабораторних експериментів, що є напрочуд наочними, але мають декілька недоліків:

- Потребують значної матеріально-технічної бази.
- Можуть становити загрозу для здоров'я та життя учнів.
- Виходячи з попереднього, потребують значного рівня компетенції та концентрації уваги не тільки від викладача, а й від учня.
- Виходячи з всіх вище вказаних недоліків, ймовірність повторення лабораторного експерименту для кожного учня індивідуально з метою закріплення чи перевірки пройденого матеріалу, є малоімовірним.

Окрім того, рівень розвитку сучасних технологій не тільки швидко породжує нові, ті ж самі технології, але й не меншою мірою потребує

технічних спеціалістів, що будуть займатись їх супроводженням та розвитком. Однак, опанування подібними професіями, особливо тими, що передбачають роботу з електронікою або в інших наукомістких галузях, передбачає високий рівень володіння прикладними дисциплінами, що ґрунтуються на, якраз таки, знаннях які можна отримати лише з природничих наук. Наприклад, радіотехнік чи інженер мікроконтролерів навряд чи буде компетентним у своєму фасі без елементарних знань про природу та властивості електричного струму. І як вже зазначалось раніше, збільшення учнів з «кліповим мисленням», може призвести до проблем у розвитку сучасних технологій, якщо не будуть розроблені нові методики навчання, що будуть враховувати особливості сприйняття інформації цих дітей.

Тому не дивно, що проблемою пошуку нових засобів та видів візуалізації навчального матеріалу, стурбовані безліч спеціалістів сфери освіти. Саме тут ми можемо сформулювати **одне з питань проблематики** цього дослідження:

- Доцільність використання сучасних інформаційних технологій для збільшення якості наочності візуалізації при вивченні матеріалу з природничих наук.

А враховуючи, що найбільше цього потребує така наука як фізика, то ми можемо уточнити це питання як «збільшення якості наочності візуалізації при демонстрації фізичних законів та явищ».

На щастя, інформаційні технології не тільки породили проблему, а й декілька інструментів її вирішення, таких як AR[2]- та VR[18]-симуляції.

Оскільки в цьому дослідженні розглянута саме AR технологія, то розглянемо детальніше її сутність та засоби якими вона володіє, що можуть допомогти у вирішенні визначеної проблематики:

AR (англ. augmented reality – аугментована (доповнена) реальність) – термін, що позначає всі технологічні засоби та проекти, що спрямовані на доповнення нашої об'єктивної реальності будь-якими елементами віртуальної реальності [16]. AR – є складовою такого поняття, як змішана реальність (*mixed reality*). Протилежною технологією є AV (англ. augmented virtuality – доповнена віртуальність), що спрямована на протилежну ціль – інтегрування реальних об'єктів у віртуальне середовище.

Узагальнене визначення для AR-застосунку [3] – це система, яка:

1. Поєднує реальне та віртуальне середовище.
2. Взаємодія в реальному часі.
3. Працює в тривимірному просторі.

Сама архітектура застосунку зазвичай складається з таких частин:

- Апаратне забезпечення
- Система розпізнавання
- Пристрій зчитування

Зараз головною сферою використання AR-застосунків є сфера розваг. Однак, найбільш ймовірними користувачами можна також вказати:

- Освіта
- Медицина
- Військові технології
- Реклама та маркетинг
- Поліграфія

Однак це не повний список.

Виходячи з опису та можливостей технології можна оцінити її ступінь актуальності в контексті поставленого питання використання AR, як засобу візуалізації фізичних явищ та процесів. Для цього виділимо декілька пунктів доцільності:

- По-перше, AR-технологія є одним провідних трендів сучасних інформаційних технологій. Немає сумнівів, що ця сфера буде й надалі швидко розвиватись.

- По-друге, вона дає можливість використовувати себе, як елемент дистанційного та індивідуального навчання.

- По-третє, такі переваги AR технології, як візуалізація [7] та контрольованість, робить її ефективним засобом донесення інформації.

Отже, з вище наведених пунктів можна зробити висновок, що є цілком доцільним та актуальним засобом розв'язання проблеми збільшення якості наочності візуалізації фізичних законів та явищ.

Тепер ми можемо детальніше розглянути застосування AR-технологій до вибраної проблематики.

По-перше, виділимо аспекти, де засоби AR-технологій можна застосувати:

1. Візуалізація об'єктів та процесів,
2. Моделювання об'єктів та процесів,
3. Моніторинг та контроль знань учнів у процесі навчання,
4. Можливість швидкого налаштування та модифікації програмних засобів під потреби навчального процесу.

Розглянемо кожний з аспектів детальніше:

1. Візуалізація об'єктів та процесів – має надзвичайно сильний фактор впливу на сприйняття та засвоєння матеріалу учнями [23]. Не зважаючи на те, що в класичному підході до викладання курсу фізики теж застосовуються методи візуалізації, такі як ілюстрації в підручниках чи лабораторний експеримент, однак вони мають недоліки, які й були вказані в проблематиці. Засоби AR, в першу чергу, у питанні демонстрації, мають дві незаперечних переваги: показ в тривимірному просторі та можливість налаштування параметрів симуляції під поточні потреби.

2. Моделювання об'єктів та процесів – важливим аспектом вивчення матеріалу природничих наук є не тільки його розуміння, а й набуття вміння прогнозувати перебіг фізичних процесів в різних умовах та початковими параметрами. На основі власних зроблених розрахунків учні повинні вміти відтворити хід процесу, що вони вивчають. В класичному підході для відтворення використовується лабораторний експеримент, однак, як вже було вказано, він, як інструмент навчання, є напрочуд витратним та може містити фізичної загрози для учня. AR симуляція, своєю чергою, повністю безпечна для життя та має властивість запуску симуляції безліч разів, без витрат на кожну нову її ітерацію.

3. Контроль знань – один з ключових етапів навчання, що дає можливість оцінити рівень засвоєння учнем знань та його розуміння фізичних процесів. В класичному навчанні, роль моніторингу успіхів відіграють такі засоби, як контрольні роботи з проведенням розрахунків, візуальне зображення яких учень, зазвичай не має змоги побачити та фізичні експерименти. Однак, AR дозволяє додати до цього списку ще декілька інструментів. Наприклад, можна дати можливість учню запускати симуляції власноруч, з його власними параметрами, які він отримав, наприклад, в результаті тих же самих власних розрахунків, а

програма AR застосунку буде автоматично опрацьовувати отримані результати симуляції. В такому разі, ми отримуємо деякий аналог фізичного експерименту, на відміну від якого, учень діє самостійно та не потребує постійного контролю зі сторони вчителя. До того ж подібну систему можна модифікувати для постійного моніторингу дій учня та яка буде формувати оцінку його успіхів в автоматичному режимі. Окрім того, слід зазначити, що можливості AR та інтернету дають змогу проводити заняття та фізичні експерименти без очної присутності учня.

4. Можливість швидкого налаштування та модифікації програмних засобів під потреби навчального процесу – є однією з головних переваг AR, яку вона отримує, як будь-який програмний продукт. AR технологія має повну інтерактивність не тільки у використанні, а і в створенні нових та редагуванні старих інструментів. Якщо, наприклад, ілюстрацію, у випадку виявлені помилки, можна виправити лише у випадку повного передрукування всього видання підручника, то у випадку AR достатньо лише задати нові параметри візуалізації. Якщо ж, помилка в самому застосунку, то достатньо щоб розробник зробив виправлення власного коду та надіслав оновлення, що є повністю безплатним для кожного з користувачів. Окрім того, такі виправлення можна проводити повністю в автоматичному режимі, без залучення самого користувача.

Отже, підбиваючи підсумки розгляду аспектів застосування, можна виділити такі переваги AR технології в порівнянні з класичними засобами навчання:

- Можливість візуалізувати майже будь-який фізичний процес чи явище в тривимірному просторі, навіть ті, що зазвичай неможливі за умов звичайного лабораторного експерименту.

- Мінімальні матеріально-технічні вимоги до учня. Останньому, для запуску симуляції, необхідний лише пристрій з підтримкою AR-технології, наприклад, звичайний смартфон з камерою.

- Забезпечення повної безпеки для життя та здоров'я учня в ході симуляції фізичних процесів, навіть тих, що є безперечно небезпечними у звичайних лабораторних умовах.

- Можливість автоматизації моніторингу та контролю успіхів учня у ході проходження матеріалу.

- Можливість проводити заняття без очної присутності учня.

- Повна гнучкість в модифікації AR інструментів, з мінімальними вимогами до витрат зі сторони матеріально-технічної бази.

Однак можна зазначити й деякі недоліки:

- AR інструменти потребують пристрої виводу. Не завжди вдається забезпечити всіх учнів необхідним приладдям з фінансових причин.

- Зазвичай будь-яка робота з AR застосунками потребує постійного підключення до мережі Інтернет.

- Повна безпека під час симуляції фізичних експериментів може дати учню хибну впевненість та нерозважливість у випадку роботи з реальними фізичними явищами, реагентами тощо.

- AR-симуляція потребує спеціально розроблених інтерактивних об'єктів, що повинні, відповідно до вимог, змодельовані.

Саме останнє, дає змогу **уточнити питання проблематики до свого кінцевого варіанту**, що розглядається у цьому дослідженні, а саме:

- Розробка інтерактивних 3D об'єктів доповненої реальності в комп'ютерних програмах навчального призначення.

- Описання методології моделювання та архітектури цих інтерактивних об'єктів та їх взаємодії з архітектурою самого AR-застосунку.

- Доцільність розробки таких інтерактивних об'єктів у контексті їх подальшого використання у візуалізаціях фізичних законів та явищ.

Саме з наведених тез та викладеного в цьому розділі матеріалу, були сформовані завдання дослідження.

Отже, визначившись з проблематикою та завданнями дослідження, ми можемо перейти до наступного розділу, а саме огляду наукової літератури з напрямку дослідження.

1.2. Огляд наукової літератури з напрямку дослідження

В дослідженні були розглянуті декілька наукових робіт, що торкаються проблематики напрямку дослідження.

А саме:

1. Кваліфікаційна робота «Методи та технології віртуальної та доповненої реальності в комп'ютерних програмах навчального призначення» за авторством Пулінець А.Ю. [20]. В розглянутому матеріалі були досягнуті такі результати:

- Визначено доцільність використання AR-застосунку, як засобу навчально-виховного процесу.

- Описані основи методології створення інтерактивних об'єктів засобами Unity 3D та Vuforia.

- Описані основи архітектури та моделі AR-застосунку та його роботи.

- Розроблені приклади AR-застосунків для використання у візуалізації симуляції деяких фізичних процесів.

Однак, в ньому залишились деякі нерозглянуті питання:

- Не розроблено точного описання методології створення моделі та архітектури самих інтерактивних 3D об'єктів.

- Не було розроблено точних методологічних порад для розробки універсальної архітектури AR-застосунку, що дозволив би використовувати декілька симуляцій в одному застосунку.

Також хочеться зазначити, що у створеній у роботі використовується технологія Vuforia, як основа архітектури AR-застосунку, що є застарілою на 2021 рік. Тому, власнім дослідженні ми будемо використовувати актуальну технологію ARfoundation.

2. Наукова публікація «Мультимедійна віртуальна лабораторія з фізики в системі дистанційного навчання» за авторством Козловського Є.О. та Кравцова Г.М. [21]. В розглянутому матеріалі були опрацьовані такі питання:

- Опис методології створення віртуальних лабораторій з фізики, її архітектура та модель за допомогою засобів Unity 3D.

- Розроблені та описані компоненти програмного модуля віртуальної лабораторії.

- Розроблені методичні відомості про створення компонентів програмного модуля віртуальної лабораторії, враховуючи тематику дисципліни «фізика».

- Опис методології інтеграції програмного модуля в вже наявну систему дистанційного навчання «Херсонський віртуальний університет».

Однак, результати цього дослідження можна буде доповнити в деяких аспектах:

- Описаний програмний модуль передбачає використання персонального комп'ютера, як пристрої симуляції, що не підходить для розв'язання питання нашого дослідження, тому потребує адаптації результатів до умов AR.

- Описана методологія передбачає використання програмного модуля, як вбудовану частину системи дистанційного навчання. В нашому ж дослідженні програмний продукт описаний, як повністю самостійний засіб навчання.

Оскільки в нашому дослідженні для прикладу демонстрації буде розроблено інтерактивний об'єкт, що візуалізує процес з такого розділу фізики, як «Сила струму. Вимірювання силу струму», тому була розглянута робота з суміжної теми: “About One Approach to Building Systems for Testing Physical Knowledge” за авторством Львова М., Кузьменкова С. та Кравцова Г. [1]. З цього дослідження були почерпнуті такі відомості та результати:

- Методологія побудови архітектури програмного засобу для симуляції фізичних законів.

- Методологія побудови моделі інтерактивного об'єкта симуляції, що візуалізує роботу з законом Ома та електричного кола в цілому.

Отже, проаналізувавши розглянуті наукові матеріали можна вказати такі завдання цього дослідження, робота над якими продовжувала б і розвивала питання розвитку AR-технологій, як навчальних засобів:

- Розробка програмного AR засобу візуалізації фізичних процесів засобами Unity 3D.

- Створення методичних даних з розробки інтерактивних 3D об'єктів доповненої реальності, засобами Unity 3D.

- Створення методичних даних з розробки моделі та архітектури інтерактивних 3D об'єктів доповненої реальності, для використання при візуалізації фізичних законів.

Розглянувши наукову літературу з напрямку дослідження, ми можемо перейти до наступного розділу – огляду програмних продуктів, що намагаються розв'язати питання, які були розглянуті.

1.3. Огляд наявних програмних рішень, що торкаються проблематики дослідження

Merge EDU

Багатофункціональний комплексний застосунок, зосереджений в цілому на полегшенні сприйняття та вивчення матеріалу природничих наук учнями [11].

В основі презентації своїх навчальних матеріалів розробники керуються технологіями STEM.

AR можливості застосунку:

- Показ об'єктів та повноцінних симуляцій.
- Взаємодія користувача з AR елементами симуляції.
- Керування жестами та симуляція тактильних взаємодій з об'єктами.
- Відстежування погляду користувача.
- Інтеграція з засобами VR.

Приклади інтерактивних AR об'єктів, що були створені:

- Інтерактивна модель сонячної системи, з якою можна працювати за допомогою жестів.
- Анатомічний атлас, площиною для якого може виступати тіло користувача.

- Інтерактивні симуляції хімічних та фізичних процесів. Наприклад симуляція камери згорання.



Рисунок 1.1 - MergeEDU: Інтерактивна модель сонячної систем

Однак існують і значні недоліки цієї розробки:

- Ціна: від 995 до 2 495 доларів на рік, за «класну» та «шкільну» версію. Наявна випробувальна версія.

- Доступна лише на території США.

- Розрахована в цілому на групове навчання під керівництвом вчителя, тому мало підходить для індивідуального використання.

- Потребує додаткових інструментів для показу деяких AR об'єктів. Наприклад, спеціального одягу з нанесеним маркуванням.

Google Expeditions

Суміш AR та VR від Google, що дозволяє створювати та переглядати інтерактивні об'єкти 3D та сцени з панорамним оглядом в 360 градусів [6].

Приклади AR об'єктів:

- Сцени записані в 360 градусів, переміщення в яких узгоджується з переміщенням користувача в реальності.
- Перегляд інтерактивних 3D об'єктів симуляції.
- Перегляд інтерактивних екскурсій у вигляді записаних 3D панорам.



Древний Египет



Houghton Mifflin Harcourt, Vida Systems
VR · 5 сцен

Открыть

Смотреть в VR



Начать экскурсию в режиме гида

Египетская цивилизация развилась тысячи лет назад вдоль реки Нил в Северной Африке. Древний Египет был мощным царством в течение более чем 3000 лет. Его продуктивные сельскохозяйственные угодья вдоль реки Нил являлись базой для большого и сложного государственного аппарата и позволяли египтянам развивать богатую и изощренную культуру. В этой экспедиции вы

Рисунок 1.2 - Приклад інтерактивної екскурсії в Google Expedition

Однак існують деякі недоліки цього продукту:

- Хоча застосунок і працює зі смартфонами, в першу чергу його робота передбачає використання VR пристроїв виводу.

- Розрахований в першу чергу на використання в навчальних закладах, а не в індивідуальному навчанні.

- Google припинив підтримку проєкту в червні 2021.

Зробивши огляд вже наявних програмних засобів з напрямку дослідження, можна дійти висновку, що інтерактивні об'єкти доповненої реальності, які розробляються автором, будуть достатньо актуальними у контексті наявних AR-засобів, у випадку якщо вони будуть:

- Підходити як для групового, так і індивідуального використання.

- Їх функціонування не буде залежати від постійної підтримки розробника.

- Будуть повністю безплатними.

Отже, опрацювавши наукову літературу і наявні програмні засоби та визначивши, що з них ми можемо почерпнути для власного дослідження, ми можемо перейти до наступного розділу – практичної частини.

РОЗДІЛ 2

ПРОЄКТУВАННЯ АРХІТЕКТУРИ AR ЗАСТОСУНКУ.

ПРОЄКТУВАННЯ АРХІТЕКТУРИ ІНТЕРАКТИВНОГО ОБ'ЄКТА ДОПОВНЕНОЇ РЕАЛЬНОСТІ. РОЗРОБКА ІНТЕРАКТИВНОГО ОБ'ЄКТА ДОПОВНЕНОЇ РЕАЛЬНОСТІ.

2.1. Проєктування архітектури AR застосунку.

Для початку проєктування архітектури, треба схарактеризувати основні вимоги до роботи, що вона повинна виконувати та визначити основні компоненти.

В першу чергу, керуючись теоретичною інформацією, що ми отримали при опрацюванні літератури, визначимо основні компоненти AR-технології. Це:

- Апаратне забезпечення – це сукупна назва для компонентів пристрою, що використовується для опрацювання AR симуляції. В загальному воно складається з: процесору, датчиків для роботи з навколишнім середовищем, пристроїв вводу та виводу інформації.

- Система розпізнавання – це основа будь-якої технології AR. Вона є складним програмним комплексом, що отримуючи різноманітні дані, наприклад, геолокацію, просторову орієнтацію, час чи розпізнанні форми, від апаратного забезпечення, намагається ідентифікувати в фізичному просторі об'єкти та людей в реальному часі. В залежності від того, як система розпізнавання намагається інтерпретувати отримані дані, вона ділиться на локаційно-залежну та об'єктно залежну. До того ж остання може бути реалізована різними способами: за допомогою штучних маркерів та без.

Для того, щоб звузити вимоги до нашого застосунку, змоделюємо умовну ситуацію, коли необхідне його використання. Опишемо узагальнений вид роботи застосунку:

- Користувач (учень) запускає застосунок на своєму пристрої. Застосунок вмикає пристрій камери.

- Користувач наводить камеру на об'єкт, що нам треба розпізнати.

- Система розпізнавання зчитує всі об'єкти, що потрапляють до камери.

- У випадку, якщо шуканий об'єкт був знайдений, запускається симуляція.

Отже, змодельовавши ситуацію використання AR-застосунку, ми можемо зробити першу ітерацію визначання вимог до архітектури застосунку. А саме:

- Застосунок повинен мати зв'язок з апаратним забезпеченням пристрою користувача.

- Визначити вид системи розпізнавання: об'єктно чи локаційно-залежна.

- Визначити спосіб збереження інформації про шуканий об'єкт.

- Визначити спосіб роботи з симуляцією.

Для наступної ітерації формування вимог, уточнимо змодельовану ситуацію, конкретизувавши її роботу, додавши контекст використання застосунку, як засобу візуалізації фізичного процесу. Тому, нехай змодельована ситуації буде таке:

- Учень запускає застосунок.

- Учень наводить камеру пристрою на ілюстрації фізичного процесу в підручнику.

- Система спостереження розпізнавши ілюстрацію та зіставивши її зі збереженими даними, запускає відповідну симуляцію.

З цієї ітерації ми можемо уточнити деякі з вимог, а саме:

- Тип системи розпізнавання повинен бути оснований на відстежуванні об'єктів, адже ілюстрація яку ми шукаємо є об'єктом, точніше частиною об'єкта підручника, а під час симуляції ми будемо відстежувати зміну положення саме об'єкта відносно пристрою.

- Дані про об'єкт розпізнавання ми будемо зберігати в базі даних з якого зв'язується застосунок.

Отже, спрощену архітектуру застосунку можна змодельувати такою:

- Апаратне забезпечення (АЗ) – комплекс технічних засобів, у нашому випадку смартфон, що отримує з довкілля необхідні для симуляції дані, передає їх до системи розпізнавання, а потім використовує повернені дані для опрацювання поточного стану симуляції.

- Система розпізнавання (СР) – розпізнає та відстежує об'єкти за даними, які надходять з пристроїв візуального введення (камери).

- Об'єкт маркер (ОМ) – безпосередньо шуканий об'єкт, який знаходить і відстежує СР. Може бути, як об'ємним об'єктом, так і плоским зображенням. Дані про ОМ зберігаються та надаються з бази даних (БД).

- База даних (БД) – колекція інформації, що містить в собі дані про ОМ та об'єкт симуляції.

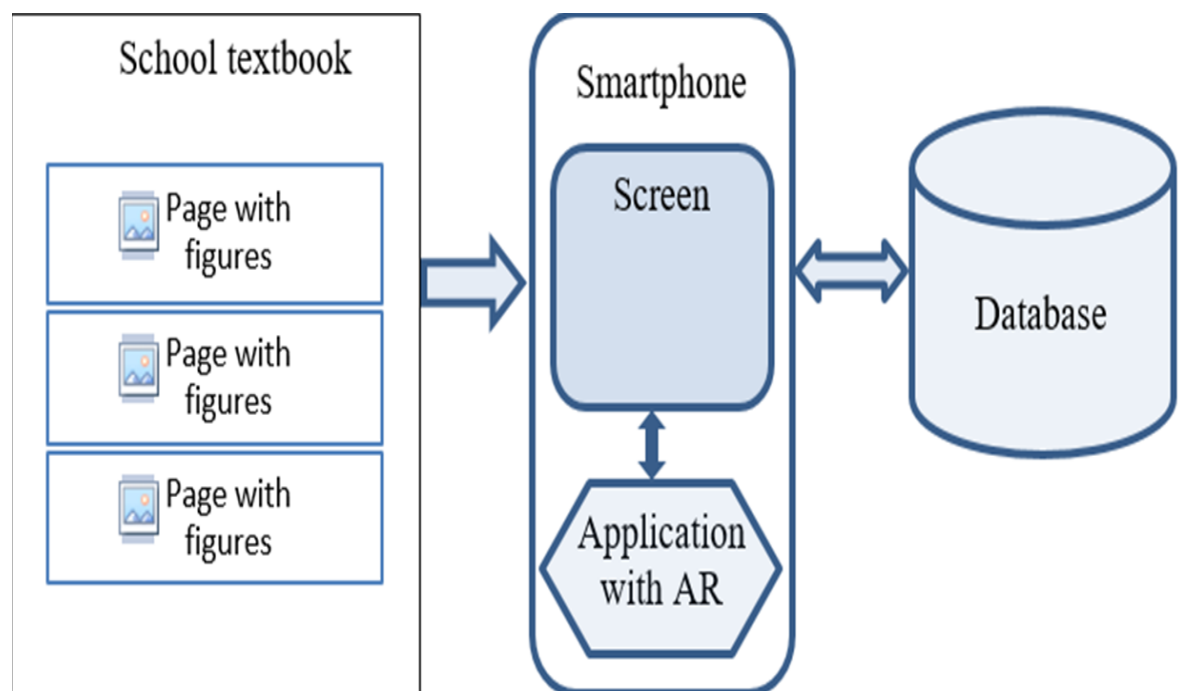


Рисунок 2.1 - Спрощена архітектура застосунку

Визначившись з найпростішим представленням архітектури застосунку, перейдемо до наступної ітерації її створення. Для цього ми визначимось з безпосередньо з технологіями, які ми будемо використовувати в застосунку.

Основу архітектури створюють дві технології: ігровий рушій Unity 3D та його дочірня розробка AR Foundation.

Unity 3D [17] - один з найпопулярніших сучасних ігрових рушіїв. Має широке використання не тільки у сфері розваг, а і як інструмент створення наукових та інших професійних застосунків, як то віртуальні лабораторні роботи, AR застосунки тощо. Причиною такої популярності рушію є його основні переваги: безплатність, зручність у використанні, підтримує розробку застосунків майже під будь-яку сучасну платформу, а відкритий код дозволяє користувачам модифікувати функціонал рушію під свої потреби та ділити своїми модифікаціями з іншими.

При розробці архітектури застосунку, самого застосунку та інтерактивних об'єктів симуляції, використовується версія Unity 3D 2019.

Наступна використана технологія - AR Foundation:

AR Foundation [22] - раніше – Vuforia, а зараз, після того, як Unity викупила розробку, технологія є комплексом декількох AR інструментів: ARKit, ARCore, Magic Leap та HoloLens. В розробці програмного продукту дослідження та під час моделювання його архітектури використовується лише ARCore.

ARCore – це крос платформна розробка, що дає для розробників вже готові технологічні рішення для роботи з AR, а саме комплекс інструментів для роботи системою розпізнавання та контролю симуляції за допомогою засобів Unity, без необхідності створення власного рішення.

ARCore XR plugin, дає такі можливості:

- Отримання та опрацювання даних з датчиків пристрою користувача, включаючи зображення з камери, в реальному часі.

- Контроль над системою спостереження: створення баз даних зображень-маркерів, їх пошук у просторі та оновлення сцени симуляції відносно змін стану об'єктів у реальному часі.

Отже, визначивши технології, які використовуються для розробки застосунку та можна описати архітектуру та його роботу:

Складові AR-застосунку:

- AR Session Origin – компонент, що містить основні налаштування роботи AR та його компонентів, таких, як AR Camera та AR Tracked Image Manager.

- AR Camera – компонент, що слугує цілі візуалізації сцени на екран пристрою користувача. Дозволяє користувачу коректно переміщатись відносно об'єктів симуляції, просто змінюючи своє положення та пристрою.

- AR Tracked Image Manager – компонент, що відповідає за контроль та налаштування системи розпізнавання у випадку роботи з зображеннями. Звертається до об'єкта з типом XRReferenceImageLibrary.

- XRReferenceImageLibrary – являє собою базу даних, що містить зображення для розпізнавання та їх налаштування (реальний розмір тощо).

- Інтерактивний об'єкт – відповідає за логіку роботи візуалізації фізичного явища.

Основний принцип та хід роботи застосунку, на прикладі попередньо змодельованої ситуації:

- Користувач запускає застосунок.

- Запускається AR Session Origin та проводить первинне налаштування системи розпізнавання та AR компонентів.

- Запускається компонент AR Camera та починає передавати інформацію з камери.

- Запускається компонент AR Tracked Image Manager, що контролює систему розпізнавання. Відправляючи запит до AR Camera та отримуючи дані про кожен кадр, Manager намагається розпізнати об'єкти, дані про яких він отримує від компонента XRReferenceImageLibrary.

- У випадку, коли шуканий об'єкт знайдений, запускається програма інтерактивного об'єкта, що стоїть відповідно до об'єкта маркера.

- Далі контроль симуляцією переходить до інтерактивного об'єкта.

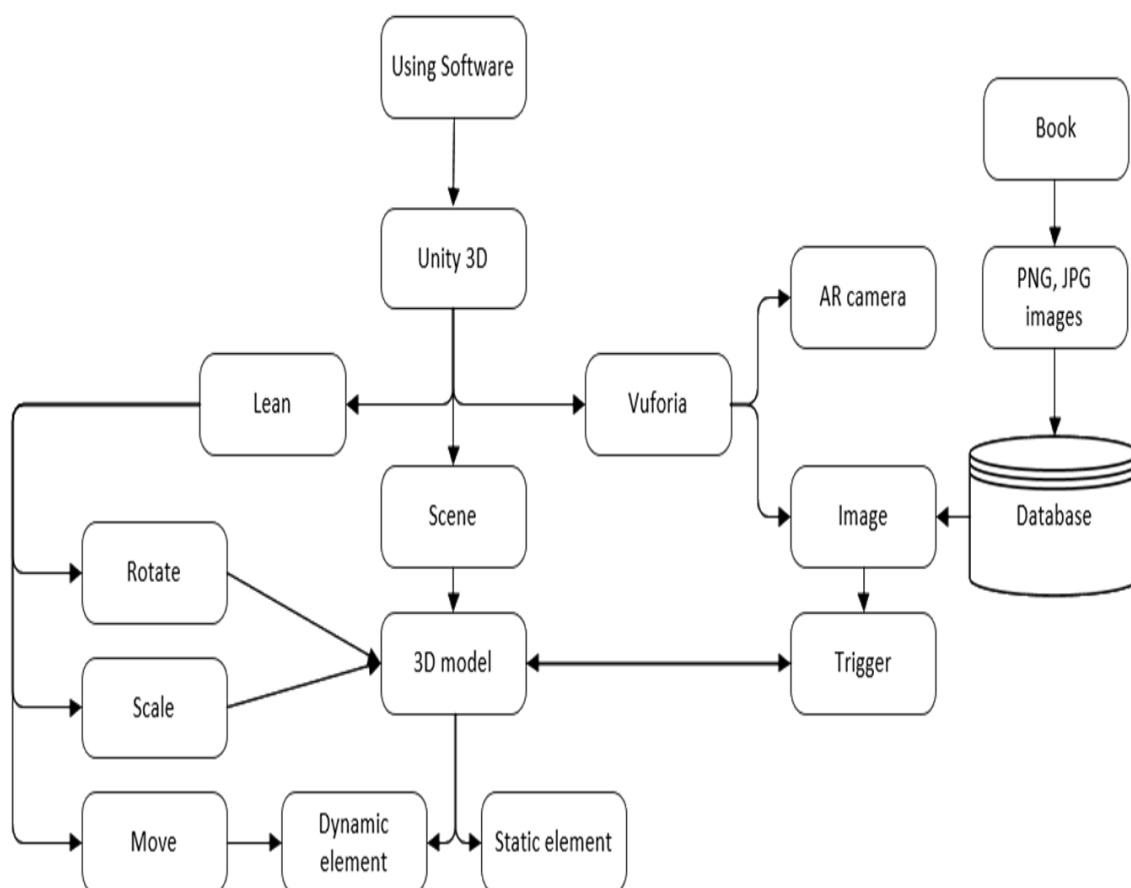


Рисунок 2.2 - Узагальнена схема архітектури AR застосунку

Визначивши, архітектуру AR-застосунку, ми можемо перейти до наступної частини розділу, а саме моделювання архітектури інтерактивного 3D-об'єкта симуляції та визначення моделі його роботи.

2.2. Проектування архітектури інтерактивного об'єкта доповненої реальності.

Для формування архітектури інтерактивного об'єкта (ІО) симуляції ми скористаємось способом, що ми використовували для архітектури застосунку, а саме моделювання ситуації в якій найхарактерніше проявиться робота ІО і можна буде поступово вичленити вимоги до нього.

За змодельовану ситуацію ми візьмемо лабораторну роботу з тематики, що була раніше вказана в огляді літератури, а саме «Сила струму. Вимірювання сили струму». Для цього візьмемо опис подібного лабораторного експерименту зі шкільного підручника з фізики за 8 клас [24]. В загальному вигляді суть експерименту є такою:

- В електричному колі послідовно з'єднані: джерело струму, споживач енергії (лампочка) та амперметр.
- До споживача струму паралельно приєднаний вольтметр.
- Задаючи різні параметри для сили опору споживача та напруги джерела учень повинен визначити силу струму та перевірити свої розрахунки за допомогою амперметра.

Тепер потрібно змодельовати ситуацію використання учнем AR-застосунку з ІО, що симулює роботу подібного експерименту. Отже, в загальному вигляді, хід роботи буде виглядати так:

- Розпізнавши ОМ застосунок запускає симуляцію відповідного ІО.

- Перед учнем з'являється тривимірна модель електричного кола, що складається з таких інтерактивних компонентів (ІК): амперметр, вольтметр та лампочка.

- Для контролю параметрами симуляції учню доступні такі елементи інтерфейсу: поля вводу напруги, силу опору лампочки та кнопка подачі/вимкнення напруги.

- Для моніторингу параметрів симуляції учню доступні такі елементи інтерфейсу: циферблати на амперметрі та вольтметрі.

- Задавши потрібні дані сили опору та напруги й ввімкнувши джерело струму, користувач може побачити як відповідно змінюються показники циферблатів та вмикається/вимикається лампочка.

- Користувач може змінювати дані в будь-який час, незалежно від поточного стану симуляції.

З цього опису ми можемо виділити такі вимоги до архітектури ІО:

- Повинен складатись з таких елементів: користувацький інтерфейс (КІ), об'єкти сцени (ОС) та контролер симуляції (КС).

- Користувач повинен задавати дані та керувати перебігом симуляції за допомогою КІ.

- КІ передає дані до КС, який контролює стан об'єктів сцени.

- ОС містить в собі всі об'єкти, що залучені в симуляції, як статичні, так і динамічні компоненти.

- Кожен компонент повинен керуватись одним КС, а не кожен окремо.

Тепер перейдемо до опису кожного з компонентів архітектури окремо:

1. Користувацький інтерфейс – компонент ІО, що відповідає за контроль та моніторинг даних симуляції. Користувач може

впливати на симуляцію лише за допомогою КІ. КІ складається з таких компонентів:

- Елементи вводу даних, що відповідають за зміну поточних даних симуляції. В нашому ІО до цих елементів відносяться:
 1. Поле вводу значення напруги джерела.
 2. Поле вводу значення опору споживача.
 3. Кнопки підтвердження для кожного з вище вказаних полів.
 4. Кнопка зміну стану ключа кола, що розмикає джерело струму.
- Елементи виведення даних, що відповідають за показ поточних даних симуляції. В нашому ІО до цих елементів відносяться:
 1. Циферблат амперметра.
 2. Циферблат вольтметра.
 3. Тіло лампочки, що змінює свій колір, в залежності від свого стану.

2. Об'єкти сцени (ОС) – це вся сукупність компонентів, що присутні під час симуляції на сцені та відповідає за візуальне зображення стану симуляції. ОС складається з таких компонентів:

- Елементи зображення симуляції, до яких відносяться:
 1. Елементи виводу даних КІ.
- Елементи контролю симуляції, до яких відносяться:
 1. Елементи вводу даних КІ.
 2. Контролер симуляції (КС).
- Статичні 3D об'єкти:
 1. Джерело струму.
 2. Дроти
 3. Клеми пристроїв.
 4. Основи тіл пристроїв.
- Динамічні 3D об'єкти:

1. Амперметр.
2. Вольтметр.
3. Лампочка.
4. Циферблат амперметра.
5. Циферблат вольтметра.

3. Контролер симуляції (КС) – ланка архітектури об'єкту, що з'єднує всі компоненти симуляції та контролює їх поведінку та хід роботи. КС складається з таких частин:

- Контролери об'єктів – відповідають за контроль динамічних компонентів.
- Контролери інтерфейсу – відповідає за роботу інтерфейсу.

Отже, визначивши складові архітектури ІО, ми можемо перейти безпосередньо до створення самого об'єкту.

2.3. Розробка інтерактивного об'єкта доповненої реальності.

Для демонстрації розробки ІО, ми створимо новий AR-застосунок та ІО, що буде візуалізувати певне фізичне явище. Прикладом останнього була вибрана демонстрація лабораторного експерименту «Вимірювання сили струму» з тематики «Сила струму». AR- застосунок та ІО будуть розроблені відповідно до методології описаної в попередніх розділах.

Перейдемо до розробки.

Хід роботи

Ще раз коротко визначимо використані інструменти:

- Ігровий рушій Unity 3D 2019.
- Плагін ARCore для Unity 3D, що входить до AR Foundation.
- Середовище розробки Visual Studio 2019.

- Мова програмування C#.

Опишемо компоненти застосунку, що треба розробити:

- Систему розпізнавання:
 1. Менеджер розпізнавання зображень.
 2. База даних об'єктів пошуку.
- Інтерактивний об'єкт:
 1. Статичні об'єкти:
 1. Джерело струму.
 2. Дроти.
 3. Клеми «мінус».
 4. Клеми «плюс».
 5. Тіло амперметра.
 6. Тіло вольтметра.
 7. Тіло лампочки.
 2. Динамічні об'єкти:
 1. Аналоговий циферблат амперметра.
 2. Цифровий циферблат амперметра.
 3. Аналоговий циферблат вольтметра.
 4. Цифровий циферблат вольтметра.
 5. Лампочка.
 3. Елементи інтерфейсу:
 1. Поле вводу напруги від джерела.
 2. Поле вводу сили опору споживача.
 3. Кнопка підтвердження значення напруги.
 4. Кнопка підтвердження значення опору.
 5. Кнопка контролю стану ключа розімкнення кола.
 4. Контролер симуляції.

Завершивши опис компонентів перейдемо до розробки:

- Для початку встановимо компонент AR Core XR Plugin версії 2.1.1. до нашого Unity проєкту, що дає нам змогу використовувати AR можливості пристроїв на платформі Android 7 Nougat та вище. Разом з AR Core автоматично встановлюються також компоненти: AR Foundation та AR Subsystem.

- Проведемо первинні налаштування Build Settings нашого застосунку: а саме вкажемо платформу застосунку в Build Settings, як Android. Перейдемо до вікна «Player Settings» в Build Settings та на вкладці «Player» вберемо галочку з пункту «Multithreaded rendering» та виберемо «Android 7.0 ‘Nougat (API level 24)’» в пункті «Minimum API level». На вкладці XR виберемо «Optional» в пункті AR Core/Requirement.

- Створимо нову сцену та дамо їй довільну назву.

- Додаємо два необхідні компоненти для роботи з AR до нашої сцени: AR Session Origin та AR Session.

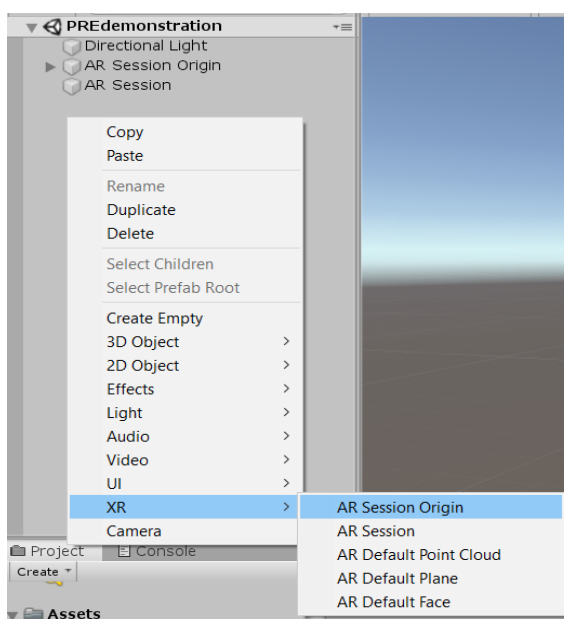


Рисунок 2.4 - AR Session Origin та AR Session

- Далі нам треба створити основу системи розпізнавання: додамо до AR Session Origin компонент AR Tracked Image Manager. Для подальшої роботи компонента нам потрібно вказати бібліотеку шуканих зображень («Reference Library»). Створимо її.

- Створимо новий об'єкт PRE_RL типу Reference Library та додамо до неї нове зображення

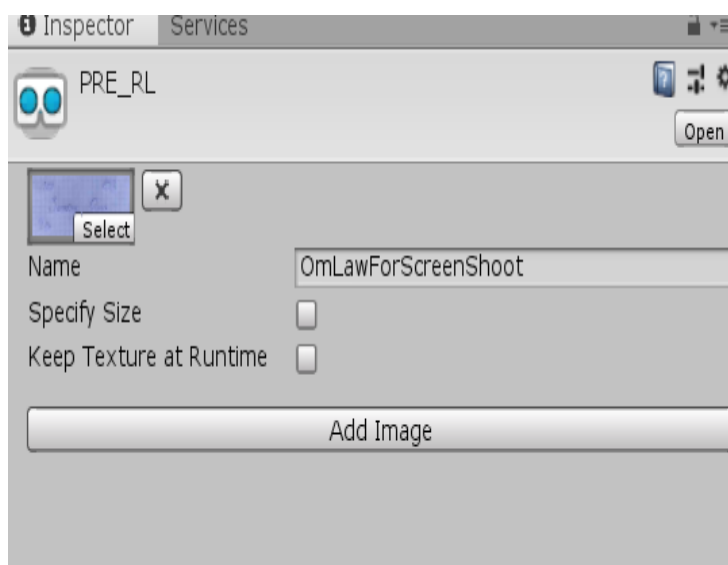


Рисунок 2.5 - Reference Library

- Як шукане зображення ми будемо використовувати текстуру, що була створена з фото реального об'єкта

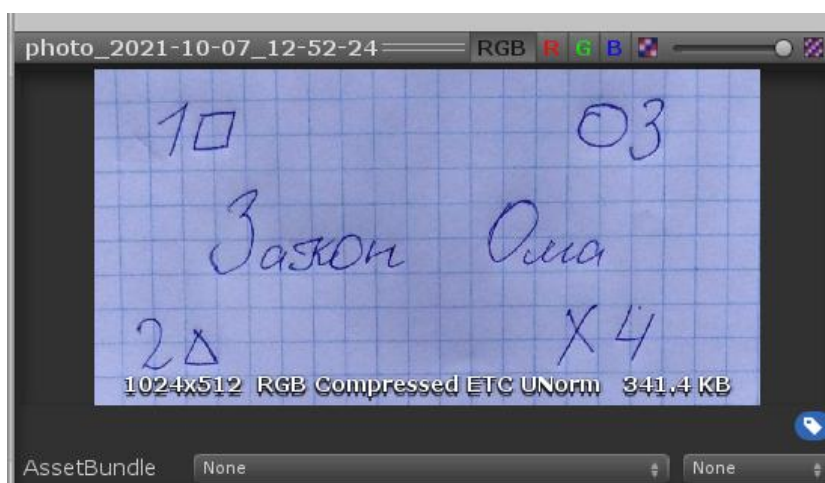


Рисунок 2.6 - Текстура шуканого зображення

- На цьому первинне налаштування загальної архітектури застосунку завершено. Перейдемо до розробки інтерактивного об'єкта.

- Створимо новий префаб OmLawNew, що є актором інтерактивного об'єкта.

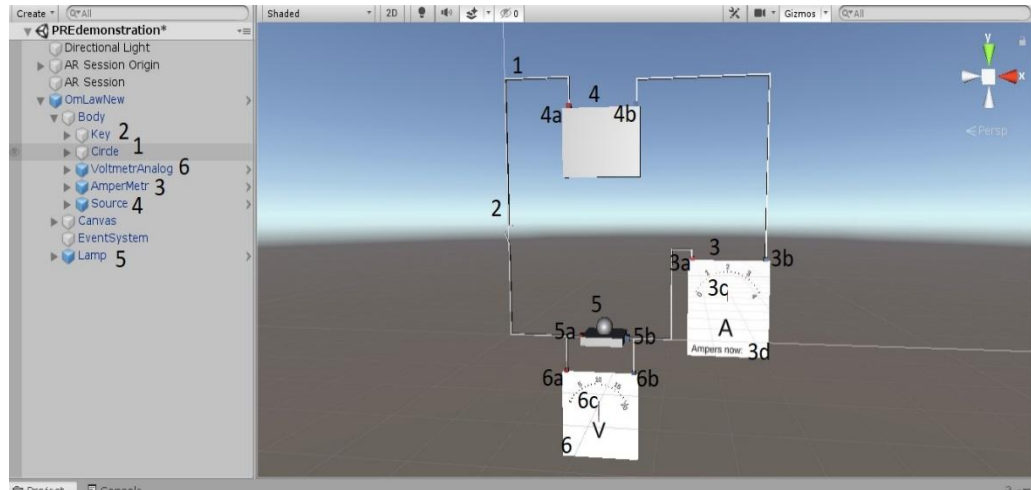


Рисунок 2.7 - Структура інтерактивного об'єкта

- Створимо компонент Circle (Рисунок 2.7, позначка 1) - статичний об'єкт (static mesh), що виконує лише візуальну роль: зображає з'єднання дротів в електричному колі. Розміщення та кріплення об'єктів в колі, за допомогою дротів, повторює реальні правила розміщення: кожен об'єкт має клему «плюс» до якої приєднаний провід та вихідну клему «мінус», об'єкти з'єднані послідовно або паралельно так же само як і в реальному колі.

- Створимо компонент Key (Рисунок 2.7, позначка 2) - динамічний об'єкт, що змінює свою форму в залежності від стану ключа ввімкнення/розімкнення кола. Є візуальним зображенням останнього в симуляції.

- Створимо компонент AmperMetr (Рисунок 2.7, позначка 3) - динамічний об'єкт, що є візуальним зображенням амперметра в симуляції. Має такі складові частини як: клему «плюс» (Рисунок 2.7, позначка 3a) для вхідного сигналу струму, клему «мінус» (Рисунок 2.7,

позначка 3b) для вихідного сигналу, аналоговий циферблат (Рисунок 2.7, позначка 3c) для показу поточної сили струму – стрілка вказує на відповідну цифру на циферблаті, цифровий циферблат (Рисунок 2.7, позначка 3d), що виконує ту же роль що й аналоговий, але показує точне значення сили струму.

- Source (Рисунок 2.7, позначка 4) - статичний об'єкт, що є візуальним зображенням джерела струму в симуляції. Має клему «плюс» (Рисунок 2.7, позначка 4a) для вихідного сигналу та клему «мінус» (Рисунок 2.7, позначка 4b) для вхідного сигналу.

- Створимо компонент Lamp (Рисунок 2.7, позначка 5) - динамічний об'єкт, що є візуальним зображенням споживача струму (лампочка) в симуляції. Змінює власний колір та освітлення, якщо отримує напругу на вхідну клему (Рисунок 2.7, позначка 5a). Також має вихідну клему «мінус» (Рисунок 2.7, позначка 5b).

- Створимо компонент VoltmetrAnalog (Рисунок 2.7, позначка 6) - динамічний об'єкт, що є візуальним зображенням вольтметра в симуляції. Має такі складові частини як: клему «плюс» (Рисунок 2.7, позначка 6a) для вхідного сигналу струму, клему «мінус» (Рисунок 2.7, позначка 6b) для вихідного сигналу, аналоговий циферблат (Рисунок 2.7, позначка 6c) для показу поточної сили опору провідника – стрілка вказує на відповідну цифру на циферблаті.

- Почнемо створювати інтерфейс інтерактивного об'єкта.

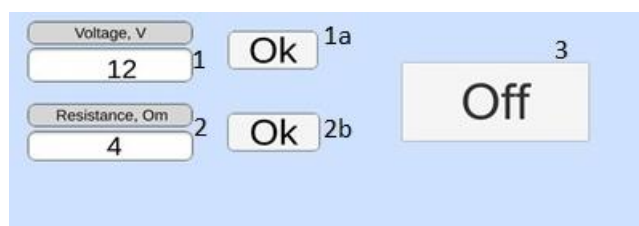


Рисунок 2.8 - Інтерфейс контролю симуляції

- Створимо компонент Canvas, що слугує основою інтерфейсу.

- Створимо системний компонент EventSystem, що дозволяє опрацьовувати події взаємодії з інтерфейсом.

- Створимо об'єкт Napruqa (Рисунок 2.8, позначка 1, 2.9. Позначка 1) типу InputField, що відповідає за поле вводу значення напруги.



Рисунок 2.9 - Структура об'єкта Canvas

- Створимо об'єкт Ok (Рисунок 2.8, позначка 1а, Рис. 2.9 Позначка 2) типу Button, що відповідає за підтвердження вводу значення напруги.

- Створимо об'єкт Resistance (Рисунок 2.8, позначка 2, Рисунок 2.9, позначка 3) типу InputField, що відповідає за поле вводу значення опору споживача.

- Створимо об'єкт Ok (Рисунок 2.8, позначка 2б, Рисунок 2.9, позначка 4) типу Button, що відповідає за підтвердження вводу значення напруги.

- Створимо об'єкт On/Off (Рисунок 2.8, позначка 3, Рисунок 2.9, позначка 5) типу Button, що відповідає за зміну стану ключа (Key)

- Розпочнемо створення об'єкту контролера симуляції.

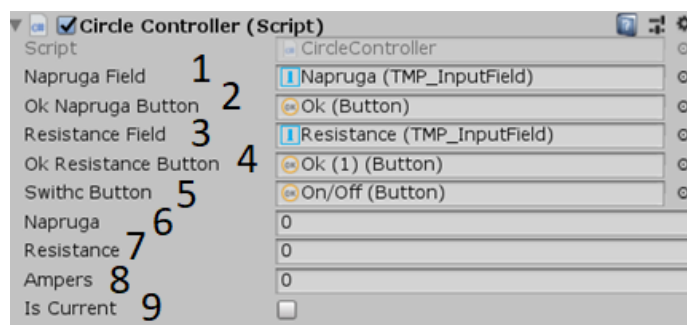


Рис. 2.10 - Структура компонента контролера

- Об'єкт контролера симуляції складається з таких частин:

1. Napruga Field (Рисунок 2.10, позначка 1) – поле, що відповідає за посилання на частину інтерфейсу Napruga.
2. Ok Napruga Button (Рисунок 2.10, позначка 2) - поле, що відповідає за посилання на частину інтерфейсу Ok, яке відповідає за підтвердження вводу даних в полі Napruga.
3. Resistance Field (Рисунок 2.10, позначка 3) – поле, що відповідає за посилання на частину інтерфейсу Resistance.
4. Ok Resistance Button (Рисунок 2.10, позначка 4) - поле, що відповідає за посилання на частину інтерфейсу Ok(1), яке відповідає за підтвердження вводу даних в полі Resistance.
5. Switch Button (Рисунок 2.10, позначка 5) – поле, що відповідає за посилання на частину інтерфейсу On/Off.
6. Napruga (Рисунок 2.10, позначка 6) – поле, що відповідає за збереження поточного значення сили напруги в колі.
7. Resistance (Рисунок 2.10, позначка 7) – поле, що відповідає за збереження поточного значення сили опору провідника споживача струму.
8. Ampers (Рисунок 2.10, позначка 8) – поле, що відповідає за збереження поточного значення сили струму в колі.
9. Is Current (Рисунок 2.10, позначка 9) – поле, що відповідає за збереження поточного значення стану кола: розімкнуте чи ні.

- На цьому розробка компонентів ІО завершена.

- Тепер створимо компонент Inst Obj, що працює з СР та запускає ІО у випадку, якщо ОМ був знайдений.

- Компонент Ins Obj складається з таких компонентів:

1. Масив об'єктів для розміщення. З цього масиву беруться дані про зображення та йде пошук відповідності знайдених

зображень та зображень з PRE_RL (бібліотеки зображень пошуку)

2. Кількість зображень до пошуку.
3. ІО, що стоїть відповідно до зображення в PRE_RL.

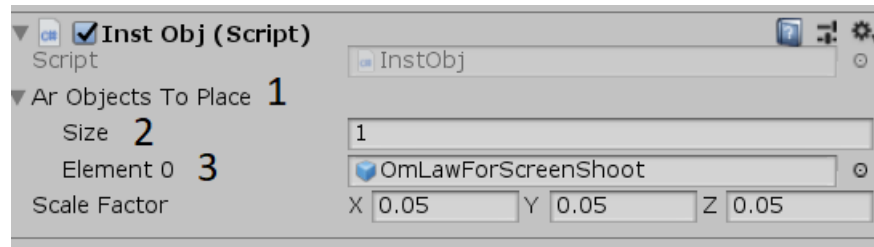


Рисунок 2.11 - Структура Inst Obj

- На цьому розробка нашого AR застосунку завершена.

Кінцевий результат ІО в редакторі Unity виглядає так:

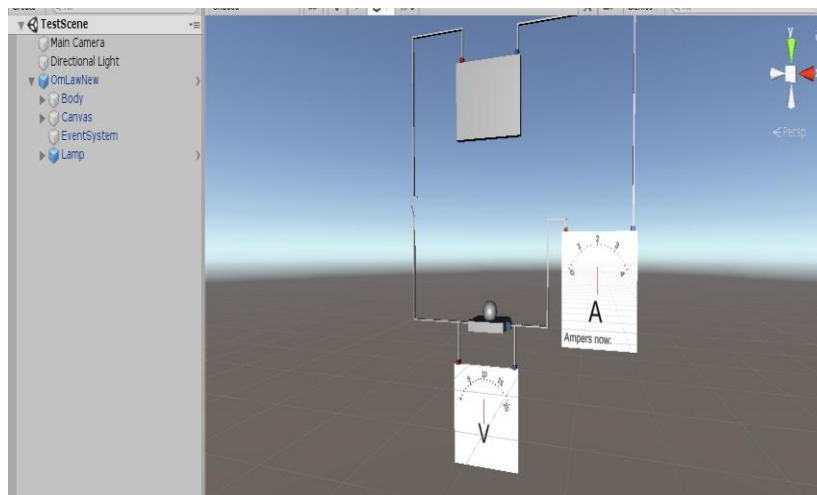


Рисунок 2.12 - Кінцевий результат вигляду ІО в редакторі Unity

Приклад, вигляду ІО під час симуляції:

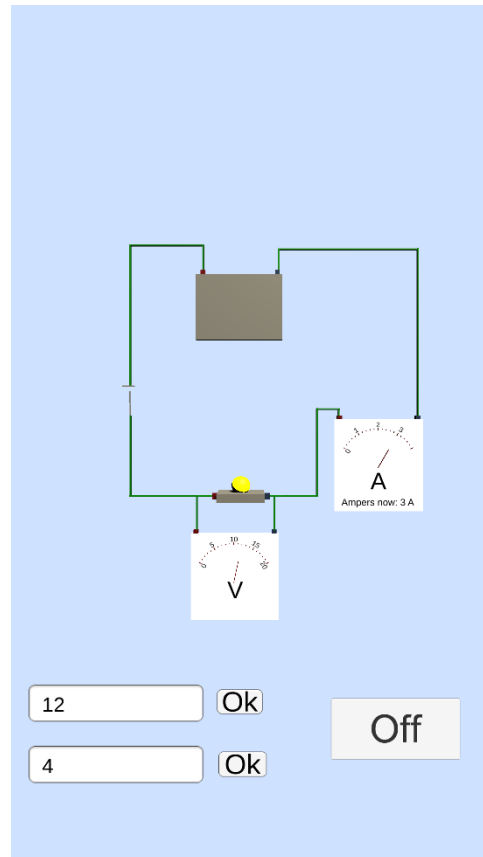


Рисунок 2.13 - Кінцевий результат вигляду ІО під час симуляції.

Далі вказано посилання на код з поясненнями в додатках, для кожного скрипту, кожного елементу:

- Inst Obj - Додаток А (Код А.1).
- CircleController - Додаток А (Код А.2).
- KeyAnalog - Додаток А (Код А.3).
- Ampermetr Analog - Додаток А (Код А.4).
- Lamp – Додаток А (Код А.5)

ВИСНОВКИ

Розробка нових методів візуалізації матеріалу при вивченні природничих наук у закладах середньої освіти є достатньо пріоритетним напрямком. Розглянута технологія AR-симуляцій є напрочуд доцільним інструментом розв’язання проблеми недостатньої наочності викладання у таких дисциплінах, як фізика. Тому й створення інтерактивних 3D

об'єктів для подібних є настільки ж актуальним, як і сама технологія AR. Ми описали методику моделювання архітектури інтерактивних об'єктів та їх розробки.

Зараз вже є декілька програмних продуктів, що використовується закладами середньої освіти у США та Європі. Викладачі та учні, в більшості випадків, використовують їх, як доповнення до класичних методів візуалізації матеріалу.

Розроблений інтерактивний об'єкт можна використовувати, як основу для створення інших подібних та поступового вдосконалення вже наявного застосунку AR-симуляції. Ми плануємо вдосконалити поточний застосунок для використання не тільки для ілюстрації фізичних процесів, а й з суміжних дисциплін: хімія, біологія тощо.

- Було розглянуто літературу на предмет розробки та використання інтерактивних 3D об'єктів доповненої реальності у процесі навчання в закладах середньої освіти.

- Було розглянуто програмні реалізації AR-застосунків, що використовують подібні об'єкти

- Було створено архітектуру AR-застосунку, що використовує інтерактивні об'єкти.

- Було створено архітектуру інтерактивних об'єктів, що використовують для симуляції фізичних явищ та законів.

- Була описана методологія моделювання архітектури та створення: AR-застосунку, що візуалізує фізичне явище; інтерактивний об'єкт, що симулює роботу фізичного явища.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. “About One Approach to Building Systems for Testing Physical Knowledge”. Michail Lvov, Sergiy Kuzmenkov, Hennadiy Kravtsov. ICTERI 2019. Part I: 4th International Workshop on Professional Retraining and Life-Long Learning using ICT: Person
2. Augmented Reality // Investopedia: [Веб-сайт]. 2020. URL: <https://www.investopedia.com/terms/a/augmented-reality.asp> (дата звернення: 19.10.2021).
3. Azuma R. A Survey of Augmented Reality // Presence: Teleoperators and Virtual Environments. 1997., вип. 1 Т. 3. С. 355-385.
4. Dichotomy of the 'clip thinking' phenomenon // ICEPS 2017 . 2017. URL: https://www.researchgate.net/profile/Tatiana-Volkodav/publication/320225378_Dichotomy_of_the_'Clip_Thinking'_Phenomenon/links/5a02e6ae6fdcc6b7c99149b/Dichotomy-of-the-Clip-Thinking-Phenomenon.pdf (дата звернення: 19.10.2021).
5. Digital 2021: Global Overview Report // DATEREPORTAL: [Веб-сайт]. URL: <https://datareportal.com/reports/digital-2021-global-overview-report> (дата звернення: 13.10.2021).
6. Google Arts&Culture // Google: [Веб-сайт]. 2021. URL: <https://artsandculture.google.com/project/expeditions> (дата звернення: 20.10.2021).
7. Hennel P. , Schmidt-Kraepelin M. , Van den Eynden S. Benefits of Augmented Reality in Educational Environments – A Systematic Literature Review // Internationale Tagung Wirtschaftsinformatik. Osnabrück, 2015. Т. 12. С. 1542-1556.
8. Information explosion // New World Encyclopedia: [Веб-сайт]. URL: https://www.newworldencyclopedia.org/entry/Information_explosion (дата звернення: 14.10.2021).

9. J. Chelikowski, «Introduction: Silicon in all its Forms», Silicon: evolution and future of a technology (Editors: P. Siffert, E. F. Krimmel), p.1, Springer, 2004 ISBN 3-540-40546-1.

10. Lee J. Impagliazzo J. History of computing in education. Toulouse: Springer, 2006. 172 с. ISBN 1-4020-8135-9.

11. Learn Science, Master STEM, Be Future Ready // MERGE EDU: [Веб-сайт]. 2021. URL: <https://mergeedu.com/> (дата звернення: 20.10.2021).

12. Mars is a real place. Virtual Engineering by DrashVR LLC., [Веб-сайт] <http://www.titansofspacevr.com/marsisarealplace.html>

13. Real men program in C // Embedded Systems Design. TechInsights (United Business Media):[Веб-сайт]. URL: <https://www.embedded.com/real-men-program-in-c/> (дата звернення: 13.10.2021).

14. Seemiller C. , Grace M. Generation Z Goes to College: книга. : John Wiley & Son, 2016. 1 с.

15. Šmihula D. Long waves of technological innovations // Studia Politica Slovaca. 2011., вип. 2 Т. 4. С. 50-68.

16. The Lengthy History of Augmented Reality // Huffington Post: [Веб-сайт]. 2016. URL: https://images.huffingtonpost.com/2016-05-13-1463155843-8474094-AR_history_timeline.jpg (дата звернення: 13.10.2021).

17. Unity, платформа розробки в реальному часі // Unity: [Веб-сайт]. 2021. URL: <https://unity.com/ru> (дата звернення: 16.10.2021).

18. What is Virtual Reality? [Definition and Examples] // MARXENT 3D COMMERCE: [Веб-сайт]. 2020. URL: <https://www.marxentlabs.com/what-is-virtual-reality/> (дата звернення: 19.10.2021).

19. Інформатизація суспільства та проблема «кліпового мислення» // КПІ ім. Ігоря Сікорського: [Веб-сайт]. Київ. URL: <https://kpi.ua/1102-7> (дата звернення: 13.10.2021).

20. Методи та технології віртуальної та доповненої реальності в комп'ютерних програмах навчального призначення // DSpace Repository: [Веб-сайт]. Херсон, 2020. URL: <http://ekhsuir.kspu.edu/handle/123456789/13481;jsessionid=2E765A0259A7A68C3A104ADA1C4C7A94> (дата звернення: 15.10.2021).

21. Мультимедійна віртуальна лабораторія по фізиці в системі дистанційного навчання / Е. О. Козловський, Г. М. Кравцов // Інформаційні технології в освіті. - 2014. - Вип. 18. - С. 80-89. http://nbuv.gov.ua/UJRN/itvo_2014_18_11

22. Платформа Unity AR Foundation // Unity: [Веб-сайт]. 2021. URL: <https://unity.com/ru/unity/features/arfoundation> (дата звернення: 16.10.2021).

23. Успешность визуализации информации в процессе обучения // Pedagogical Review. 2013., вип. 2 Т. 2. С. 30-35.

24. Фізика : підруч. для 8 кл. загальноосвіт. навч. закл. / [В. Г. Бар'яхтар, Ф. Я. Божинова, С. О. Довгий, О. О. Кірюхіна] ; за ред. В. Г. Бар'яхтара, С. О. Довгого. — Х. : Вид-во «Ранок», 2016. — 240 с. : іл., фот. ; ISBN 978-617-09-2855-9

ДОДАТКИ

Додаток А

Код (А.1 частина 1)

```

using System.Collections; // Підключаєм необхідні бібліотеки Unity
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.ARFoundation;

[RequireComponent(typeof(ARTrackedImageManager))]//Додаєм вимогу до компоненту CP
public class InstObj : MonoBehaviour //Основний клас
{
    [SerializeField]
    private GameObject[] arObjectsToPlace; //Додаєм масив IO

    [SerializeField]
    private Vector3 scaleFactor = new Vector3(0.1f,0.1f,0.1f);

    private ARTrackedImageManager m_TrackedImageManager; //Посилання на CP

    private Dictionary<string, GameObject> arObjects = new Dictionary<string, GameObject>();
//Посилання на словник IO
    private void Awake() //Робота при запуску IO
    {
        m_TrackedImageManager = GetComponent<ARTrackedImageManager>(); //Отримуємо посилання на CP
    }
    private void OnEnable()// Робота при активації CP
    {
        m_TrackedImageManager.trackedImagesChanged += OnTrackedImageChanged;//Змінюєм зображення для
відстеження
    }
    private void OnDisable()// Робота при деактивації CP
    {
        m_TrackedImageManager.trackedImagesChanged -= OnTrackedImageChanged;//Видаляєм зображення
для відстеження
    }
    void OnTrackedImageChanged(ARTrackedImagesChangedEventArgs eventArgs)//Опрацьовуєм OnEnable або
OnDisable
    {
        foreach (ARTrackedImage trackedImage in eventArgs.added)//для кожного зображення, що
розпізнано
        {
            foreach (GameObject arObject in arObjectsToPlace)
            {
                GameObject newARObject = Instantiate(arObject, Vector3.zero,
Quaternion.identity);//Завантажуєм IO
                newARObject.name = arObject.name;//Даємо IO ім'я для відстеження
                arObjects.Add(arObject.name, newARObject);//Додаємо до масиву доданих IO в сцені
            }
            UpdateARImage(trackedImage);//Оновлюємо CP
        }
        foreach (ARTrackedImage trackedImage in eventArgs.updated)//для кожного зображення, що
змінює свої параметри
        {
            UpdateARImage(trackedImage);//Оновлюємо CP
        }
        foreach (ARTrackedImage trackedImage in eventArgs.removed)//для кожного зображення, що
більше не розпізнається
        {
            arObjects[trackedImage.name].SetActive(false);//Видаляємо IO, що відповідає цьому
зображенню
        }
    }
}

```


Код (А.1 частина 2)

```

private void UpdateARImage(ARTrackedImage trackedImage)//Процедура оновлення CP
{
    AssignGameObject(trackedImage.referenceImage.name, trackedImage.transform.position);//Перезаписуєм
    положення об'єкта
}
private void AssignGameObject(string name, Vector3 newPosition)//Процедура перезаписування об'єкта
{
    if(arObjectsToPlace != null)//Якщо об'єкт існує, то
    {
        arObjects[name].SetActive(true); //Актувати його
        arObjects[name].transform.position = newPosition; // Змінює позицію на нову задану
        arObjects[name].transform.localScale = scaleFactor;
        foreach(GameObject go in arObjects.Values)//Для всіх об'єктів,...
        {
            if (go.name != name)//...що не є поточним...
            {
                go.SetActive(false); //деактивувати їх
            }
        }
    }
}
}
}

```

Код (А.2 частина 1)

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine; // Підключаєм необхідні бібліотеки Unity
using UnityEngine.UI;
using TMPro;

public class CircleController : MonoBehaviour //Тіло класу
{
    [SerializeField]
    private TMP_InputField NapruqaField; //Посилання на поле вводу напруги
    [SerializeField]
    private Button OkNapruqaButton; //Посилання на кнопку підтвердження вводу напруги
    [SerializeField]
    private TMP_InputField ResistanceField; //Посилання на поле вводу опору
    [SerializeField]
    private Button OkResistanceButton; //Посилання на кнопку підтвердження вводу опору
    [SerializeField]
    private Button SwithcButton; //Посилання на кнопку переключення стану ключа

    public int napruqa = 0; //Значення напруги в контролері
    public int resistance = 0; //Значення напруги в опору
    public int Ampers = 0; //Значення сили струму в контролері
    public bool isCurrent = false; //Значення стану кола
    private int EnteredVolts = 0; //Значення введеної напруги
    private int EnteredResistance = 5; //Значення введеного опору
    private void Awake()//робота при запуску елемента
    {
        UpdateAmpers(); //Оновлюємо силу струму
        var fieldNapruqa = new TMP_InputField.SubmitEvent(); //Посилання на поле вводу напруги
        fieldNapruqa.AddListener(DebugVolts);
        NapruqaField.onEndEdit = fieldNapruqa; //Додаєм процедуру оновлення значення напруги при
        редагуванні поля вводу
        OkNapruqaButton.onClick.AddListener(ConfirmVolts); //Процедура підтвердження вводу напруги
    }
}

```

Код (А.2 частина 2)

```

var fieldR = new TMP_InputField.SubmitEvent();//Посилання на поле вводу опору
fieldR.AddListener(DebugResistance);
ResistanceField.onEndEdit = fieldR;//Додаємо процедуру оновлення значення опору при редагуванні поля
вводу
OkResistanceButton.onClick.AddListener(ConfirmResistance);//Процедура підтвердження вводу напруги
SwitchButton.onClick.AddListener(UpdateKey);//Процедура оновлення стану кола
}

private void Update()//Робота на кожен кадр
{
    UpdateAmpers();//Оновлюємо силу струму
}
private void UpdateKey()//Функція оновлення стану ключа
{
    isCurrent = !isCurrent; //Змінюємо значення на протилежне
    Debug.Log("Status: " + isCurrent);
}
private void UpdateAmpers()//Функція оновлення значення сили струму
{
    if(napruha == 0 | resistance == 0)//Якщо опір або напруга дорівнює нулю
    {
        Ampers = 0;//то сила струму нульова
    }
    else//інакше
    {
        Ampers = napruha / resistance;//Звичайна формула визначення сили струму
    }
}
private void DebugVolts(string arg0)//Функція зчитування введеного значення напруги
{
    EnteredVolts = int.Parse(arg0);//Парсим введене значення
    Debug.Log("Entered" + arg0);
}
private void ConfirmVolts()//Функція остаточного підтвердження напруги
{
    Debug.Log("Entered before confirm " + EnteredVolts );
    napruha = EnteredVolts;//Поточна напруга рівна введеному значенню
    Debug.Log("Assigned " + napruha);
}
private void DebugResistance(string arg1)//Функція зчитування введеного значення опору
{
    Debug.Log("Entered" + arg1);
    EnteredResistance = int.Parse(arg1);//Парсим введене значення
    Debug.Log("To go" + EnteredResistance);
}
private void ConfirmResistance()//Функція остаточного підтвердження опору
{
    Debug.Log("Entered before confirm " + EnteredResistance);
    resistance = EnteredResistance;//Поточний опір дорівнює введеному значенню
    Debug.Log("Assigned " + resistance);
}

```

Код (А3 частина 1)

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;//Підключаємо необхідні бібліотеки Unity
using UnityEngine.UI;
using TMP;
public class KeyAnalog : MonoBehaviour //Тіло класу
{
    [SerializeField]
    private GameObject CircleObj; //Посилання на об'єкт кола

    [SerializeField]
    private GameObject KeyObject; //Посилання на об'єкт ключа
}

```

Код (А3 частина 2)

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine; // Підключаєм необхідні бібліотеки Unity
using UnityEngine.UI;
using TMPPro;
public class KeyAnalog : MonoBehaviour //Тіло класу
{
    [SerializeField]
    private GameObject CircleObj; //Посилання на об'єкт кола

    [SerializeField]
    private GameObject KeyObject; //Посилання на об'єкт ключа
    private int AnimStage = 0; //Поточний стан анімації ключа
    private bool cStatus = false; //Стан ключа
    private CircleController cController; //Посилання на контролер кола
    /*
    0 - default
    1 to of
    2 to on
    */
    Animator KeyAnimator; //Створюємо новий об'єкт анімації
    private void Awake() //робота при запуску елемента
    {
        UpdateIsCurrent(); //Оновлюємо стан кола
        if (KeyObject != null) //Якщо об'єкт ключа існує...
        {
            KeyAnimator = KeyObject.GetComponent<Animator>(); //...отримуємо аніматор цього об'єкту
        }
    }
    private void Update() //Робота на кожен кадр
    {
        UpdateIsCurrent(); //Оновлюємо стан кола
        if (cStatus == true) //Якщо коло ввімкнено
        {
            doWork(); //Запускаємо роботу
        }
        else //інакше
        {
            resetWork(); //Скидуємо результати роботи
        }
    }
    private void doWork() //Функція роботи
    {
        UpdateKey(); //Оновлюємо ключ
    }
    private void resetWork() //Функція скидування результату роботи
    {
        UpdateKey(); //Оновлюємо ключ
    }
    private void UpdateIsCurrent() //Функція оновлення статусу кола в ключі
    {
        cController = CircleObj.GetComponent<CircleController>(); //Отримуємо компонент контролера кола
        cStatus = cController.isCurrent; //Оновлюємо статус кола в ключ, беручи це ЗНАЧЕННЯ з
        контролера
    }
    private void UpdateKey() // Функція оновлення стану ключа
    {
        if (cController.napruga > 0) //Зміна анімації ключа в залежності від поточного стану та статусу кола
        {
            if (AnimStage == 0 & cStatus == false)
            {
                KeyAnimator.SetTrigger("DefaultToOff");
                AnimStage = 1;
            }
        }
    }
}

```

Код (А.3 частина 3)

```

        if (AnimStage == 0 & cStatus == true)
        {
            KeyAnimator.SetTrigger("DefaultToOn");
            AnimStage = 2;
        }
        if (AnimStage == 1 & cStatus == true)
        {
            KeyAnimator.SetTrigger("OffToOn");
            AnimStage = 2;
        }
        if (AnimStage == 2 & cStatus == false)
        {
            KeyAnimator.SetTrigger("OnToOff");
            AnimStage = 1;
        }
    }
    if (cController.napruha <= 0)
    {
        if (AnimStage == 2 & cStatus == true)
        {
            KeyAnimator.SetTrigger("OnToOff");
            AnimStage = 1;
        }
    }
}

```

Код (А.4 частина 1)

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine; // Підключаєм необхідні бібліотеки Unity
using UnityEngine.UI;
using TMPro;

public class AmpermetrAnalog : MonoBehaviour //Тіло класу
{
    [SerializeField]
    private GameObject ControllerObject; //Посилання на об'єкт контролера симуляції
    [SerializeField]
    private TextMeshPro countB; //
    public Transform armTransform; //Положення стрілки циферблату
    private CircleController cController; //Посилання на контролер симуляції
    private int CountAmper = 0; //Значення сили струму
    private float MinDegree = 60; //Мінімальний кут стрілки циферблату (позначка нуль на циферблаті)
    private float CurrDegree = 0; //Поточний кут стрілки
    private float DegreeTo = 0; //Значення кута до якого треба повернути стрілку
    private float NextDegree = 0; //Значення кута стрілки при наступній спробі повороту
    private float stepOfAngle = 6f; //Крок повороту
    private bool cStatus = false; //Стан кола
    private void Awake() //робота при запуску елемента
    {
        UpdateIsCurrent(); //Оновлюємо статус кола
        CurrDegree = MinDegree; //Установлюємо положення стрілки на мінімальне
        armTransform.localRotation = Quaternion.Euler(0, 0, CurrDegree); //Повернемо стрілку до поточного
кута
    }
    private void Update() //Робота на кожен кадр
    {
        UpdateIsCurrent(); //Оновлюємо статус кола
        UpdateCountText(); //Оновлюємо цифровий циферблат
        if (cStatus == true) //Якщо коло ввімкнуте, то
        {
            Debug.Log("Its work " + cStatus);
            doWork(); //Запускаєм роботу
        }
    }
}

```

Код (А.4 Частина 2)

```

else//Інакше
{
    Debug.Log("Its not work " + cStatus);
    resetWork();//Зупиняем роботу
}
}
private void doWork()//Функція роботи
{
    updateCount();//Оновлюємо значення ампер
    UpdateCountText();//Оновлюємо текст циферблату
    updateDegreeTo();//Визначаємо до якого кута треба повернути стрілку
    updateNextDegree();//Визначаємо наступний кут повороту
    UpdateArmPointer();//Оновлюємо стрілку
}
private void resetWork()//Функція зупинки роботи
{
    CountAmper = 0;//Скидуємо значення струму до нуля
    UpdateCountText();//Оновлюємо текст циферблату
    updateDegreeTo();//Визначаємо до якого кута треба повернути стрілку
    updateNextDegree();//Визначаємо наступний кут повороту
    UpdateArmPointer();//Оновлюємо стрілку
}
private void UpdateIsCurrent()//Функція оновлення статусу кола
{
    cController = ControllerObject.GetComponent<CircleController>();//отримуємо компонент контролера з
об'єкту, що містить контролер
    cStatus = cController.isCurrent; //Привласнюєм значення з отриманого компоненту
}
private void updateCount()//Функція оновлення значення сили струму
{
    cController = ControllerObject.GetComponent<CircleController>();//отримуємо компонент контролера з
об'єкту, що містить контролер
    CountAmper = cController.Ampers;//Привласнюєм значення з отриманого компоненту
}
private void updateDegreeTo()//Функція визначення кута на який треба повернути стрілку
{
    if (CountAmper == 0)//Якщо немає струму
    {
        DegreeTo = MinDegree;//Кут рівен мініальному
    }
    if (CountAmper == 2)//Якщо ампер = 2
    {
        DegreeTo = 0;//Стрілка повинна стати в своє стандартне положення - вказувати ввєрх
    }
    if(CountAmper < 2 | CountAmper > 2)// Якщо значення коливається в діапазоні від 0 до 4
    {
        DegreeTo = MinDegree - ((CountAmper*5) * stepOfAngle);/*Кут рівен = Мініальному - ((кількості
ампер -
        -* на кількість поділок циферблату) * на крок стрілки)*/
    }
    if (CountAmper > 4)//Якщо значення ампер перевищує максимальне значення циферблату
    {
        DegreeTo = MinDegree - (20 * stepOfAngle);//Кут буде рівним максимально можливому
    }
}
private void updateNextDegree()//Функція знаходження наступного кута для повороту
{
    if(CurrDegree > DegreeTo)//Якщо поточний кут більше наступного
    {
        NextDegree = CurrDegree - 1;//Наступний буде на градус менше за поточний
    }
    if (CurrDegree < DegreeTo)//Якщо поточний кут менше наступного
    {
        NextDegree = CurrDegree + 1;//Наступний буде на градус більше за поточний
    }
}
}

```

Код (А.4 Частина 3)

```

private void UpdateArmPointer()//Оновлюємо положення стрілки
{
    if (CurrDegree != DegreeTo)//Якщо поточний кут не рівний кут на який треба повернути
    {
        armTransform.localRotation = Quaternion.Euler(0, 0, NextDegree);//Повертаєм стрілку на
наступний кут
        CurrDegree = NextDegree;//Поточний кут тепер рівний наступному
    }
}
private void UpdateCountText()//Функція оновлення тексту циферблату
{
    string newText;
    TextMeshPro nText = countB.GetComponentInChildren<TextMeshPro>();//Отримуємо компонент тексту з
об'єкту тексту
    newText = CountAmper.ToString();//Конвертуємо значення ампер в текстове значення
    nText.text = "Amps now: " + newText + " A";//Присвоюємо це значення до значення тексту
}

```

Код (А.5 частина 1)

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine; // Підключаєм необхідні бібліотеки Unity
using UnityEngine.UI;
using TMPPro;

public class Pulp : MonoBehaviour //Тіло класу
{
    [SerializeField]
    private GameObject CircleObj; //Посилання на об'єкт, що містить контролер симуляції
    [SerializeField]
    private GameObject PulpObject; //Посилання на об'єкт лампочки
    private int AnimStage = 0;//Поточний етап анімації лампочки
    private bool cStatus = false;//Стан кола
    private CircleController cController;//Посилання на контролер симуляції
    /*
    0 - default
    1 to of
    2 to on
    */
    Animator PulpAnimator;//Створюємо об'єкт анімацій

    private void Awake()//робота при запуску елемента
    {
        UpdateIsCurrent();//Оновлюємо стан кола
        if (PulpObject != null)//Якщо об'єкт лампочки існує
        {
            PulpAnimator = PulpObject.GetComponent<Animator>(); //отримуємо аніматор лампочки
        }
    }
    private void Update()//Робота на кожен кадр
    {
        UpdateIsCurrent();//Оновлюємо стан кола
        if (cStatus == true)//Якщо коло вімкнуте
        {
            doWork(); //Запускаємо роботу
        }
        else //Якщо коло розімкнуте
        {
            resetWork(); //Припиняєм роботу
        }
    }
}

```

Код (А.5 частина 2)

```

private void doWork() //Функція роботи
{
    UpdatePulp(); //Оновлюємо лампочку
}
private void resetWork() //Функція призупинення роботи
{
    UpdatePulp(); //Оновлюємо лампочку
}
private void UpdateIsCurrent() //Функція оновлення кола
{
    cController = CircleObj.GetComponent<CircleController>(); //отримуємо компонент контролера симуляції
    cStatus = cController.isCurrent; //отримуємо з нього статус кола
}
private void UpdatePulp() //Функція оновлення лампочки
{
    if (cController.napruha > 0) //Зміна стану лампочки в залежності від статусу кола
    {
        if (AnimStage == 0 & cStatus == false)
        {
            PulpAnimator.SetTrigger("DefaultToOff");
            AnimStage = 1;
        }
        if (AnimStage == 0 & cStatus == true)
        {
            PulpAnimator.SetTrigger("DefaultToOn");
            AnimStage = 2;
        }
        if (AnimStage == 1 & cStatus == true)
        {
            PulpAnimator.SetTrigger("OffToOn");
            AnimStage = 2;
        }
        if (AnimStage == 2 & cStatus == false)
        {
            PulpAnimator.SetTrigger("OnToOff");
            AnimStage = 1;
        }
    }
    if (cController.napruha <= 0)
    {
        if (AnimStage == 2 & cStatus == true)
        {
            PulpAnimator.SetTrigger("OnToOff");
            AnimStage = 1;
        }
    }
}
}

```

Додаток Б

**КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНЬСЬКОГО
ДЕРЖАВНОГО УНІВЕРСИТЕТУ**

Я, Соболь Микола Вікторович,
учасник(ця) освітнього процесу Херсонського державного університету, **УСВІДОМЛЮЮ**, що академічна добросесність – це фундаментальна етична цінність усієї академічної спільноти світу.

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

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

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

15.10.21
(дата)


(підпис)

Микола Соболь
(ім'я, прізвище)