

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

Факультет комп'ютерних наук, фізики та математики

(повна назва факультету)

Кафедра комп'ютерних наук та програмної інженерії

(повна назва кафедри)

**3D ОБ'ЄКТИ ВІРТУАЛЬНИХ ЛАБОРАТОРНИХ РОБІТ ДЛЯ МОБІЛЬНИХ
ПРИСТРОЇВ**

(назва теми великими літерами)

**Кваліфікаційна робота (проект)
на здобуття ступеня вищої освіти “бакалавр”**

Виконав(ла): здобувач (ка)	<u>4 курсу 441 групи</u> (курс, група)
Спеціальність	<u>121 Інженерія програмного забезпечення</u> (шифр, назва)
Освітньо-професійна програма	<u>Інженерія програмного забезпе- чення першого (бакалаврського) рівня вищої освіти</u> (назва)
Керівник	<u>Заремба Святослав Петрович</u> (П.І.Б.) <u>Кандидат фізико-математичних наук, доцент Кравцов Геннадій Михайлович</u> (наук. ступінь, вчене звання, П.І.Б.)
Рецензент	<u>Кандидат педагогічних наук, доцент Зайцева Тетяна Василівна</u>

ЗМІСТ

ЗМІСТ	2
ВСТУП	2
РОЗДІЛ 1 ВІРТУАЛЬНІ ЛАБОРАТОРІЇ ТА ТЕХНОЛОГІЇ РОЗРОБЛЕННЯ ВІРТУАЛЬНИХ ЛАБОРАТОРНИХ РОБІТ	5
1.1 Огляд віртуальних лабораторій (VirtuLab, All-Fizika, Phet interactive simulations).....	5
1.2 Стандарти віртуальних лабораторій (IMS, SCORM).....	9
1.3 Програмні технології та середовища розроблення віртуальних лабораторних робіт (Unity, Unreal Engine, CryEngine, Autodesk 3ds MAX, Blender, Visual Studio)	13
РОЗДІЛ 2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО МОДУЛЮ ДЛЯ СТВОРЕННЯ 3D-ОБ'ЄКТІВ	40
2.1 Модель 3D-об'єкту віртуальної лабораторної роботи	40
2.2 Проектування програмного забезпечення 3D-об'єкту віртуальної лабораторної роботи.....	42
2.3 Проектування віртуальної лабораторної роботи. Діаграма класів	44
РОЗДІЛ 3 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	46
3.1 3D-об'єкти у віртуальних лабораторних роботах.....	46
3.2 Розробка 3D-об'єкту віртуальних лабораторних робіт з курсу фізики.....	49
3.3 Інтерфейс програмного забезпечення віртуальної лабораторної роботи.....	51
ВИСНОВКИ	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54

ВСТУП

Актуальність: Через великі темпи розвитку інформаційної галузі, величезне розуміння грає наявність програмного продукту який дозволить користуватись лабораторними роботами онлайн.

Об'єкт дослідження: електронно-інформаційні ресурси з використанням мобільних додатків за технологією доповненої реальності.

Предмет дослідження: віртуальні лабораторні роботи з 3D об'єктами для мобільних додатків за технологією доповненої реальності.

Мета дослідження: створити моделі а також 3D об'єкти для віртуальних лабораторних робіт з використанням мобільних додатку.

Наукова новизна одержаних результатів: за час виконання даної кваліфікаційної роботи було досліджено існуючі системи для дистанційного навчання а також зроблено висновок, що на даний момент в мобільних пристроях відсутні додатки якіб допомогли проходити лабораторні роботи віртуально.

Тому, було прийнято рішення розробити Asset для Unity який допоможе полегшити розробку для мобільних телефонів.

Практичне значення одержаних результатів: результатом даної роботи є набір методів та 3D об'єктів для взаємодії на основі котрих можна створити віртуальну лабораторію з фізики.

Завдання дослідження:

1. Дослідити методи створення і використання 3D-об'єктів на мобільних пристроях.
2. Побудувати об'єкти для лабораторної роботи.
3. Створити модель для лабораторної роботи.
4. Розробити програмний додаток на мобільний пристрій для роботи с 3D-об'єктами в лабораторній роботі.

Використанні технології та програмні засоби:

1. Unity.
2. Autodesk 3ds Max.
3. Blender.
4. Microsoft Visual Studio.

РОЗДІЛ 1 ВІРТУАЛЬНІ ЛАБОРАТОРІЇ ТА ТЕХНОЛОГІЇ РОЗРОБЛЕННЯ ВІРТУАЛЬНИХ ЛАБОРАТОРНИХ РОБІТ

1.1 Огляд віртуальних лабораторій (VirtuLab, All-Fizika, Phet interactive simulations)

В сучасному світі спостерігається тенденція до використання інтерактивних, мультимедійних освітніх ресурсів що сприяють більш ефективним умовам навчання. Саме системи віртуальної реальності якраз і забезпечують активні методи навчання в освітньому процесі.

Моделюючи поведінку об'єктів реального світу в комп'ютерному освітньому середовищі, віртуальні лабораторії являються електронними ресурсами, які допомагають учням і студентам краще засвоїти знання з науково-природничих дисциплін.

Мета віртуальної лабораторії створити умови для забезпечення учнів розумінням і прийняттям сутності досліджуваного об'єкту.

Саме віртуальні лабораторії формують пізнавальну і творчу активність учнів, дозволяючи моделювати об'єкти і процеси навколишнього світу, забезпечуючи доступ до реального лабораторного обладнання.

Аналізуючи різноманітні віртуальні лабораторії було досягнуто розуміння, що Android є найкращою платформою для їх створення, а лабораторні роботи повинні бути в тривимірному просторі.

VirtuLab

Один з найбільших сервісів по проходженню віртуальних лабораторій, в якому зосереджені лабораторії з фізики, хімії, екології, біології, одна фізика налічує 6 розділів:

1. механічні явища;
2. теплові явища;
3. квантові явища;
4. електроенергія;
5. молекулярна фізика;
6. оптика.

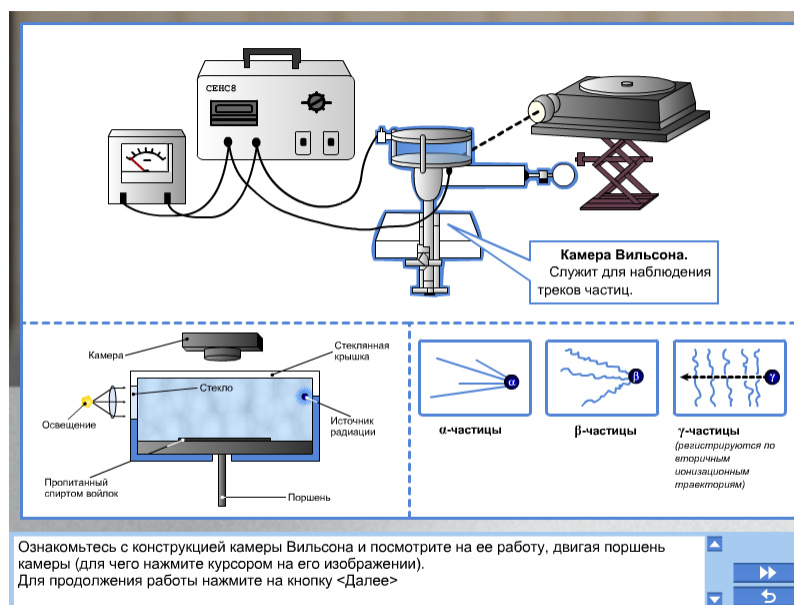
Переваги:

1. Лабораторії представлені в дво- і тривимірних просторах.

Недоліки:

1. У більшості лабораторних робіт відсутня можливість що-небудь змінити більшість робіт – анімації.

Изучение взаимодействия частиц и ядерных реакций



>>Опыт Резерфорда

Рис. 1.1 Досвід Резерфорда VirtuLab

All-Fizika

Предмет фізики представлений величезним сервісом де є фізичний довідник, який вміщую всі одиниці міри, великий збірник формул з фізики. Є всі необхідні дані з фізики для занять учнів.

Переваги:

1. Кожну роботу можна налаштувати як ти забажаєш і провести заміри за допомогою того ж таки приладу, що тут є.

Недоліки:

1. Лабораторні роботи мають недолік, бо викладені у формі невеликих дослідів, а відкриваються за допомогою нестабільного плагіну flash player.
2. Лабораторії представлені в двовимірних просторах.

Прибор Атвуда. Проверка второго закона Ньютона

Описание работы лабораторной работы можно найти во вкладках флэш-ролика.

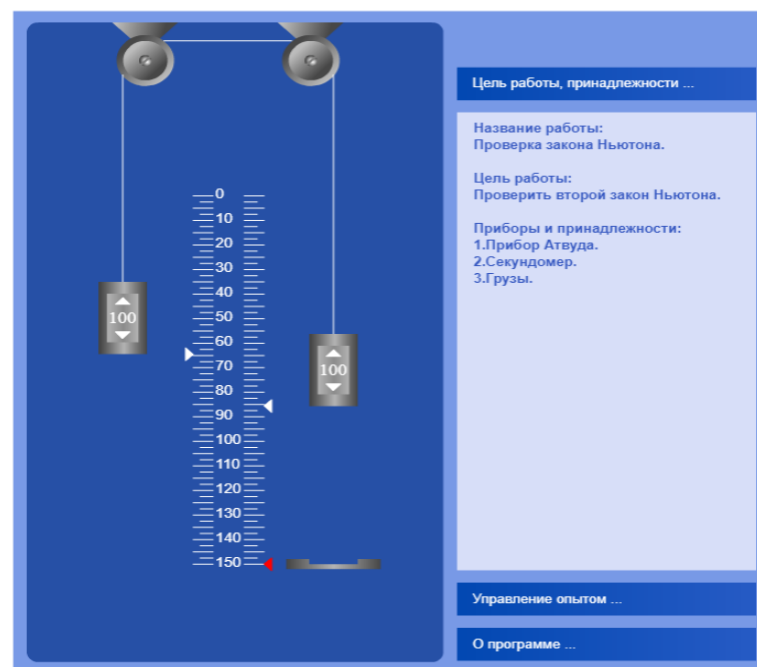


Рис. 1.2 Перевірка другого закону Ньютона All-Fizika

Phet interactive simulations

У 2002 році в університеті Колорадо Боулдер, Карло Віман, заснував проект PhET[15] Interactive Simulations. Це дало змогу безкоштовно показувати учням і студентам наукові досліди в інтерактивній ігровій формі, що дозволило краще засвоювати матеріали з природничих і точних наук.

Переваги:

1. Усі віртуальні симуляції представлені в інтерактивній ігровій формі, що стимулює процес навчання.

Недоліки:

1. Всі симуляції виконуються на англійській мові.

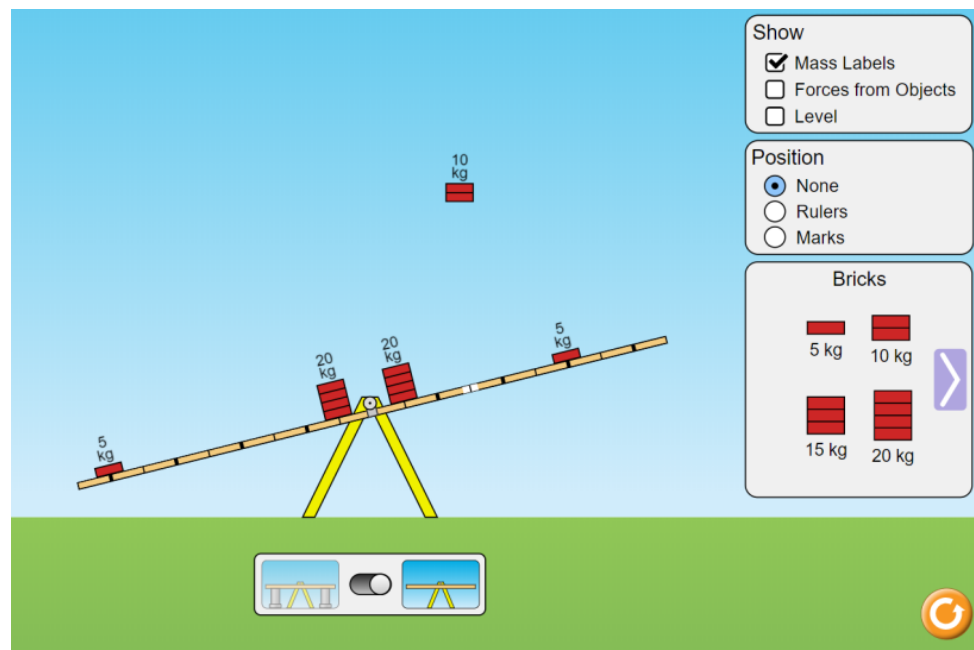


Рис. 1.3 Лабораторна робота з Phet Interactive Simulations

1.2 Стандарти віртуальних лабораторій (IMS, SCORM)

В наш час, коли електронне навчання впроваджується в життя в широких масштабах, покладено початок ADL Ініціативи.

Стандарт - це формат, затверджений визнаним інститутом стандартизації або прийнятий підприємствами галузі де-факто як зразок. Згідно стандартам користувачі можуть комбінувати устаткування і програми різних виробників. При відсутності єдиного стандарту обираються пристрої і програми одного виробника. Існує безліч стандартів:

1. для мов програмування;
2. форматів представлення даних;
3. операційних систем;
4. протоколів зв'язку і т.д

Стандартизації підлягають не тільки устаткування, але й і програмне забезпечення.

Найпопулярнішим серед стандартів є SCORM та IMS, які розроблені для дистанційного навчання. Саме ці стандарти містять всі вимоги щодо організації навчального матеріалу і всієї системи дистанційного навчання.

SCORM

Він заснований на стандарті XML, і є піонером на шляху розвитку концепції ADL (ініціативна група Advanced Distributed Learning), оскільки визначає структуру навчальних матеріалів і інтерфейс середовища виконання. На основі створених іншими організаціями специфікації і стандартів електронної освіти SCORM, описує цю технічну структуру за основними принципами, специфікаціями і стандартами, електронної освіти.

ADL створив SCORM для інтеграції різних стандартів і специфікацій (наприклад, LOM, IMS CP) в єдину модель контенту.

SCORM – це технічна інфраструктура що дає можливість спільно використовувати об'єкти в розподіленому навчальному середовищі. Зразкова модель об'єкта контенту для спільного використання (Sharable Content Object Reference Model, SCORM) визначає модель агрегації контенту і робоче оточення учбових об'єктів у рамках вебнавчання. Згодом цей проект було вдосконалено. У SCORM використовуються результати розробок цілого ряду проектів і організацій: Inc.3, Aviation Industry CBT, Alliance of Remote Instructional Authoring & Distribution, IMS Global Learning Consortium.

Для опису навчального контенту на основі специфікації створених IMS Global Learning Consortium і IEEE LTSC, версія SCORM вводить концепцію упаковки контенту з оновленими метаданими. Через широке застосування SCORM, при створенні систем навчання, які спираються на інтернет ця версія отримала популярність.

Модель SCORM складається:

1. вступ, або оглядова частина;
2. описи моделі інтеграції змісту;
3. описи робочого середовища.

У вступі, або оглядовій частині іде опис стандартів ADL, і обґрунтовується створення еталонної моделі. У другій частині викладено практичні поради, як виявити ресурси і перетворити їх у структурований навчальний матеріал. В третій частині – поради що до зв'язку з веб – середовищем і відстеженням його змісту.

На даний час SCORM є ідеальною моделлю для перевірки ефективності і практичності застосування набору окремих специфікацій і стандартів, він використовується розробниками стандартів IEEE і IMS для об'єднання створених специфікацій. Ідеальність SCORM полягає в тому, що усі елементи навчальних програм функціонально сумісні з усіма LMS системами.

IMS

Стандарти розроблені консорціум глобального навчання IMS (IMS Global Learning Consortium) сприяють впровадженню технології навчання заснованої на функціональній сумісності, тому вони отримали всесвітнє визнання і стали стандартами для навчальних продуктів і послуг. На відміну від чинних систем організації навчання, у яких керувальні функції здійснюються по різному і викликають велику собівартість навчальних матеріалів навчальних матеріалів, специфікації IMS, допомагають уникнути цих проблем.

Основними напрямками розробки специфікацій IMS є :

1. метадані,
2. упаковка змісту;
3. управління змістом;
4. сумісність питань і тестів.

Стандартними для метаданих слугує мінімальна кількість атрибутів, визначення місцезнаходження і оцінка навчальних об'єктів.

Атрибути навчальних об'єктів:

1. тип;

2. ім'я власника;
3. ім'я автора;
4. формат об'єкта;
5. терміни поширення.

За необхідності, стандарти можуть включати опис взаємодії студента і викладача, стиль викладання, рівень отримання знань і підготовка студента.

Модель упаковки змісту IMS, створена для визначення стандартного набору структур для обміну навчальними матеріалами в умовах Інтернету. Мета специфікації сумісності питань і систем тестування IMS – розширити можливості користувачів, експортувати і імпортувати матеріали, а також, зробити сумісним зміст навчальних програм із системами оцінки.

1.3 Програмні технології та середовища розроблення віртуальних лабораторних робіт (Unity, Unreal Engine, CryEngine, Autodesk 3ds MAX, Blender, Visual Studio)

Unity

Історія



Рис. 1.4 Логотип Unity

У 2006 році ігровий двигун Unity, який був запущений роком раніше і дав поштовх розвитку ігор, переміг у категорії «Найкраще використання графіки Mac OS X» в Apple Design Awards Apple Inc, згодом додав підтримку для Microsoft Windows та веб-браузерів.

Незабаром у 2007 році було запущено Unity 2.0. Він мав 50 нових можливостей, включаючи оптимізований двигун місцевості для детальних 3D середовищ, динамічних тіней у режимі реального часу, спрямованих променів та вогнів тощо. Для створення багатовикористовуваних ігор основі протоколу User Datagram, для розробників був включений мережевий рівень.

У 2008 році Unity додав підтримку iPhone для компанії Apple, і заслужував на повагу зі сторони розробників ігор для iOS. Процес розвитку триває і 2010 року було запущено Unity 3.0, який забезпечений функціями, що дають більші графічні можливості настільним комп'ютерам та консолям.

Також Unity 3.0 передбачає інтеграцію інструменту Beast Lightmap Illuminate Labs 'Beast Lightmap, відкладене візуалізацією, вітчизняне відображення шрифтів, вбудований редактор дерева, автоматичне ультрафіолетове відображення, аудіофільтри. Unity 3.0 підтримує Android.

У 2012 Unity Technologies поставила Unity 4.0, яка додала DirectX 11, і підтримку Adobe Flash, нові інструменти для анімації Mecanim та доступ до попереднього перегляду Linux.

За допомогою ігрового двигуна Unity Facebook інтегрував комплект розробки програмного забезпечення для ігор а через 3 роки компанія розробила нову ігрову платформу для ПК.

З метою зробити ігри загально доступними, ігровий двигун продовжують удосконалювати. З'являється Unity 5.0, що пропонує глобальне освітлення в режимі реального часу, попередній перегляд світлового відображення Cloud Cloud, нову аудіосистему та фізичний двигун Nvidia PhysX 3.3.

У 2015 році продовжується вдосконалення версії, в червні в Unity 5.1 було виправлено ряд помилок і зроблено підтримку VR (віртуальної реальності). Згодом в цьому ж році оновлено Unity 5.2 задані нові можливості для розробників додатків під андроїд, для роботи із звуком додано підтримку Spatialization API для модулів розширення аудіо. У версії Unity 5.3 оновлено MonoDevelop, одночасне редагування декількох сцен, інструмент для 2D, автоматичне тестування. Оптимізація графіки, включно з новим ядром OpenGL, експериментальна підтримка Metal в OS X і оновлена система частинок. Покращена підтримка WebGL и iOS 9, підтримка інтегрова-

них внутрішньоігрових придбань. Додана експериментальна підтримка DirectX 12.

Версії Unity 5.4 покращили графічний рендер, допрацьовано VR платформу, додано нові дзеркальні ефекти, покращено імпорт графіки. Найбільшим нововведенням в Unity 5.5 2016 році стала підтримка Microsoft HoloLens.

Cinematic Image Effects представило Unity 5.6, додавши ефекти освітлення та підтримку Nintendo Switch.

Представлений Unity 2017 демонструє двигун відображення графіки в реальному часі, класифікацію кольорів та світобудування, аналітику операцій і звітність про продуктивність, тобто плани Unity Technologies вийшли за межі відеоігор. Завдяки удосконаленню камер Cinemachine виникла можливість слідувати за об'єктами і перемикатись за тримерами, а також змінювати об'єкти зображень. Впроваджено інструмент Post - Processing Stack, який об'єднує в собі всі візуальні ефекти.

У травні 2017 представлено Unity 2018, продовжується робота розвитку версії компілятор на основі LLVM (Low Level Virtual Machine), різні конвеєри для різного рівня графіки. Для підвищення продуктивності гри впроваджено систему ігрового коду C# Job System для підвищення продуктивності гри.

У 2019 році вдосконалено конвеєри рендеренгу. Для курування заставками через тригери впроваджено інструмент Timeline Signals, створено можливість оновлювати ігри. В цьому ж році перебудовано ядро Unity на базі DOTS (Data-Oriented Technology Stack) з використанням багатопотоковості процесора.

Unity 2020 ще більш вдосконалюється: детальнішими стають інформація про набори ресурсів і їх помилки, налаштування об'єктів у сцені. Вводяться нові інструменти налагодження коду, прискорюється обробка

спрайтів, система візуального створення скрипів Bolt, а також оптимізовано роботу з VR, і додано підтримку HDR-дисплеїв. Для оцінки споживання ресурсів грою, запущено інструмент Profiler.

Unreal Engine

Історія

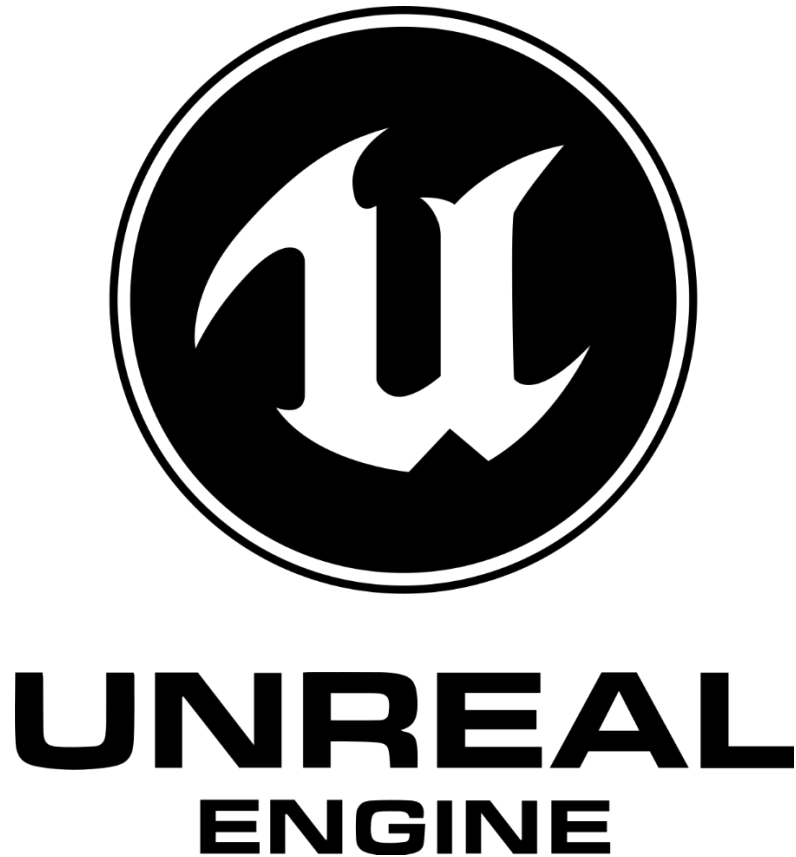


Рис. 1.5 Логотип Unreal Engine

В 1998 році Epic Games розробив ігровий двигун Unreal Engine, на основі якого розроблено безліч проектів. Саме цей двигун став піонером в розвитку відеоігор, а створив його програміст, засновник компанії Epic Games, Тім Суїні. Проекти Тіма Суїні використовувалися в багатьох культових проектах і були конкурентно спроможними, вони дали поштовх розвитку відеоігор.

Перша версія двигуна Unreal Engine 1.0 вийшла 24 роки тому і зразу ж випередила двигун id Tech 3, бо включала: графічний двигун, фізичний двигун, управління файловою і мережевою системами, штучний інтелект, готове середовище розробки ігор UnrealEd. Мережевий складник двигуна був кроссплатформенним і гравці Windows могли грати на одних серверах з гравцями на macOS. Сам же двигун працював на macOS, Windows, PlayStation 2 і Xbox.

Особливість Unreal Engine полягала в тому, що двигун був векторним, тобто працював з графами сцени, який є структурою що містить просторове уявлення графічної сцени і допомагає знайти логічний зв'язок між об'єктами. Цей логічний ланцюжок взаємопов'язаних об'єктів спрощує написання гри і взаємодію об'єктів гри.

Unreal Engine мав і інші особливості: Skybox, технологія warp.

Skybox – це проектування на поверхню і малювання з іншої точки, він відображав зворотню грань полігонів із накладеною текстурою на об'єкт. Завдяки прийому масштабування текстур відображалось реалістичне навколишнє середовище. Використання мультитекстурування дозволяло домогтися ефекту тривимірності поверхонь, з'явився ефект гало навколо джерел світла.

Технологія варпінгу (warp) – це можливість при промальовуванні замінювати зображення однієї поверхні проекцією зображення іншу, паралельну поверхню.

Недолік: ця технологія виглядала у грі як портали.

Через рік двигун було оновлено, Unreal Engine 1.5 мав лицьову анімацію для персонажів, роздільна здатність збільшилася 1024x1024, ведено систему частинок, що розширюється, технологію S3TC. Інтегровано другу версію редактора Unreal Ed.

В наш час перша версія двигуна, недоступна для ліцензування, але вихідний код для збирання файлів для персонального використання поширюється за версією Unreal Retail. На Unreal Engine вийшло більше двадцяти ігор, але найкращою стала Unreal Tournament.

Unreal Engine 2 у 2002 році Unreal Engine зазнав перетворень, повністю переписане ядро та механізм рендерингу, інтегровано третю версію редактора Unreal Ed. Фізичний двигун створено іншим видавцем Karma. Двигун мав ряд особливостей: рідку поверхню, яка складалась з великої кількості полігонів що імітує поверхню, підтримку VoIP, розпізнавання мовлення, нерухомі транспортні засоби, та транспортні засоби що кріпляться до інших. EAX 3.0 – двигун тривимірного звуку, перехід на двигун Karma і 32 – бітну графіку.

Версія Unreal Engine 2.5 направлена на оптимізацію двигуна завдяки підтримці DirectX 9, OpenGL 2 і Pixomatic. Додано підтримку 64 – бітних операційних систем Windows і Linux, підтримку юнікоду (16-біт), дозвіл текстур розширено 4096x4096 пікселів.

Unreal Engine 2X и Runtime це спеціальні версії двигуна - в першому випадку для Xbox: були введені нові візуальні ефекти, такі як depth of field, динамічна гамма-корекція, bloom та різні варіації blur.

Unreal Engine 3 розрахована на багатоядерні ПК, PlayStation 3 і Xbox 360. Двигун використовує два потоки: основний потік і потік рендеренгу. З'явилася підтримка Wii, iOS та Windows RT. Новий двигун забезпечений підтримкою HDR, попиксельним освітленням, динамічними тінями, Pixel Shader 4.0, геометричними шейдерами.

Фізичний змінено на PhysX, інтегрований анімаційний двигун FaceFX поліпшив реалістичність міміки.

Unreal Engine 3.5 пройшла процес оптимізації: двигун почав працювати з DirectX 11, додано пост обробки Ambient occlusion, для покращення

освітлення. Покращена фізика м'яких тіл і оточення, що руйнується. На движку вийшло більш ніж сто проектів. Ціна повної версії движка підємна тільки великим ігровим студіям.

Unreal Engine 4 – стала нановішою версією і постійно розвивається, вона підтримує консолі поточного покоління, (PlayStation 4 і Xbox One), API DirectX 12 і Nintendo Switch.

Додано підтримку очок віртуальної реальності, SteamVR, HTC Vive та PlayStation 4 Morpheus. За допомогою нової системи рендеренгу знижено навантаження на відеокарту. Новий фотореалістична глибина різкості (Depth of Field), підбиття напівпрозорих поверхонь, можливість перемотування останніх декількох хвилин геймплею у мережних іграх. Для створення спрайтів – вбудований двигун Paper2D. Unreal Engine 4 досить популярний серед великих і малих студій, в розробці знаходиться дуже багато проектів.

CryEngine

Історія



CRYENGINE®

Рис. 1.6 Логотип CryEngine

У 2000 році компанія Crytek розробила один з найважчих двигунів, ігри на якому по правду вважаються найкрасивішими. Двигун CryEngine був представлений на виставці ECTS 2000 для демонстрації можливостей відеокарти Nvidia і отримав позитивні відгуки. Уже 2002 році Crytek закінчив роботу над двигуном.

Розробники додали нові фішки і допрацювали існуючі: рендер, який працював з DirectX 9 і безшовно малював відкриті і закриті локації, фізичну систему, яка підтримує інверсну кінематику персонажів, транспортні засоби, тіла, рідину і rag doll, ефекти м'якого тіла. Штучний інтелект працював командно, а звукове оточення реагувало на дію гравця і персонажів гри. Щоб зменшити навантаження на відеокарту додано Geometry Instancing, що дало змогу економити ресурси. У двигуні підтримувалося реальне попиксельне освітлення, заломлення, об'ємні ефекти жару, вікна, прозорі комп'ютерні дисплеї, вибоїсті відображення. Незважаючи на ряд переваг, права на двигун було продані Ubisoft а на CryEngine вийшла тільки одна гра Aion.

CryEngine 2, який вийшов у 2007 році був орієнтований на ПК, бо консолі того часу були для нього заслабкими. Рендер працював на DirectX 10, що давало якісні малюнки об'єктів, завдяки x64 системам продуктивність була вищою ніж на x86.

Завдяки parallax occlusion mapping поверхневі текстури мають багато деталей, нанесених без полігонів, джерела освітлення виглядають природно і створюють м'які тіні. Дякуючи технологіям управління деталізації LOD близькі до камери об'єкти та ландшафт рендеруються у максимальній якості, а далекі навпаки. Завдяки цій технології отримали змогу відображати сцени до 8 км. Двигун використовує наперед визначені характеристики розломів та руйнувань. Значно покращено анімацію людей моделей, транспо-

рту. Інтегрований редактор персонажів надає попередній перегляд анімацій усередині редактора Sandbox 2.

CryEngine 3 зазнав незначних змін: шейдери пишуться одноразово мовою програмування високого рівня, а потім автоматично компілюються під кожну платформу, введення шейдерів оптимізується через налаштування художника і під тривимірне середовище, що дало змогу створити «ефект невидимості». Підтримка до чотирьох джерел світла, кубічні карти, відбиття, туман, дзеркальні структури тощо – забезпечила шейдерна програма «Übershader».

Нова версія CryEngine 4, яка вийшла в 2014 році відрізнялась від попередньої лише підтримкою консолей Xbox One і PlayStation 4.

У 2016 році з'являється CryEngine V. Версія движка була абсолютно доступною через її безкоштовність. CryEngine V мав підтримку DirectX 12, новий API дозволяє рендерувати найкрасивішу місцевість з найменшим навантаженням на процесор і відеокарту. Створено нову волюметричну систему, адаптовано для роботи з VR-проектами, що оптимізувало 3D-рендеринг і збільшило продуктивність, но система частинок допомогла сфокусувати все навантаження на рендерзі в режимі реального часу на CPU.

Autodesk 3ds MAX

Історія



Рис. 1.7 Логотип Autodesk 3ds MAX

У 1988 році Gary Yost розробив програму забезпечення для роботи з 3D моделюванням - 3D Studio і встановив відносини з Autodesk. Tom Hudson розпочав роботу на програмним забезпеченням додатку для моделювання і рендеренгу, в якому було чотири модулі: Shaper, Loft, Editor і Material Editor. З приходом другого розробника Dan Silva в компанію було представлено розроблений ним п'ятий модуль кейфреймінг. 3D Studio перетворилась на програму для 3D анімації. Autodesk разом з групою Yost видали реліз 3D Studio DOS у 1990 році.

3D Studio R2 у 1992 році вийшов 3D Studio R2, в якому було введено інтерфейс IPAS для плагінів, обробників зображень так процедурне моделювання, зміни GUI. Залишався недолік: незручне перемикання між модулями.

3D Studio R3 вийшов у 1993 році і мав ряд новинок: два нових види розширень для обробки бітмапів та кейфремінгу.

3D Studio R4 був останнім релізом для платформи MS-DOS і мав ряд оновлень: inverse kinematics, shaded preview, keyscript, перший реліз із початковою підтримкою DXF.

Команда розробників Jaguar поповнилася новим членом Rolf Berteig, розробником модуля ІК для 3D Studio і налічувало четверо осіб. Модуль персонажної анімації Cheetah мав складну концепцію і започаткував зміни в архітектурі і дизайні Jaguar.

3D Studio MAX у 1995 році Autodesk анонсувала 3D Studio MAX для платформи Windows NT на SIGGRAPH у Лос-Анджелесі. Після цього показу 3D Studio MAX було представлено на Autodesk Cad Camp, що і поклало початок бета – тестуванню. У 1996 році вийшов реліз для платформи Windows NT. У 1996 році вийшов реліз для платформи Windows NT.

Studio MAX 1.0 вимагав Windows NT 3.5 і підтримував перші карти 3D прискорення 3D Labs GLiNT з драйверами Heidi. Згодом додали SDK а також був апдейт для підтримки WinNT 4.

З'явилась концепція Undo у 3D додатку. Розроблено Modifier Stack, Material Editor і Animation Controller System були покращені. Додалися сторонні рендери.

32 - бітовий додаток, мультипоточковий з об'єктами – орієнтованою архітектурою з системою посилань не змінився і досі. Це був додаток з повноцінною модульною архітектурою, завдяки інтерактивній графіці новий інтерфейс 3D Studio MAX HEIDI піднявся на другий рівень.

Cheetah продемонстрував можливість анімувати все що завгодно.

3D Studio MAX R2 на Siggraph 97 у 1997 році було презентовано Studio MAX R2 в якому було додано Ray-tracing у Scanline рендерер за допомогою Raytrace матеріалу та карт розробленими у Blur Studio Steven Blackmon та Scott Kirvan. У Digimation інтегровано Lens Effects, Post Effects і інструмент моделювання NURBS. Вбудована мова програмування MAXScript. Підтримка OpenGL. Оновлено NURBS та підтримку імпорту VRML.

3D Studio MAX R3 вийшов у червні 1999 року, який став останній реліз під логотипом Kinetix. У ньому було перероблено інтерфейс з метою забезпечення підтримки іконок і панелей інструментів, додано підтримку Anti-aliasing і Super-sampling. Формат RPF використовувався в композитингу Paint/Composite, After Effects і Digital Fusion.

У травні 2000 року був готовий конектор для 3D Studio MAX R3 до mental ray stand-alone.

Gmax – безкоштовна програма, виготовлена на версії 3D Studio MAX 3.1. За задумом Autodesk, її призначенням була розробка моделей для ігор, однак вона підходила і для початкового освоєння 3D графіки.

Discreet 3ds max 4 включає в себе новинки: контекстні меню - QuadMenus і систему налаштування інтерфейсу користувача ActionItems, режим прев'ю-рендера ActiveShade, новий набір інструментів для моделінгу.

Було випущено два проміжні оновлення для користувачів і один – для студентської версії.

У 2002 році Discreet випустила самостійно створений 3ds max 5. У версії з'явилася підтримка плагінів попередньої версії, але головним нововведенням став стороній рендер mental ray, і просунута підсистема освітлення у рендерері Scanline яка складалася з брутфорсовий модуль глобального освітлення і модуля російського освітлення. Серед нововведень: інтеграція модуля Reactor, функція Render To Texture, нові Transform gizmos, підтримка шарів. 3ds max 5 отримав нагороди: «Вибір редакції від NextGenElectronics»,

«Atomic Hot від журналу Atomic Maximum Power Computing».

Новий 3ds max 6 вимагав перекомпіляції всіх плагінів. Основні перевтілення: Particle Flow, новий модуль Vertex Paint та інші оновлення інструментів.

У серпні 2004 року вийшов еволюційний апдейт, заснований на ядрі шостої версії Discreet 3ds max 7.

Основні функції: інструменти в Editable Poly tools – Bridge, Deform, Relax painting, Soft Selections Painting, опція Preserve UVs. Підтримка генерації та рен-дерингу карт нормалей. mental ray 3.3, що включає шейдери підповерхневого розсіювання, Ambient Occlusion і підтримку Render To Texture.

Autodesk 3ds Max 8 вийшов через сімнадцять років 2005 році. Він мав сумісність SDK яка забезпечувала три версії поспіль, тобто плагіни ві шостої та сьомої версії могли використовуватися без рекомпеляції у восьмій.

Підрозділ discreet через наближеність до Autodesk переіменований на Autodesk Media and Entertainment Division, а літера «М» у слові «Max» стала знову великою.

Основні функції: Asset Tracking з підтримкою сторонніх рішень та Autodesk Vault у стандартному постачанні, Hair and Fur. Відбулась велика робота по оновленню та додаванню нових функцій у наявні інструменти.

Autodesk 3ds Max 9 – перший реліз, який вийшов у 32 - та 64 - х біт-ному форматі. Були перекомпільовані плагіни новому компіляторі Visual Studio. Головними новинками стали: підтримка .NET в MAXScript, підтримка симуляцій HAVOK 3, покращена підтримка mental ray 3.5. ProBoolean та ProCutter.

31 березня 2008 року Autodesk 3ds Max 2009 і реліз 3ds Max 2009 Design. Відмінності їх полягали в тому що 3ds Max мав SDK, а 3ds Max 2009 Design мав Lighting Analysis.

Обидва релізи пройшли однакові досконалення: додано навігацію в ViewCube та Steering Wheel, перероблено фотометричні джерела світла, Virtual Frame Buffer, Autodesk ProMaterials.

Найвизначнішою подією з часів виходу 3ds max 4 був вихід релізу 3ds Max 2010, адже в ньому закладено багато нововведень: до програми включено набір інструментів для моделювання PolyBoost, завдяки яким з'явилося більше ста нових інструментів. У релізі представлено до трьохсот нововведень, а також Graphite Modeling Tools. Версія для архітекторів і дизайнерів 3ds Max Design включала всі можливості 3ds Max 2010 за винятком SDK, також до її складу ввійшла технологія Exposure, імітація сонячного світла та штучних джерел освітлення.

Обидві версії мали нові інструменти: прев'ю Ambient Occlusion та Exposure Controls у вікнах проекції, Containers – для роботи кількома художниками над одним проектом, підтримку Soft-shadows. Додано параметри до mental ray Global Tuning. Покращено Particle Flow заснований на Orbaz Particle Flow Tools Box #1, на базі технологій Scene Explorer впроваджено Material Explorer.

Журнал Game Developers відзначив 3ds Max Award як найкращий Art Tool.

Презентований Autodesk 3ds Max 2011 повністю підтримував ОС Windows 7 і мав редактор матеріалів Slate Material Editor, поліпшене моделювання, локальне редагування контейнерів, вбудований набір інструментів анімацій. Функції 3ds Max дозволяють здійснювати корекцію кольору, проектування камери, векторне малювання, кодування. Надана можливість збереження у форматі попередньої версії, механізм апаратного рендерингу Quicksilver. Імпортування файлів Google SketchUp.

Найкращим релізом з усіх став Autodesk 3ds Max 2012, який вийшов у 2011 році. Його особливістю переписування ядра програми, зокрема, вікон проекції, що підвищувало продуктивність праці, революційне оновленням стали вікна проекції, вони стали швидшими за попередні, а також нова система симуляції динамічних об'єктів mass FX.

Autodesk 3ds Max 2013 вийшов у 2012 році у складі Autodesk Suites.

Переваги: новий інструмент Tabbed View для роботи з наборами вікон проєкцій, додано список менеджера налаштування робочої області Workspace. State Sets – система запису параметрів та налаштування сцени в окреме сховище. Додано інструменти mass FX.

Через рік вийшов Autodesk 3ds Max 2014, як і попередня версія у складі Autodesk Suites. Робота направлена на переписування ядра програми у рамках проєкту Excalibur. Переваги: оновлено верхнє меню програми, перероблено інструмент Camera Match, доступнішими стали додаткові функції box два та три для Particle Flow.

Autodesk 3ds Max 2015 вийшов у 2014 році. У релізі було припинено підтримку 32-бітної версії 3ds Max, змінено UI Particle Flow. До Scene Explorer функції шарів. Значущою подією виявилось нововведення Quad Chamfer. Було досягнуто можливість передавати персонажів в русі, оновлено систему автоматичного створення натовпу, удосконалено інтеграції 3ds Max і After Effects, додано об'єкт хмара пікселів.

Autodesk 3ds Max 2016 відрізнявся від попередніх релізів відсутністю поділу на 3ds Max та 3ds Max Design.

Головною перевагою стали інструменти Open SubDivision, нова технологія для роботи з високополігональними об'єктами OpenSubdiv, розроблено Pixar, що дозволило працювати з об'єктами, використовуючи ресурси CPU та GPU паралельно. «Схематичний інтерфейс скриптингу» дозволив створювати об'єкти та модифікувати їх.

Autodesk 3ds Max 2017 був найгіршим за всю історію релізів, допущено ряд недоліків, викликавший незадоволення користувачів: не інтегрований Mental ray, підтримка HiDPI. Єдине покращення було Boolean.

Серед нововведень наступного релізу Autodesk 3ds Max 2018 стало: можливість попереднього перегляду шляху анімованих об'єктів у вікнах

проекцій завдяки Motion paths. Mental ray замінила інтеграція Arnold. Витяг інформації безпосередньо з моделі і використання для передачі про інші типи модифікацій стало можливим завдяки Data Channel Modifier, а можливість змішувати стики різних текстурних карт - Blended Box Map. З червня 2017 року у 3ds Max 2018 було включено 3ds Max Interactive, інструмент реального часу, заснований на Autodesk Stingray, це двигун VR, який вдосконалив 3ds Max для створення інтерактивних візуалізацій. Arnold Render оновлено до версії 5.2 та він підтримує запікання текстур. Забезпечено підтримку Open Shading Language з більш ніж 100 включених шейдерів.

Autodesk 3ds Max 2019 був спрямований на оновлення 2018.1, 2018.2, 2018.3, 2018.4.

Autodesk 3ds Max 2020 продовжує роботу по оновленню 2019.1, 2019.2, 2019.3 та 2020.

Було покращено продуктивність Animation Preview, відновлено модифікатор Chamfer і продовжено роботу над OSL shaders.

Наступною моделю став Autodesk 3ds Max 2021 який вийшов у березні 2020 року, новим атрибутом стала надана можливість розробникам плагінів не перекомпілювати плагіни, а також відключити установку Autodesk Material Library в установнику. З'явилась можливість авторизуватися під своїм обліковим записом, встановити 3ds Max 2021 прямо з хмари. Новий інтерфейс з лінійним відображенням параметрів та підтримкою скрипів, PBR, OSL шейдерів та Mikkt Normal maps – це все робота по вдосконаленню версії. Було додано повну підтримку Physical Material у форматі FBX, а також підтримку для розрахунку weighted normals та застосування до моделей у 3ds max. Додано кілька функцій для прощеного використання PBR технологій у вікнах проекцій та рендерингу.

Blender

Історія



Рис. 1.8 Логотип Blender

У Нідерландах у 1988 році Тонм Розендалем було заснована найперспективніша в Європі NeoGeo. Ця компанія створювала продукцію для великих корпоративних клієнтів серед яких був Philips. Тон займається художнім напрямом і питаннями розробки внутрішнього програмного забезпечення. У 1995 році було створено пакет 3D моделюванні – Blender.

У 1998 році Розендалем заснована дочірня компанія Not a Number для подальшої розробки Blender. Задумкою автора Blender була розробка безкоштовного кросплатформенного пакета моделювання. У 1999 році компанія Not a Number представила Blender широкому колу користувачів на конференції SIGGRAPH, де він завоював симпатію відвідувачів.

Влітку 2000 року було випущено Blender версії 2.0, в який було додано інтегрований ігровий двигун, і кількість користувачів перевищувала 250000.

У 2001 році вийшов перший комерційний продукт Blender Publisher, який був націлений на ринок інтерактивної веб – орієнтованої 3D – медіа продукції.

Через комерційні негаразди Тон Розендаль був вимушений заснувати не комерційну організацію Фонд Blender, який мав на меті продовжити розробку Blender, як громадського проекту з відкритим вихідним кодом.

13 жовтня 2002 року Blender був випущений на умовах громадської ліцензії GNU. Очолювана Тонем Розенбелем розробка Blender триває до цих пір.

Версія Blender:

- 1.00 – січень 1994 року: Blender у розробці на анімаційній студії NeoGeo;
- 1.23 – січня 1998 року: Версія SGI, опублікована на веб, IrisGL;
- 1.30 – квітень 1998 року: Версії для Linux та FreeBSD, портування на OpenGL та X11;
- 1.3x – червень 1998 року: Засновано NaN;
- 1.4x – вересень 1998 року: Випущено Альфа-версію для Sun та Linux;
- 1.50 – листопад 1998 року: Опубліковано перший Посібник;
- 1.60 – квітень 1999 року: C-key (нові функції поза блокуванням, \$95), випущено для Windows;
- 1.6x – червень 1999 року: Випущено для BeOS та PPC;
- 1.80 – червень 2000 року: Кінець C-key, Blender знову став повністю безкоштовним (безкоштовним);
- 2.00 – серпень 2000 року: 3D-інтерактивний та реально-часовий рух;
- 2.10 – грудень 2000 року: Новий рух, фізика та Python;
- 2.20 – серпень 2001 року: Система анімації персонажів;
- 2.21 – жовтень 2001 року: Запуск Blender Publisher;
- 2.2x – грудень 2001 року: Версія для macOS;

Версія Blender «Open Source»:

- 2.25 – жовтень 2002 року: Blender Publisher стає вільно доступним, та створюється експериментальне дерево Blender'а;

- 2.26 – лютий 2003 року: Відкрито-джерельний випуск Blender'a;
- 2.27 – травень 2003 року: Відкрито другий джерельний випуск Blender'a;
- 2.28x – липень 2003 року: Перший випуск із серії 2.28x;
- 2.30 – жовтень 2003 року: Попередній випуск 2.3x UI, представлений на другій Конференції Blender;
- 2.31 – грудень 2003 року: Оновлення проекту до стабільної 2.3x UI;
- 2.32 – січень 2004 року: Великий перегляд здатностей внутрішнього зображення, рендеринга;
- 2.33 – квітень 2004 року: Ігровий рушій повертається, нові процедурні текстури;
- 2.34 – серпень 2004 року: Взаємодії частинок, LSCM розкладання UV, функціональна інтеграція YafRay, виважені складки у підподіленних поверхнях, градієнтні шейдери;
- 2.35 – листопад 2004 року: Повноцінні поліпшення: гачки об'єктів, деформації кривих та звуження кривих, дублікатори частинок;
- 2.36 – грудень 2004 року: Стабілізована версія, покращення розкладання нормалей та зміщення їх;
- 2.37 – червень 2005 року: Інструменти трансформацій а також віджети, м'які тіла, силові поля, відхиляння, прирістне підподілення поверхонь, прозорі тіні та багато-потоківий рендеринг;
- 2.40 – грудень 2005 року: Повна перероблення системи арматури, ключі форм, хутра, рідини й тверді тіла;
- 2.41 – січень 2006 року: Багато виправлень а також добавлені деякі функції Ігрового Рушія;
- 2.42 – липень 2006 року: Випуск вузлів, модифікатор масиву, векторне розмиття, новий фізичний механізм, рендеринг, синхронізація губ;

- 2.43 – лютий 2007 року: Сітки з багаторазовою роздільною здатністю, багатошарові ультрафіолетові текстури, багатошарові зображення та багатопрхідна візуалізація та запікання, скульптурування, ретопологія, кілька додаткових матів, вузли спотворення та фільтрації, покращення моделювання та анімації, краще малювання кількома пензлями, частинки рідини, проксі-об'єкти, переписування секвенсора та пост-продакшн УФ-текстурування;
- 2.44 – травень 2007 року: Великі новини, на додаток до двох нових модифікаторів та відновлення підтримки 64-бітних ОС, було додано підповерхневе розсіювання, яке імітує розсіювання світла під поверхнею органічних та м'яких об'єктів;
- 2.45 – вересень 2007 року: виправлення помилок щодо деяких проблем продуктивності;
- 2.46 – травень 2008 року: Випуск Reach став результатом величезних зусиль понад 70 розробників, які забезпечили покращення волосся та хутра, нову систему частинок, покращений перегляд зображень, тканину, безшовний і ненав'язливий кеш фізики, покращуючи відображення, АО та gender baking, модифікатор Mesh Deform для м'язів тощо, краща підтримка анімації за допомогою інструментів арматури та малювання, обробки шкіри, кольорового редактора дій;
- 2.47 – серпень 2008 року: Випуск виправлення вад;
- 2.48 – жовтень 2008 року: Реліз Arpicot, шейдери GLSL, підсвічування та покращення GE, знімки, симулятор неба, модифікатор Shrinkwrap та покращення редагування Python;
- 2.49 – червень 2009 року: Базовані на вузлах текстури, нарисування арматури, покращення булевих операцій із сіткою, підтримка JPEG2000, проекційне малювання для прямого переносу зображень на моделі, а також значний каталог скриптів на Python. Покращення

ігрового рушія, оновлення рушія фізики Bullet, купольний рендеринг та багато викликів з API GE;

Blender «Перепис коду»:

- 2.5x – з 2009 по серпень 2011 року: У цій серії було випущено чотири перед-версії – pre-version, від Альфа 0 у листопаді 2009 року до Бета у липні 2010 року і три стабільні версії із 2.57 - квітень 2011 року по 2.59 - серпень 2011 року. Це один з найбільш важливих проектів розробки з повним переробленням програми, з новими функціями, перебудовою внутрішнього віконного менеджера та системи обробки подій/інструментів/даних, а також новим Python API;

Blender «Покращення і Стабілізування»:

- 2.60 – жовтень 2011 року: Інтернаціоналізація інтерфейсу користувача, покращення у системі анімації та ігровому рушії, групи вагомостей вершин модифікаторів, 3D аудіо і відео та виправлення вад;
- 2.61 – грудень 2011 року: Рендерер Cycles було додано у стовбур, додано відстежувач камери, динамічне малювання для модифікування текстур, модифікатор «Океан» для імітації океану і піни на його поверхні, нові додатки, розширення Python API;
- 2.62 – лютий 2012 року: Була додана бібліотека Carve library для покращення булевих операцій, додана підтримка для відстеженні об'єктів, додано модифікатор «Перебудова», покращення матриці й вектору у Python API;
- 2.63 – квітень 2012 року: Vmesh було об'єднано у стовбур з повною підтримкою n-бічних полігонів, панорамна камера для Cycles, дзеркально-кульові текстури середовища та текстури з точністю дійсних чисел, маскові шари для шарів рендера, загорода оточення, показ в

оглядвікні фонових зображень та шарів рендера. Нові додатки імпорту та експорту;

- 2.64 – жовтень 2012 року: Додано редактор масок, а також покращений трекер руху, покращення циклів, покращення секвенсора, кращі інструменти сітки, нові вузли ключів, маскування скульптури, покращення Collada, новий модифікатор шкіри, новий композитинг бекенд вузлів;
- 2.65 – грудень 2012 року: Покращення системи вогню і диму, анізотропичний шейдер для Cycles, покращення модифікаторів, засіб Скосу тепер включав округлення;
- 2.66 – лютий 2013 року: Динамічна топологія, симуляція твердого тіла, покращення в інтерфейсі користувача та використовності, Cycles став підтримувати волосся, засіб Скосу став підтримувати скошування окремих вершин, нові модифікатори «Кеш Сіті» – Mesh Cache та «Жолоблення UV» – UV Warp, новий рішальник SPH частинок рідини;
- 2.67 – травень 2013 року: Було додано Freestyle, покращення системи малювання, підповерхневе розсіювання для Cycles, бібліотека Ceres у відстежнику руху, нові кастомні вузли на Python, нові засоби моделювання сітей, краща підтримка для тексту UTF-8 та покращення у редакторах Тексту, нові додатки для 3D друку;
- 2.68 – липень 2013 року: Нові й покращені засоби моделювання, три нові вузли для Cycles, великі покращення у відстежнику руху, скрипти та драйвери на Python;
- 2.69 – жовтень 2013 року: Ще більше засобів моделювання, покращено Cycles у багатьох ділянках, площинне відстеження було додане у відстежник руху, краща підтримка для імпорту/експорту FBX;

- 2.70 – березень 2014 року: Cycles отримує базову підтримку об'ємів на CPU, більше покращень для відстежника руху, два нові модифікатори моделювання;
- 2.71 – червень 2014 року: Для Cycles додано розмив рухом при деформуванні та підтримку системи вогонь/дим, спливні меню інтерфейсу користувача стали тепер перетягувані. Оптимізації продуктивності для режиму ліплення, нові типи інтерполяції для анімації;
- 2.72 – жовтень 2014 року: Цикли отримують обсяг і підтримку SSS на графічному процесорі, додані секторні меню та значно покращені підказки, додано інструмент моделювання перетину, новий вузол Sun Beam для Compositor, Freestyle тепер працює з циклами, покращено робочий процес малювання текстур;
- 2.73 – січень 2015 року: Cycles отримує покращену підтримку об'ємів, версія під Windows отримує редактори методу вводу – Input Method Editors (IMEs) та загальні покращення щодо малювання;
- 2.74 – березень 2015 року: Підтримка для кастомних нормалей, комбінування оглядвікна та покращення динаміки волосся;
- 2.75 – липень 2015 року: Вбудований конвеєр стерео переглядів, модифікатор Corrective Smooth і новий графік залежності розвитку;
- 2.76 – листопад 2015 року: Підтримка техніки Pixar OpenSubdiv, приріст продуктивності для Оглядів вікна та Браузера Файлів, авто-зсув для вузлів та смужка ефекту тексту для Секвенсера;
- 2.77 – березень 2016 року: Підтримка OpenVDB для кешування симуляцій диму/об'ємів, покращене підповерхневе розсіювання для Cycles, покращений робпотік та ліплення штрихів Нарисного Олівця, перероблена бібліотека оброблення для керування відсутніми та видаленими блоками даних;

- 2.78 – вересень 2016 року: Підтримка циклів сферичних стереозображень для віртуальної реальності, Grease Pencil більше схожий на інше програмне забезпечення для 2D-малювань, підтримка імпорту та експорту Alembic, а також покращення Bendy Bones та простішого монтажу;
- 2.79 – вересень 2017 року: Нові функції в Cycles: Знешумлення, Ловець тіней та новий шейдер Principled. Інші покращення були зроблені для Нарисного Олівця та Alembic. Була також додана підтримка для шаблонів застосунку;

Blender «Інтерфейс Користувача»

- 2.80 – липень 2019 року: Повністю перероблений інтерфейс користувача для спрощення навігації; покращене область перегляду та інструменти. З Eevee був створений новий фізично заснований механізм візуалізації в реальному часі. Grease Pencil був капітально перероблений і тепер є повною системою 2D малювання та анімації;
- 2.80 – листопад 2019 року: Оновлені інструменти скульптури, прискорена візуалізація Cycles OptiX, шумозаглушення, багато вдосконалень Eevee, заміни бібліотек, покращення інтерфейсу користувача;
- 2.80 – лютий 2020 року: Підтримка UDIM та USD, Mantaflow для моделювання рідин та диму, шумозаглушення AI, покращення Grease Pencil.

Visual Studio

Історія

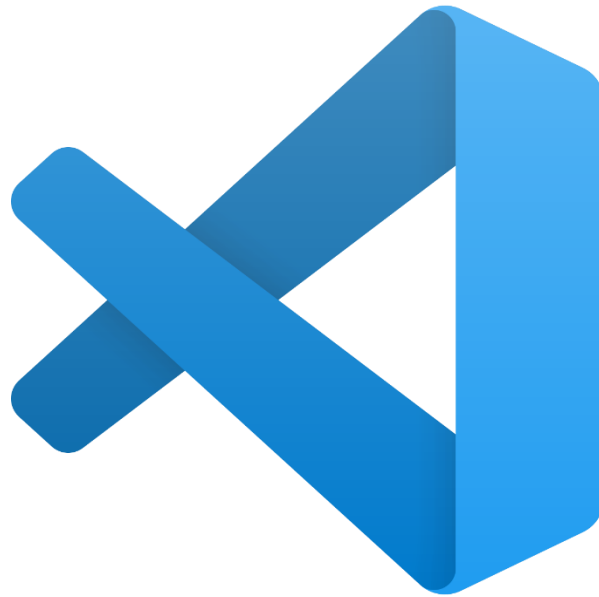


Рис. 1.9 Логотип Visual Studio

Перша версія Visual Studio вийшла 1995 році хоча в 90 роках 19 століття Microsoft уже випускала деякі продукти програмування мовами високого рівня. Розроблене Microsoft інтерактивне середовище програмування мовою Microsoft Visual C++, розширені мови C++. Саме Visual C++ у складі всіх версій Visual Studio – найпопулярніша мова програмування у світі.

Завдяки поєднанню простоти мови BASIC із новітніми об'єктивно-орієнтованими розширеннями Microsoft розробив Visual BASIC, який використовується програмістами всього світу.

Microsoft Visual FoxPro – інтерактивне середовище програмування мовою Visual FoxPro – об'єктивно-орієнтованою мовою з елементами процедурного програмування. Завдяки цій мові отримано можливість звертатись нею безпосередньо до бази даних, заснованих на мові SQL, зокрема програмування SQL-запитів на мові FoxPro.

Microsoft розроблено систему управління версіями вихідних кодів і інтегровано з середовищем Visual Studio - Microsoft Visual SourceSafe.

У 1995 році вище названі продукти були об'єднані в Visual Studio і отримали номер версії 4.0. Завдяки переplatі MSDN, де надходили новітні програмні розробки Microsoft, версія отримала широку популярність.

У 1997 році вийшла 5.0 версія, або Visual Studio 97.

Нововведення: Visual Studio InterDev, інструмент розробки інтерактивних Web-сайтів на основі технології ASP та Visual Java++; Java++ - середовище програмування на Java від Microsoft; Технологія ASP – призначена для розробки активних веб сторінок – шаблонів, для генерації HTML-сторінок для клієнтських браузерів як відповіді на Web-запитання. На основі цієї технології згодом розроблено технології ASP.NET.

Отже, версія Visual Studio 5.0 – це інтегроване середовище програмування Visual C++, Visual Basic, VJ++. Вона має інструмент Visual Studio InterDev, який призначений для розробки інтерактивних веб сайтів.

Visual Studio 6.0 вийшла 1998 році. Вона мала нові компоненти: для підтримки візуального моделювання, і аналізу продуктивності програм: Automation Manager, Application Performance Explorer, RemAuto Connection Manager, Visual Studio Analyzer.

Ця версія стала основою для створення нової версії інтегрованого середовища, яке включає єдину багатомовну платформу для використання програм - .NET.

Головним нововведенням у версії Visual Studio 7.0, випущений у 2002 році, стала реалізація багатомовної платформи .NET Framework. Саме ця платформа забезпечила надійне програмування з повним контролем типів і безпеку на базі керованого коду загальної інфраструктури мов.

Версія Visual Studio 7.0 відома за назвою Visual Studio.NET була випущена у чотирьох виданнях: Academic, Professional, Enterprise Developer, Enterprise Architect. У ній було розроблено і реалізовано нову мову програ-

мування - C#, значні розширення торкнулися і мов Visual C++ і Visual Basic.

Сумісну з NET нову версію мови Visual C++ названо Managed C++, а нову версію Visual Basic - Visual Basic.NET.

Visual Studio.NET 2003 яка вийшла у 2003 році і як попередня версія мала чотири видання глобальних змін не зазнала, однак здійснила істотне нововведення – реалізацію версії .NET для мобільних пристроїв - .NET Compact Framework та підтримку розробки програм засобами Visual Studio для мобільних пристроїв. У виданні Enterprise Architect - Visual Studio 2003 реалізована підтримка засобів моделювання програм з підтримкою UML-діаграм.

Основним нововведенням нової версії Visual Studio.NET 2005, яка вийшла у 2005 році стала реалізація параметризованих типів у версії мови C# 2.0 та у .NET Framework 2.0.

Удосконалення торкнулися ASP.NET – розвитку технологій ASP, призначеної для реалізації так активних веб сторінок, властивостей платформи NET, а також додаткових видів проектів підтримки концепції ASP.NET веб-сервісів.

Visual Studio.NET 2008, яка у корпорації Microsoft значиться як Visual Studio 9.0, вийшла у 2008 році і була призначена для Windows Vista. Вона містить розвинені засоби підтримки веб-програмування. Версія .NET Framework що відповідає версії Visual Studio 2008 .NET Framework 3.5.

У версії Visual Studio 2008 додано нового візуального дизайнера додатків, які використовують API для підтримки GUI, реалізовано підтримку налагодження багатопотокових додатків, а також підтримка мови J# - запропоновано інструмент Java Language Converter Assistant, який входить в оточення для перенесення програм з мови та середовища Java в оточення NET.

У квітні 2010 року випущено версію Visual Studio.NET 2010. Найважливішим нововведенням у цій версії є реалізація підтримки функціональної мови F#, автором якої став Дон Сайм.

Перспективність мови F# полягає в поєднанні суто функціонального програмування із засобами об'єктивно-орієнтованого програмування, паралельного програмування та засобами інтеграції з усіма можливими інтеграціями .NET Framework та Visual Studio.

Visual Studio 2012 яка вийшла у 2012 році передбачала ряд покращень: удосконалено інтерфейс користувача, підтримку кольорів для виділення фрагментів коду, розроблено новий інструмент перегляду solution explorer.

РОЗДІЛ 2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО МОДУЛЮ ДЛЯ СТВОРЕННЯ 3D-ОБ'ЄКТІВ

2.1 Модель 3D-об'єкту віртуальної лабораторної роботи

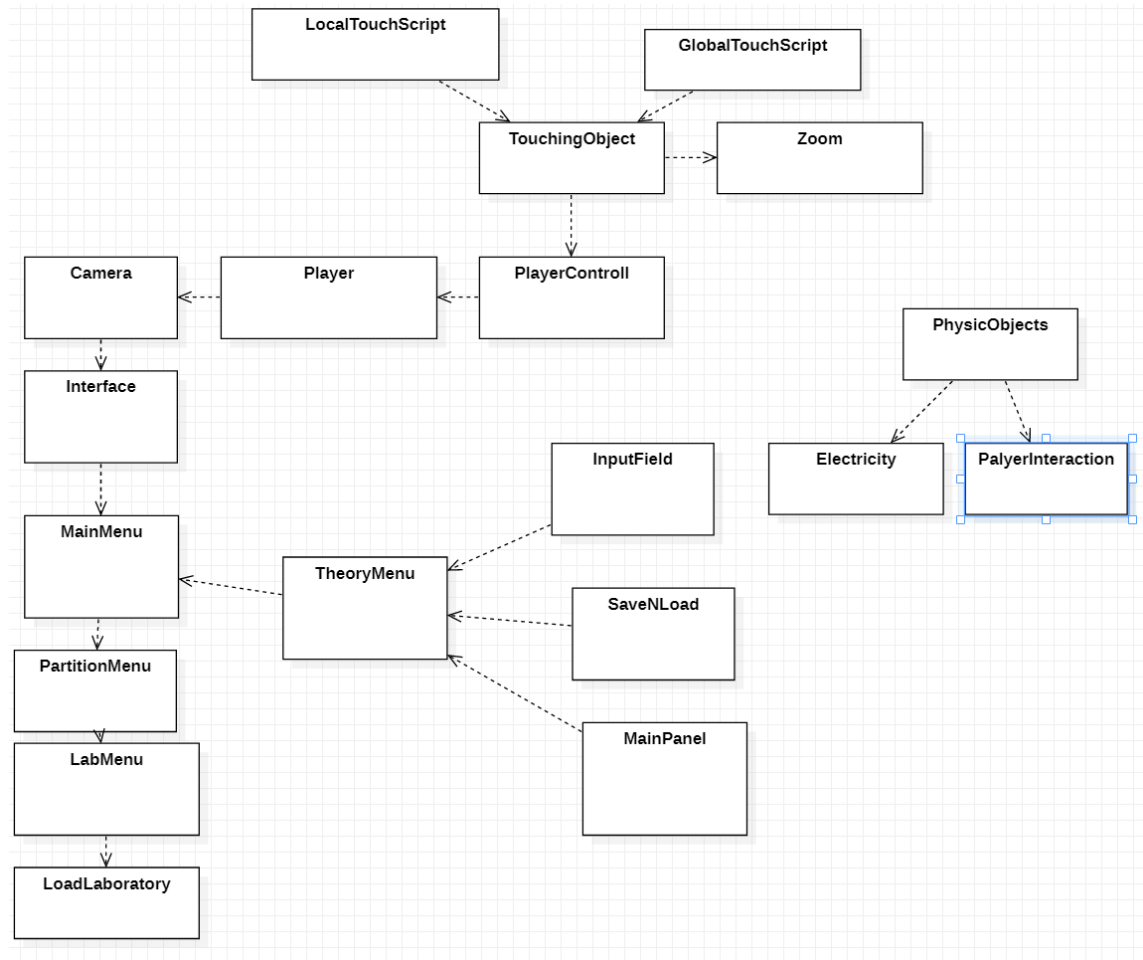


Рис 2.1 Діаграма класів

Від основного класу Camera успадковуюється більшість класів. Клас Camera – це основні зв'язки користувача з системою.

Клас Player – методи, які відповідають за появу гравця в кімнаті, звукові ефекти.

Клас PlayerControll – відповідальний за взаємодію користувача з системою, містить методи керування камерою і гравця. Методи класу переводять свайпи пальцем по екрану в координати і рухають камеру.

Клас TouchingObject – містить основні методи конвертації координат дотику до екрану в координати в тривимірному просторі. Щоб визначити точки дотику система випускає промінь, і якщо він дотикається до об'єкту, дотик зараховується. За жести по сенсору смартфона відповідають

класи LocalTouchScript та GlobalTouchScrip. Global координує координати дотику, які генеруються завдяки променів, а Local робить роботу камери плавною і відповідає за свайпи. Клас Zoom відповідає за процеси приближення і віддалення камери.

За генерацію різних пунктів меню відповідають класи Main. За процес завантаження віртуальної лабораторної роботи відповідають клас LoadLaboratory. В випадку коли лабораторна відкривається повторно відбувається зчитування з XML файлу даних.

Клас InpitField – найважливіший у системі, яка відкриває теоретичні відомості.

Оскільки в Unity немає підтримки текстових полів з полосами прокрутки та переносу тексту, цей клас модернізує стандартне поле, воно ж виконує функцію зміни шрифту і кольору.

За завантаження і зберігання тексту в XML-файлі відповідальний клас SaveNLoad.

MainPanel – генерує теоретичну панель, калькулятор, і редактор формул і є найбільш класом.

Physic – це клас самих об'єктів, який додав до них фізичну модель та теги, для взаємодії з ними.

Клас Electricity описує ідеальну електричну модель для лабораторних робіт з електрики. В ньому враховано тип матеріалу, за допомогою тегів об'єктів, електричний опір, напруга.

Завдяки класу PlayerInter об'єкти взаємодіють друг з другом та гравцем.

2.2 Проектування програмного забезпечення 3D-об'єкту віртуальної лабораторної роботи

Для введення тексту використовувався «InputField» з розробленим для нього сценарієм на мові C#.

Завдання сценарію полягає в переносі курсиву на іншу строку коли текст доходить до кінця «InputField» і переносі слова якщо воно не закінчилося у попередньому рядку.

Також до «InputField» було добавлено колесо вертикальної прокрутки, за допомогою його коли текст доходить до кінця поля поле буде прокручувати текст, що дозволить писати тексти нескінченної величини.

Далі була реалізована сама кнопка редактору, яка ховає редактор при натисканні на неї та показує його знову при повторному натисканні. По замовчуванню поле редактора та всі його кнопки невидимі, за це відповідає окремий сценарій «Enum». В ньому присутні три змінні, дві з них відповідають за панелі на яких розміщуються текстовий редактор та редактор формул, третя відповідає за саму кнопку.

При старті, сценарій, усім панелям та всім об'єктам що на них розташовані, дає методу (SetActive) атрибут (false), таким чином панелі робляться неактивні і невидимі. При натисканні на кнопку, виконується функція що змінює атрибут методу (SetActive) панелі текстового редактору на (true). Та при повторному натисканні змінює атрибут на протилежний.

2.3 Проектування віртуальної лабораторної роботи. Діаграма класів

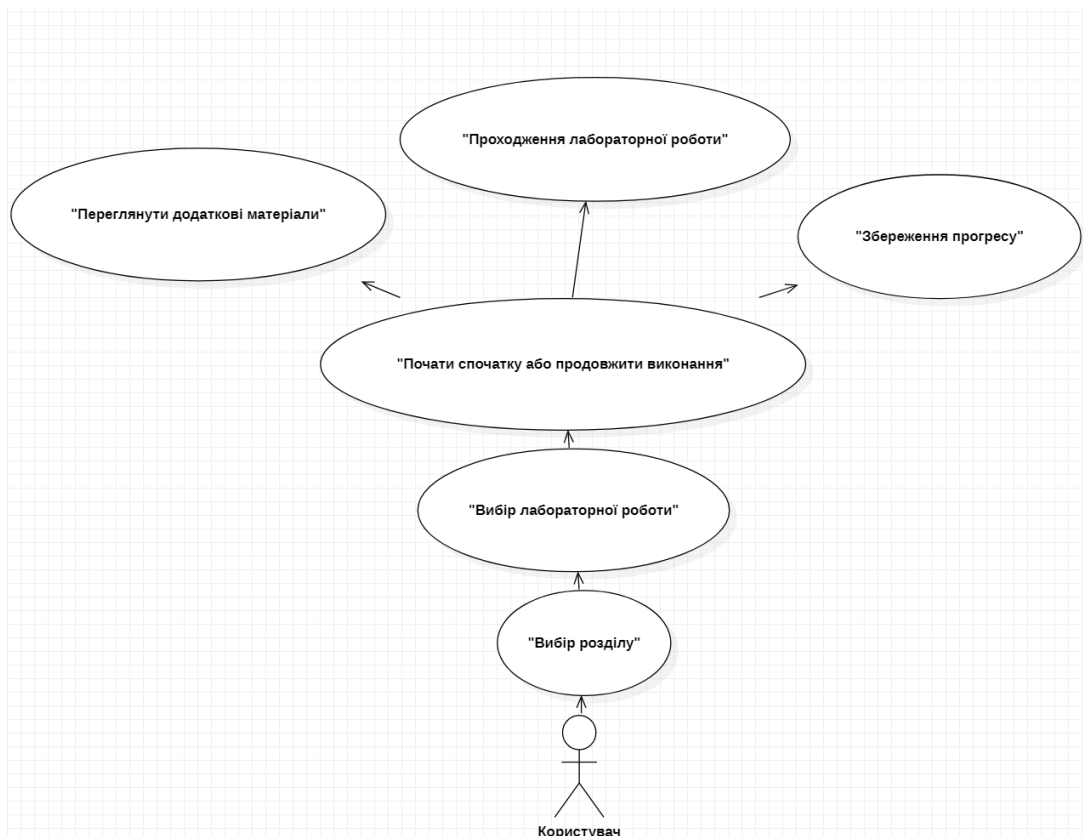


Рис. 2.5 Діаграма варіантів використання системи

Зайшовши в мобільний додаток у розділ з фізики, можна пройти віртуальну лабораторну роботу. Спочатку обираємо розділ «Механіка, електрика, молекулярна фізика». Коли відкривається меню вибору роботи, система пропонує вибір між загрузкою минулого прогресу та початком проходження з нуля. Після завантаження відкривається робочий стіл для проходження роботи. Внизу екрану є кнопка виходу та збереження прогресу, вгорі – кнопка «меню», кнопка «рестарту роботи». У «меню» з додатковими матеріалами розмішені теоретичні відомості про лабораторну, «інструкція з виконання», «редактор формул» і «калькулятор». Після виконання студент отримує повідомлення з розрахунком затраченого часу на виконання роботи і оцінку.

В залежності від кількості проходжень, прогрес проходжень записується: перший в першу клітинку, інші в другу, де і буде записано результат останнього проходження. Весь прогрес зберігається в додатку і завантажується при наступному відкритті.

РОЗДІЛ 3 РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 3D-об'єкти у віртуальних лабораторних роботах

Проведемо розробку 3D моделювання на прикладі розробки столу, для основи візьмемо куб і проведемо маштабування, поки стіл не прийме форму стільниці.

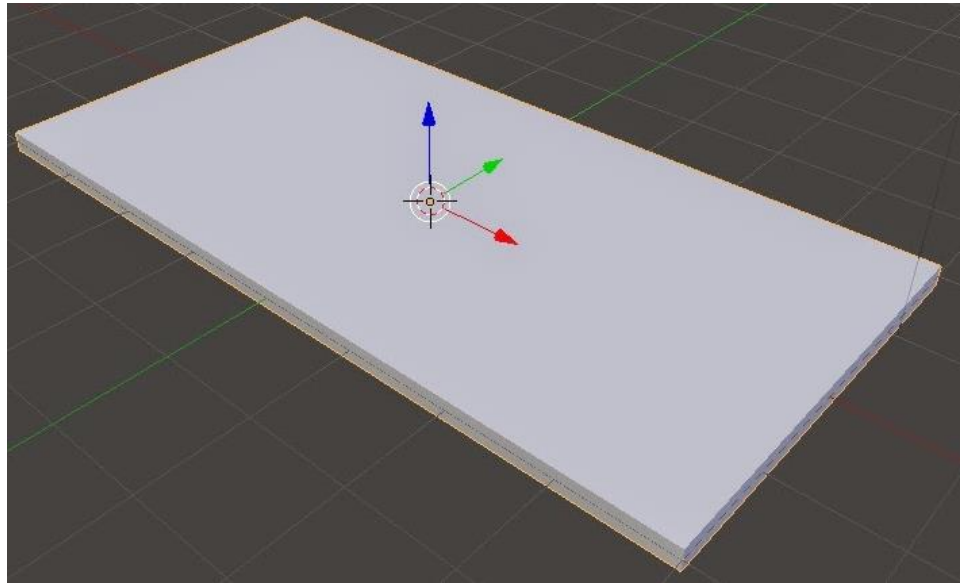


Рис 3.1 Моделювання кубу у формі стільниці

Наступний куб буде слугувати ножкою столу.

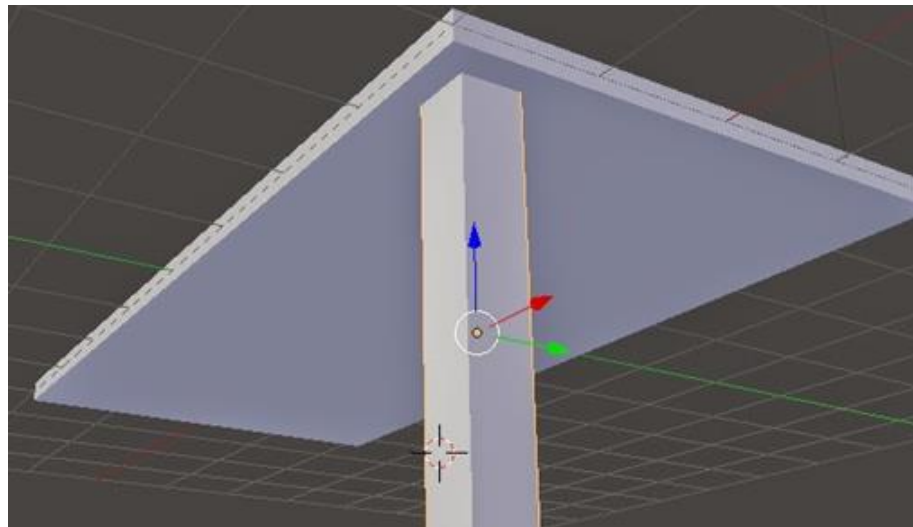


Рис 3.2 Моделювання ніжки столу

Продублюємо наступні ніжки і отримуємо модель схожу на стіл.

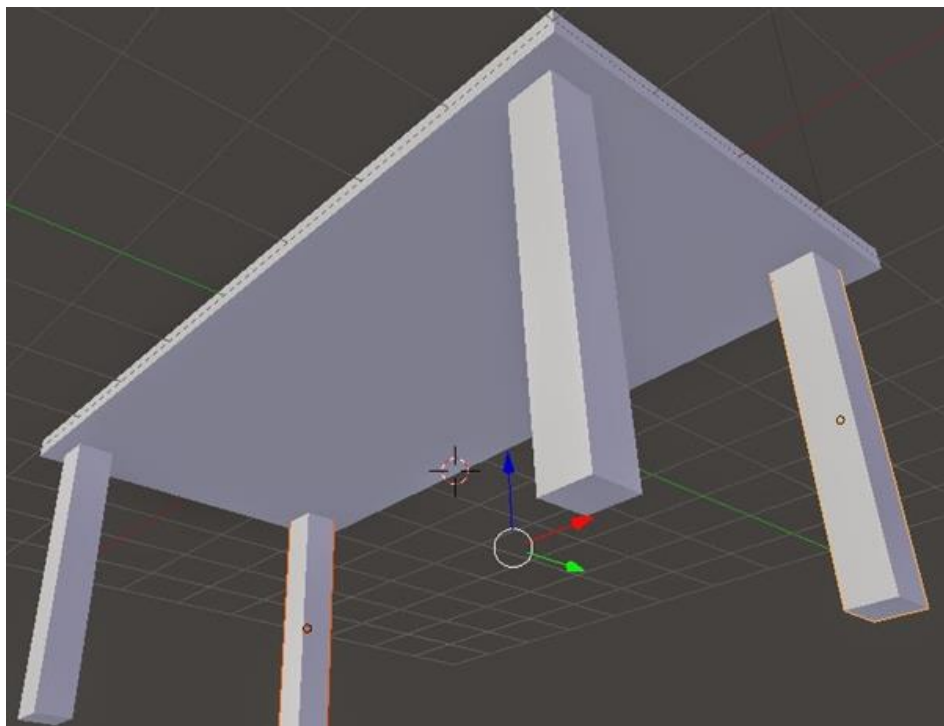


Рис 3.3 Моделювання столу

Проведемо обробку на предфінальному рендеренгу і стіл матиме такий вигляд.

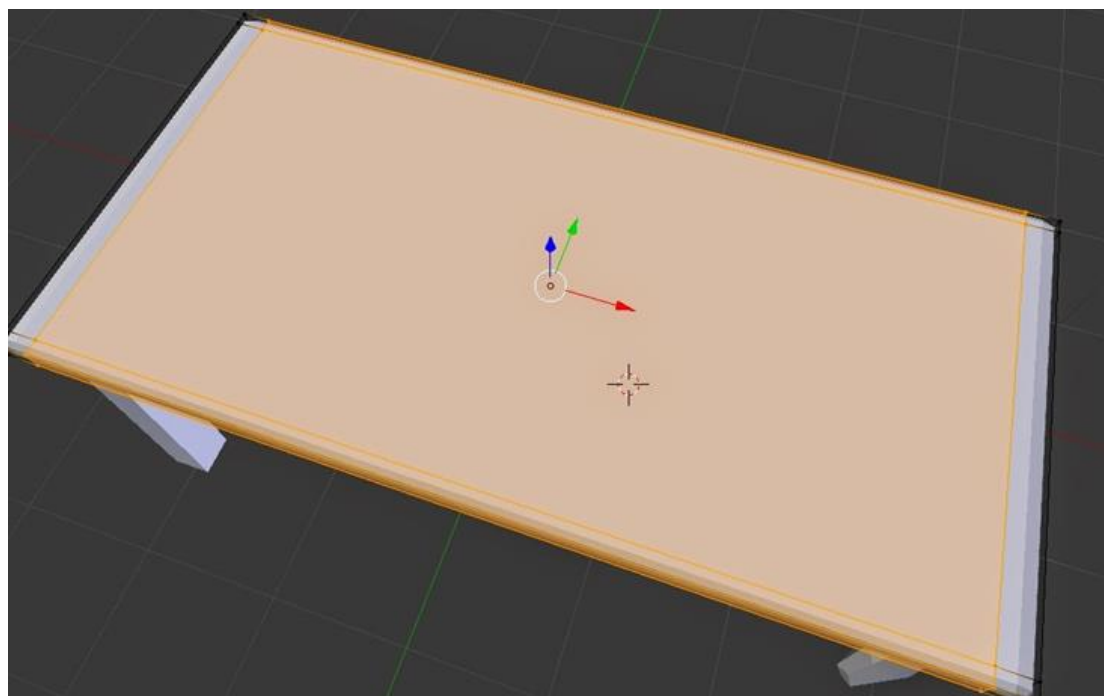


Рис 3.4 Перед фінальна модель столу

На останньому етапі заокруглюємо грані столу, окрім столу для віртуальної лабораторії, розроблені об'єкти з електрики: моделі проводів, амперметр, вольтметр, розетка, дощечки для збору електричних ланцюгів, вікно, кімната.

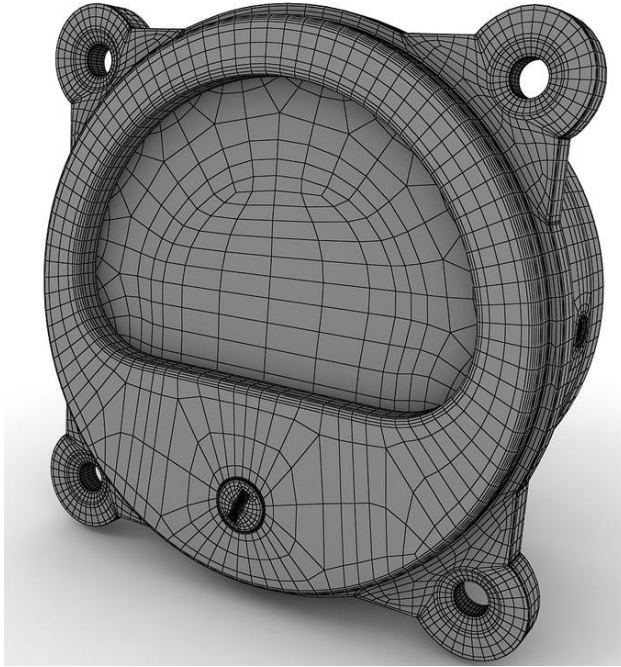


Рис 3.5 Вигляд не відрендерованої моделі вольтметра

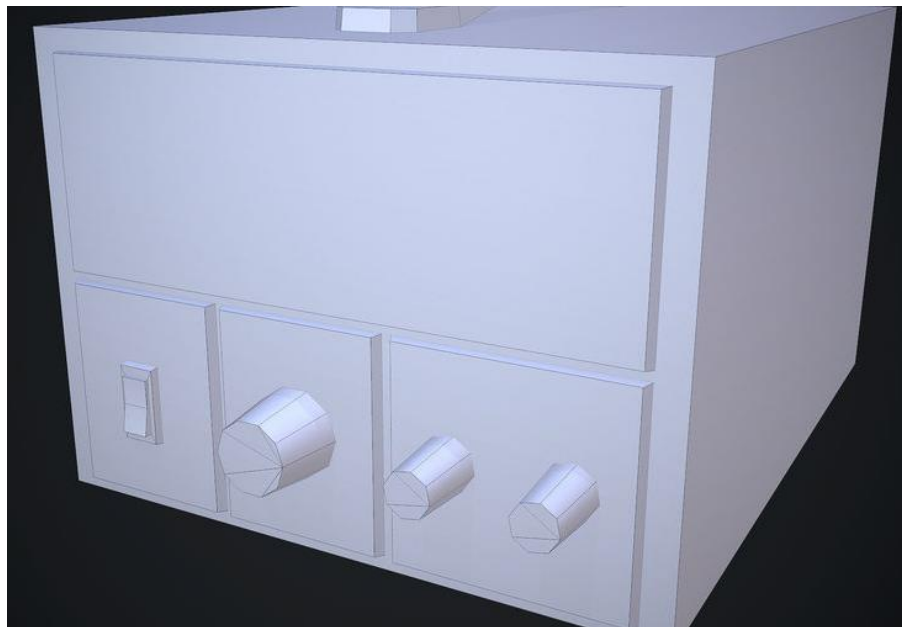


Рис 3.6 Вигляд не відрендерованої моделі амперметр

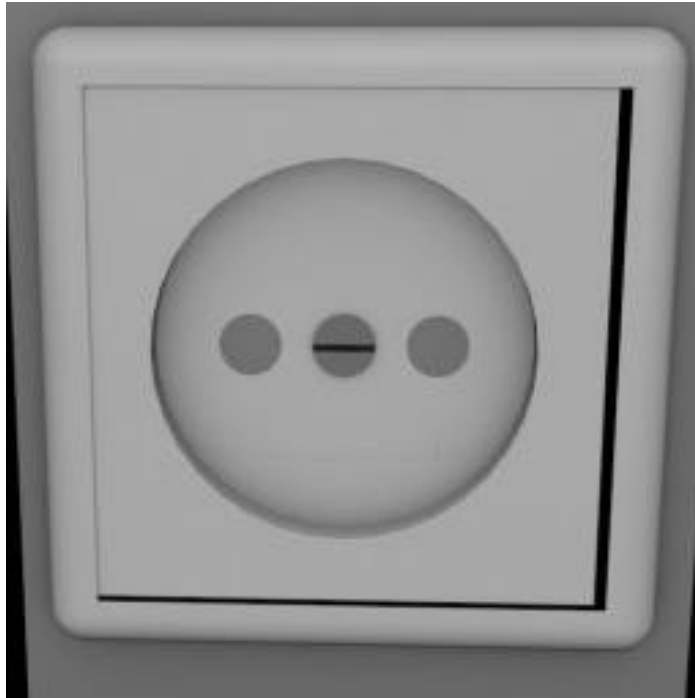


Рис 3.7 Вигляд не відрендерованої моделі розетки

3.2 Розробка 3D-об'єктів віртуальних лабораторних робіт з курсу фізики

Для створення керування гравцем у мобільному просторі не підходить звичайна модель керування за допомогою пульта у правому і лівому, нижніх кутках екрана. Така модель керування не підходить для лабораторної роботи, оскільки все базується на взаємодії гравця з об'єктами. Оскільки користувач мобільної лабораторії не переміщується по кімнаті, прийнято рішення заблокувати цю можливість. Користувач лабораторії буде стояти біля столу і взаємодіяти з об'єктами:

1. Крутити камеру, за допомогою свайпів по екрану.
2. Проходити налаштування зуму за допомогою жесту.
3. Взаємодіяти з об'єктами за допомогою дотиків.

Основа розробки свайпів запам'ятовування першої точки дотику пальцем і точки відпускання пальцю. Щоб розрахувати силу свайпа вираховували відстань між цими точками а величину масштабували. Визначились координати і поняття на скільки сильно треба повернути камеру. В залежності від довжини свайпу, система повертає камеру швидше і сильніше, чим довший свайп. Для проведення свайпу, незалежно від його відстані, скрипт має однаковий час. Завдяки цьому довгі свайпи швидкі і повертають камеру на потрібну відстань. Короткі свайпи плавно переміщують об'єкт.

В системі існує мінімальна довжина свайпу, що дозволяє уникнути помилок в роботі.

Налаштування зуму складається з одного методу, це найменший клас `PlayerControl`. Приближення та віддалення працює завдяки доторкуванню двома пальцями одночасно, при цьому відбувається запис координат дотику.

Наступні дві точки система записує коли користувач віддаляє або підводить пальці і відпускає панель. Наступний крок – система знаходить різницю між точками і визначає це жест приближення чи віддалення. Після конвертації точок у координати для камери, підсовую її вперед або назад.

Третій клас в керуванні гравцем відповідає за взаємодію з об'єктами. При дотику до екрану пальцем система випускає промінь, який зустрівшись з об'єктом зі спеціальним тегом, зараховує дотик до об'єкту. Якщо продовжити тримати палець на сенсорній панелі система буде тривати об'єкт у повітрі, ближче до камери гравця, так щоб його було видно. Якщо ж палець відпустити - промінь зникне, а гравець втратить контроль над об'єктом і піддасться силі гравітації. При наближенні об'єктів система перевертатиме об'єкти на сумісність.

3.3 Інтерфейс програмного забезпечення віртуальної лабораторної роботи

Користувачський інтерфейс створений за допомогою стандартних об'єктів кнопок, за роботу яких відповідають спеціально розроблені сценарії. Спеціально розроблені сценарії виконують різні методи роботи при натисканні кнопок. Щоб включити повторювання коду існують класи, які відповідають за пункти меню. Клас меню LoadLaboratory, зчитує інформацію з XML для завантаження лабораторної.

Головне меню передбачає наявність трьох кнопок. Коли користувач натискає кнопку «Теоретичні матеріали» система сама переходить на нову сцену, при натисканні «виходу» виконується сценарій виходу і очистка пам'яті.

Завантаження лабораторної роботи відбувається в певній послідовності LoadLaboratory, який зчитує інформацію з XML файлу і відкриває лабораторну.

Вразі повторного проходження роботи користувачу пропонується завантажитись з останнього моменту.

При виборі пункту «теоретичні матеріали» користувачеві пропонується «Розділ Фізики», а потім лабораторна робота, далі відкривається панель, яка складається із звичайного поля вводу з розробленим для нього особливим сценарієм.

Введення і підтримку тексту контролює клас, який відповідає за панель. При наявності достатньої кількості символів у рядку, курсор переноситься на новий. Є можливість зміни шрифту і кольору. Інформація зберігається у файлах формату XML і при виборі тієї чи іншої роботи завантажується у текстове поле. При необхідності прокрутити текст, достатньо

провести пальцем з низу в гору. Для використання не стандартних символів для формул існує підтримка введення нестандартних символів.

ВИСНОВКИ

Під час написання дипломної роботи, було проаналізовано існуючі віртуальні лабораторії виділено їх переваги і недоліки, шляхи удосконалення версії Unity на протязі певного часу.

Використовуючи вже існуючі системи, розроблено нову віртуальну лабораторію для мобільних пристроїв, в тривимірному просторі, побудувавши модель всієї системи, розпочата робота по проектуванню текстового редактора, розробка сценаріїв та їх взаємодії з 3D-моделями. Для останніх використано безкоштовний софт Blender. З огляду на переваги Unity серед інших версій (UnrealEngine, CryEngine), що включають зручність в роботі з мобільним додатком, легкість в зібранні рішень для мобільного і запуском його, мову програмування C#, Unity було обрано ігровим рушієм для віртуальної лабораторії. Данна версія має відміну оптимізацію для мобільних пристроїв а додатки розроблені на цьому двигуні відкриваються навіть на Android 4.0. Двигун дозволяє створювати свої власні додатки і імпортувати в нього свої власні об'єкти. Він має широкий функціонал і величезний набір інструментів, що дає можливість виконати завдання.

Розробка для мобільних телефонів вимагала враховувати апаратні можливості мобільних телефонів, особливо, при роботі з трьохвимірною графікою. Довелося провести велику роботу над оптимізацією, розроблено сценарій який орієнтується на систему променів і тегів, що дозволило досягти поставленої мети.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Джон Скит. С# для профессионалов: тонкости программирования: Пidrучник. К., 2014. 608 с.
2. Кристиан Нейгел и др. С# 5.0 и платформа .NET 4.5 для профессионалов = Professional C# 5.0 and .NET 4.5: Пidrучник. М., 2013. 1440 с.
3. А. Хейлсберг, М. Торгерсен, С. Вилтамут, П. Голд. Язык программирования С#. Классика Computers Science. 4-е издание = C# Programming Language (Covering C# 4.0): Пidrучник. К., 2012. 784 с.
4. Эндрю Троелсен. Язык программирования С# 5.0 и платформа .NET 4.5, 6-е издание = Pro C# 5.0 and the .NET 4.5 Framework: Пidrучник. М., 2013. 1312 с.
5. Джозеф Албахари, Бен Албахари. С# 6.0. Справочник. Полное описание языка = C# 6.0 in a Nutshell: The Definitive Reference: Пidrучник М.,: 2018. — 1040 с.
6. Герберт Шилдт. С# 4.0: полное руководство = C# 4.0 The Complete Reference.: Пidrучник. М., 2010. — С. 1056.
7. Кристиан Нейгел, Карли Уотсон и др. Visual C# 2010: полный курс = Beginning Microsoft Visual C# 2010.: Пidrучник. М., 2010.
8. Хокинг, Джозеф. Unity — в действии. Мультиплатформенная разработка на С#: Пidrучник. К., 2016. 336 с.
9. Торн, Алан. Искусство создания сценариев в Unity: Пidrучник. К., 2016. - 362 с.
10. Джеймс Крониестер / James Chronister. Основы Blender учебное пособие 4-е издание / Blender Basics 2.6: Пidrучник. К., 2012. С. 416.

11. *Кристиан Нейгел и др. С# 5.0 и платформа .NET 4.5 для профессионалов = Professional C# 5.0 and .NET 4.5.: Підручник. М., 2013. 1440 с.*
12. *Иванов Б. Н. Законы физики. 3-е изд.: Підручник. М., 2004. 368 с.*
13. *Фейнман Р. Фейнмановские лекции по физике. Электричество и магнетизм.: Підручник. М., 2004. — 304 с.*
14. *Дэвид Хантер, Джефф Рафтер, Джо Фаусетт, Эрик ван дер Влист, и др. XML. Работа с XML, 4-е издание = Beginning XML, 4th Edition.: М., 2009. — 1344 с.*
15. *Дэвид Хантер, Джефф Рафтер и др. XML. Базовый курс = Beginning XML.: М., 2009. — 1344 с.*
16. *Роберт Тейбор. Реализация XML Web-служб на платформе Microsoft .NET = Microsoft .NET XML Web Services.: М., 2002. — 464 с.*
17. *Ник Рендольф, Дэвид Гарднер, Майкл Минутилло, Крис Андерсон. Visual Studio 2010 для профессионалов = Professional Visual Studio 2010 Підручник. М.,: 2011. С. 1184.*
18. *Алекс Макки. Введение в .NET 4.0 и Visual Studio 2010 для профессионалов = Introducing .NET 4.0: with Visual Studio 2010. Підручник. М., 2010. С. 416.*
19. *Карли Уотсон, Кристиан Нейгел, Якоб Хаммер Педерсен, и др. Visual C# 2008: базовый курс. Visual Studio® 2008 = Beginning Visual C# 2008.: Підручник. М., 2009. С. 1216.*
20. *Пауэрс Л., Снелл М. Microsoft Visual Studio 2008 = Microsoft Visual Studio 2008 Unleashed by Lars Powers and Mike Snell. Підручник. С., 2008. С. 1200.*
21. *Майо Д. Самоучитель Microsoft Visual Studio 2010 = Microsoft Visual Studio 2010: A Beginner's Guide (A Beginners Guide). Підручник. С., 2010. С. 464.*

22. 3D-модельовання: визначення, технічні додатки та кар'єрні можливості [Електронний ресурс] – Режим доступу: <https://www.pedago.it/blog/modellazione-3d-definizione-applicazione-lavoro.htm#:~:text=Cos'%C3%A8%20la%20Modellazione%203D,programmi%20particolari%20chiamati%20software%203D>.
23. 3D-модельовання: 20 найкращих програм для 3D-друку [Електронний ресурс] – Режим доступу: <https://www.printit3d.com.br/post/modelagem-3d-20-melhores-sofware-para-impress%C3%A3o-3d>
24. Unity ігровий двигун [Електронний ресурс] – Режим доступу: [https://ru.wikipedia.org/wiki/Unity_\(%D0%B8%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA\)](https://ru.wikipedia.org/wiki/Unity_(%D0%B8%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA))
25. 3D Modeling [Електронний ресурс] – Режим доступу: https://es.wikipedia.org/wiki/Modelado_3D#Proceso_de_modelado
26. What is 3D Modeling & How Is 3D Modeling Used [Електронний ресурс] – Режим доступу: <https://homesthetics.net/what-is-3dmodeling>
27. 3D modeling [Електронний ресурс] – Режим доступу: <https://whatis.techtarget.com/definition/3D-modeling>
28. Двигун Unity – особенности, переваги и недоліки [Електронний ресурс] – Режим доступу: <https://cubiq.ru/dvizhokunity>
29. Перемещение об'єкта по нажатии кнопки в Unity3d на C# [Електронний ресурс] – Режим доступу: <https://habr.com/ru/post/437898/>
30. Движение об'єкта к точке в Unity3D [Електронний ресурс] – Режим доступу: <https://habr.com/ru/post/347904/>

31. Learning C# and coding in Unity for beginners - Unity [Электронный ресурс] – Режим доступа: <https://unity3d.com/ru/learning-c-sharp-inunity-for-beginners>
32. C Sharp [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/C_Sharp
33. C# Tutorial [Электронный ресурс] – Режим доступа: <https://www.w3schools.com/cs/>
34. What is Unity 3D & What is it Used For? [Электронный ресурс] – Режим доступа: <https://conceptartempire.com/what-is-unity/>
35. C Sharp [On-line resource] // Вікіпедія : вільна енциклопедія. [Електронний ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/C_Sharp
36. WebM [On-line resource] // Вікіпедія : вільна енциклопедія. [Електронний ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/WebM>
37. Unity (game engine) [On-line resource] // Вікіпедія : вільна енциклопедія. [Електронний ресурс] – Режим доступа: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))
38. Мультмедийная виртуальная лаборатория по физике в системе дистанционного обучения [On-line resource] // DSpace at KSU [Электронный ресурс] – Режим доступа: <http://ekhsuir.kspu.edu/handle/123456789/832>
39. Виртуальные лаборатории [On-line resource] // АвачаВики [Электронный ресурс] – Режим доступа: http://kamchatkairo.ru/wiki/index.php/Виртуальные_лаборатории
40. Frame rate [On-line resource] // Вікіпедія : вільна енциклопедія. [Електронний ресурс] – Режим доступа: https://en.wikipedia.org/wiki/Frame_rate
41. Edumedia Sciences [On-line resource] // eduMedia. [Электронный ресурс] – Режим доступа: <https://www.edumedia-sciences.com/ru/>

42. MonoDevelop [On-line resource] // Вікіпедія : вільна енциклопедія. [Електронний ресурс] – Режим доступу: <https://en.wikipedia.org/wiki/MonoDevelop>
43. Понятие виртуальной лаборатории. Их использование в образовательном процессе [On-line resource] // Вавилон.wiki. [Електронний ресурс] – Режим доступу: <http://www.wikiznanie.ru/b/index.php/>
44. Manual: Unity User Manual [On-line resource] // Unity. [Електронний ресурс] – Режим доступу: <https://docs.unity3d.com/Manual/index.html>
45. Visual Studio Code [On-line resource] // Вікіпедія : вільна енциклопедія. [Електронний ресурс] – Режим доступу: https://ru.wikipedia.org/wiki/Visual_Studio_Code
46. Visual Studio Code – Code Editing. Redefined [On-line resource] // Visual Studio Code. [Електронний ресурс] – Режим доступу: <https://code.visualstudio.com/>
47. Віртуальна фізика хімія біологія | Віртуальна лабораторія [On-line resource] // ВиртуЛаб. [Електронний ресурс] – Режим доступу: <http://www.virtulab.net/>
48. PhET: Безкоштовні онлайнві моделі з фізики, хімії, біології, математики та природознавства [On-line resource] // PhET. [Електронний ресурс] – Режим доступу: <https://phet.colorado.edu/uk/>
49. Процесс работы с активами. | Unity3d вики | FANDOM powered by Wikia [On-line resource] // FANDOM Wiki. [Електронний ресурс] – Режим доступу: http://ru.unity3d.wikia.com/wiki/Процесс_работы_с_активами